Detección y categorización de objetos invariante y multivista en imágenes digitales mediante visión artificial bioinspirada

Sergio Rodriguez Vaamonde

Director: Dr. Koldo Espinosa

- 2016 -

Detección y categorización de objetos invariante y multivista en imágenes digitales mediante visión artificial bioinspirada

Sergio Rodriguez Vaamonde

Director: Dr. Koldo Espinosa

- 2016 -

Agradecimientos

Esta tesis no habría sido posible sin el impulso del Dr. Artzai Picón, estimulando los primeros pasos de mi carrera investigadora. En lo personal, debo agradecer a mis superiores, dentro de Robotiker y Tecnalia, la posibilidad que he tenido de completar esta tesis. En especial a Ana Ayerbe por brindarme la posibilidad de realizar la investigación de esta tesis a la par del trabajo diario en Robotiker. También agradecer a Jone Echazarra el fomentar la finalización de esta tesis en los últimos años bajo su gestión en Tecnalia.

Mención especial merece mi mentora la Dra. Estibaliz Garrote, quien no sólo ha promovido técnicamente estos trabajos sino que también me ha asistido en la obtención de la financiación necesaria para completar cada paso de esta tesis.

Para el desarrollo de esta tesis se ha necesitado una gran cantidad de recursos económicos y, por ello, quiero agradecer a todas las entidades que han financiado este trabajo. En primer lugar, agradecer a Tecnalia la financiación a través del Grupo de Excelencia Internacional de Computer Vision. Asimismo al Gobierno Vasco por la financiación ETORTEK de los proyectos Smartur, Knowtour y Future, Internet II. Estos han sido el marco principal del desarrollo de las investigaciones de esta tesis. Otras entidades públicas también han colaborado financieramente en esta tesis, como son el CDTI bajo la financiación del proyecto CENIT Buscamedia así como la Comisión Europea con el apoyo y financiación del proyecto BIOPOOL. Por todo ello, quiero agradecer a las instituciones públicas su apoyo y compromiso con la investigación, ya que sin ellas esta tesis no habría sido posible.

Por último, quiero agradecer a mi director Dr. Koldo Espinosa el tiempo dedicado a la tutorización y guiado de esta tesis.

Abstract

This thesis is positioned in the automatic image annotation field within the Computer Vision research area. The main target of that particular field is to give textual labels to an image in a way that they describe the content of the actual image without human intervention.

The research trend followed by this thesis is to annotate the input image using the *nearest neighbour model*. This model is based on the following concept: if two images are visually similar then the content of the images will be similar too.

In that model the image presented to the system is compared against a known image database. For the comparison, an algorithm uses multiple visual features of the images and it generates a numerical ranking of the images in the database, where the top ranked images are the most visually similar to the input image. From that ranked list of images the automatic annotation algorithm selects the top *N* images and recollects the human-annotated labels related to the *N* images from the database. In a second step the *nearest neighbour model* applies an additional algorithm which eventually chooses the labels that are likely to be present in the original image.

The foundations of this thesis are based on the previously described model. The novelty presented by this thesis exist in the actual implementation of the two steps that involve the *nearest neighbour model*. In the first step, this thesis proposes the use of the well-known MPEG7 features to describe the similarity between images. These features are divided in two categories: colour description and texture description. While the colour description feature in MPEG7 is based on the colour representation of the human eye, the texture description is a gross calculation of gradients in a grey image. In this thesis we propose a novel implementation of the standard MPEG7 that is able to mimic the macaque´s visual cortex so the description of an image´s texture is similar to the representation in the macaque´s brain while maintaining the compatibility with the standard. This has been proven to be more effective than the standard implementation and more accurate than other cortex models present in neuroscience literature.

In the second step of the *nearest neighbour model*, this thesis proposes a novel algorithm to rank the possible labels of an image. The main advance introduced by this algorithm is the combination of textual information from the labels and visual information from the images. This generates a joint visual and textual information descriptor that is able to rank all the possible labels, selected from the previous step, in an ordered list of plausible tags which ultimately will annotate the image. This algorithm has demonstrated state of the art performance in the task of label transference outperforming other methods in that category. In addition, this thesis also proposes a novel training algorithm which has the benefit of being fast and adapted to the particular annotation task so it can obtain accurate results with a small overhead in real time annotations.

Resumen

Esta tesis se posiciona en el campo de la anotación automática de imágenes dentro del área de investigación de la Visión Artificial. El principal objetivo de este campo es generar etiquetas textuales para una imagen de tal forma que describan los objetos existentes en la imagen sin intervención humana.

La tendencia seguida por esta tesis es la de anotar una imagen de entrada utilizando el modelo de *vecinos más cercanos*, el cual se basa en el concepto de que si dos imágenes son visualmente similares, entonces su contenido también será similar.

En este modelo, la imagen de entrada al sistema se compara con todas las imágenes existentes en una base de datos conocida. Para esta comparativa se utilizan múltiples características visuales de las imágenes de forma que un algoritmo establece un orden de similitud de forma que las imágenes en las primeras posiciones son las más similares a la imagen de entrada. De esta lista ordenada de imágenes, el sistema selecciona las *N* primeras imágenes y recolecta sus etiquetas anotadas por humanos de la base de datos. En un segundo paso, el modelo de *vecinos más cercanos* aplica un algoritmo adicional que decidirá qué etiquetas representan al contenido real de la imagen original.

Esta tesis se cimenta sobre el modelo descrito anteriormente y la novedad reside en la implementación de los dos pasos específicos definidos por el modelo. En el primer paso, esta tesis propone el uso de las características MPEG7 para describir la similitud entre imágenes. Estas características están divididas en descripción de color y descripción de textura. Mientras que la descripción de color en el estándar MPEG7 se basa en la representación del color del ojo humano, la descripción de la textura es un simple cálculo y acumulación de los gradientes presentes en la imagen de entrada. En esta tesis se propone una nueva implementación del descriptor de textura del estándar MPEG7 que es capaz de replicar el córtex visual de un macaco. Así, la descripción de la textura de una imagen es similar a la del cerebro de un macaco y totalmente compatible con el estándar. Se ha comprobado como este algoritmo es más efectivo que la implementación propuesta por el estándar pero también es más preciso que otros modelos de córtex presentes en la literatura de neurociencia.

En el segundo paso del modelo, esta tesis propone un nuevo algoritmo para seleccionar las posibles etiquetas de una imagen. La principal ventaja introducida por este algoritmo es la combinación de información textual de las etiquetas e información visual de las imágenes. Esta descripción conjunta permite ordenar todas las posibles etiquetas obtenidas anteriormente y, finalmente, decidir cuáles de ellas son las que pertenecen a la imagen final. Este algoritmo ha mejorado los resultados de otros métodos en dicha categoría. Adicionalmente, esta tesis también propone un nuevo algoritmo de entrenamiento que tiene el beneficio de ser rápido y adaptado a la tarea de anotación particular. Este algoritmo obtiene resultados precisos con un minúsculo coste computacional de forma que permite su uso para el entrenamiento de anotaciones en tiempo real.

0. Indice

I. Introduction

I.1 Motivation and Trajectory

The turn of the XXI century came with an explosion of multimedia content in the world. Digitalization of contents became a reality and it began with music digitalization when the standard technology changed from analog cassettes to digital Compact Discs. Still photography was also digitalized and personal cameras were popularized. Therefore, the growth of personal computer sales was a fact as users needed a PC to store and manage these new digital images. In parallel, advances were made in the video camera recorders. These evolved from low resolution cameras which recorded video on integrated DVDs to FullHD video files stored in high speed and capacity memory cards.

In addition to hardware and market evolution, internet and access to the internet also evolved accordingly. In 2000, 56kpbs modems were the trend in home communications, until it evolved to ADSL technology or more recent Fiber–To-The-Home.

Internet services matured on this accessibility evolution. In the music area Napster appeared in 1999 and people from the entire world were able to download a song in minutes and thereby transforming the way of to consume music. Services that allowed users to store their digital images on the cloud were popularized by Flickr in 2004. Video also had its own evolution that started with the uploading of videos to personal websites and evolved to major video sharing platforms as YouTube that was born in 2005.

As a user, I personally took active part in all these changes. My family bought me my first analog camera in 1996. Then my next camera was a digital camera with a resolution of 1.3Mp (4:3) with single-lens optics which was a gift to me in 2004. Less time passed until I had the next camera, an 8Mp camera in 2007 and in 2012 I bought my latest camera with 12Mp resolution and better optics. In so far as video is concerned, my first digital camera was not able to capture video but the following ones did , and they captured video at 2Mp resolution and a 720p, respectively. Now a new revolution is under the hood. The mobile devices are even more popular than the traditional PC, and they include still photography and full HD video capabilities and additionally they are always connected.

As digital cameras became popular and taking a picture was free of charge, every time I used a camera I took hundreds of images. At that moment I was annoyed by the amount of images and the time I spent retrieving one specific image from my hard disk drive. I realized that the problem could be solved if I used a software or internet service which

allowed me to tag every single photo with my own labels. At first this idea looked right to me but then I found out that it was useless: *if I wanted to search a label that I didn't use during the image tagging process, how could I search for it?*

In my personal life I started to study Telecommunications Engineering and I realized that I liked computers to perform tasks automatically. At that moment I thought about working on Artificial Intelligence, but later, during my Final-Year-Project, I realized that this was a huge field and I started to work in the sub-field of Computer Vision or, in other words: *teaching computers to see*. During that project my advisor, Dr. Koldo Espinosa, told me that if I wanted to develop novel ways for a computer to see I would need to become a researcher because there were so many things unsolved that I would need to learn the insights of the problem, study the state of the art and propose my own methods. This led me to start working in a Research Institution and I also started my personal project: obtaining a doctorate in research.

During my first years as researcher I found that there were many unsolved problems so companies couldn't rely on many computer vision algorithms. One company approached us asking for a system to search for content in videos automatically. That presented several problems. First, video analysis was very computationally intensive and, at that time, it was not possible to build a computing cluster using commodity hardware. Video processing algorithms were not also as mature as they are now. In addition, image processing capabilities were limited: face detection and person detection were the only two major accomplishments achieved by the research community. Lot of work was also taken in detecting a small set of single object in simple images but that wasn't enough for our customer.

In late 2010, I started to think about my thesis topic and everything came to my mind. In my adolescence I had problems searching for my own images in my computer. Internet services and search engines were also unable to search images by its content. Companies needed novel algorithms to retrieve visual information but the research community was mainly focused on solving small scale problems. All this generated the idea of developing improvements in the state of the art with the goal of solving these problems and all were related to one: *retrieve images based on its content*. As I said, the idea was not to create a completely novel approach, but to improve some parts that the research community missed.

In order to start with the thesis I understood that the key problem of image retrieval was automatic image annotation. If you have an algorithm that annotates the images with the most significant words, then a text based search engine will be smart enough to retrieve the most interesting tags (a lot of research is taking place, like *query expansion*, etc).

In order to annotate images we need a couple of things: a mathematical descriptor of the images and an algorithm that performs the annotation. So I started to analyze the state of the art in these two areas and I found the basis of my thesis: the most promising approach in 2010 was based on image annotation by searching similar images and then propagating the tags from the most similar images to the query image.

This sounds simple but a lot of problems arise, as stated in a seminal work by Makadia et al [MAKA10]: you need to describe the images with a compact descriptor, you need to perform near real time similarity searches on a big database of images and you need to transfer the tags. Several methods were proposed on each part and I analyzed most of them, which led me to have a deep knowledge about the state of the art and allowed me to write several review papers and invitations to symposia. This generated one of the main pillars of this thesis.

Regarding the step of image description, we found that if you describe your images using MPEG7 based descriptors, you can yield the same performance as state of the art but at lower computational and storage cost. But the MPEG7 descriptors were peculiar. The Scalable Color Descriptor was based on human perception of color, but the Edge Histogram Descriptor was based on man-made texture filters. At that time I asked myself: *Why didn't they use a Human Based texture descriptor?* I found that it was not an easy solution as a valid cortex model for this purpose did not exist. Then I asked myself again: *Could we achieve better results using a bio-inspired texture descriptor?* At first, we didn't know so we started to work in this research line taking the foundations of the retinal and color processing from my mentor Dr. Estibaliz Garrote, which become the second main pillar of this thesis.

During my analysis of the state of the art, I also found that state of the art approaches to transfer tags were complex but they didn't include any visual information at all. In this case, I also had a question: *If the research community is working with images, why did they not use image features to propagate the tags?* That question led to me to contact

Prof. Lorenzo Torresani, leader of the Visual Learning Group at Dartmouth College (USA), and to explore during 4 months a novel technology to combine textual and visual data in order to generate rankings. With such knowledge as the inception, we started to work in a novel approach to validate if using the textual and the visual features during the tag transference stage achieves better results than using only textual features. This defined the third pillar of this thesis.

I.2 Introduction to the field

Nowadays we are living in a digital world, where the number of sensors, electronic devices and users increase every second. In 2014 more internet connected mobile phones are expected than computers are expected [MART13]. Every single digital process or every information exchange produces huge amounts of data which is added to existing data collections. This production generates several problems related to the management of such vast amounts of data. The magnitude of the problem is huge and it can be seen by the fact that the data generated during two days in 2010 is greater than the accumulated data since the origin of the civilization to 2013 [TECH10].These magnitudes scare so there are lots of efforts from the scientific community to work on solving problems generated by this *data wave*, from the bottom layer (e.g. storage of the data) to the top layer (e.g. speech recognition).

Depending on its nature, data can be classified into two main groups: structured data, which follows a model that gives meta-information about data and helps during the processing; and unstructured or raw data, which does not have any pre-defined structure or any meta-information, so it is harder to analyze. In this second group we can find multimedia data and specifically images.

“A picture is worth a thousand words". This old saying is a trend in the modern digital era. .Actually, images are one of the most important pieces of exchanged data. This can be seen in the huge amount of web services and mobile applications related to the topic, from the most traditional services, like Flickr or image search engines, to the most novel applications that let the user to modify the pictures or automatically process them to achieve a better aesthetics. One specific example of the increase of images on the internet in the last few years is that the number of new images uploaded to *Instagram* each day is more than 60 Million [INST14], which is 2.5 million images uploaded per hour.

Despite being one of the data type most relevant to users, search engines are unable to handle this data correctly [RODR14]. This means that even when users are uploading their photos to the cloud, there is no way that a user can retrieve images in a user friendly system. Most services allow users to incorporate textual labels or tags, so in order to retrieve one image users can use a search engine and query it with the tag they desire. The problem occurs when a user doesn't add any label, which is very common as users can upload several hundred photos and tag only some of them with sparse tags.

The solution to this problem was proposed in early 1990. It consisted on an automatic algorithm that analyzed the content of the images and assigned labels related to the content. The concept is clear but the implementation is not so obvious. Computer vision algorithms are needed to analyze the content of the images, but these algorithms are not yet mature enough to extract information from all the objects around the world.

Most of the algorithms proposed by the computer vision community are related to the fields of statistics, mathematics and machine learning as a whole. This means that they are based on empirical measurements of the current world, so they need huge amounts of information to model all the possible objects in the world. But human beings' brain contains complex architectures that are able to learn this kind of knowledge with less information, so using bio-inspired models to process real world data is a clear advantage [GARR11].

Until now, lots of works have been carried out in the field of automatic image annotation and most of them have improved the state of the art. Now, some of these algorithms are spreading across the industry and future opportunities are appearing .Searching of multiple objects in a single image is still a challenge, and video analysis in real time, detecting actions, emotions, concepts,… will be one of the key future research lines.

I.3 Aim of the Thesis

This thesis is related to the automatic image annotation task. The main idea of this job is how to generate the labels that most likely describe the content of a particular photo.

To this end, multiple approaches have been proposed and are analyzed in chapters II and III. This analysis has led to the conclusion that a Nearest Neighbor based model is the main baseline to be considered.

Considering that baseline, the aim of this thesis is to propose:

- A **novel mathematical description** of the content of a natural image based on an animal's visual system.
- A **novel tag transference algorithm** that relies on visual and textual information to transfer the most relevant tags from a set of possible tags.

Therefore, in order to fulfil the aim, the following elements must be looked at:

- Study of the state of the art: the automatic image annotation field involves lots of technologies from the computer vision community, multimedia community and also from other related fields like machine learning. An in-depth study of all the technologies must be performed, but also a detailed study of the image annotation field, as lots of research groups are participating as it is one of the most extensively researched fields.
- Study of color and texture descriptors: Edges and color are two of the elements that compose an image, so it is clear that low level visual descriptors that analyze such information need to be carefully looked at.
- Study of the physiological component involved in the process of perception and processing of color and texture information in an animal's visual system.
- Modelling the relevant components of the processing chain of an animal's visual system.
- Experiments running and testing of the output obtained by the models. These models must be compared with biological responses of an animal's visual system but also with other proposed bio-inspired approaches.
- Modelling and evaluation of machine learning techniques to combine textual and visual information.

I.4 Content of the Thesis

This thesis is divided into 7 chapters that provide a detailed description of the different areas included in this work.

Chapter I is devoted to the personal motivation of this work and the introduction to the topic. It also presents the objectives to be achieved and the steps to be performed.

Chapter II provides a general framework of the knowledge in which to develop this research. General information on base technologies like visual descriptors and similarity

search techniques is provided. A detailed perspective of the field of image annotation is also presented and divided into the main three research lines, which are generative models, discriminative models and nearest neighbor based models.

Chapter III is dedicated to an in-depth analysis of the main trends in the state of the art presented in the previous chapter. To this end, a common methodology composed by common image databases and common validation metrics are proposed and followed thorough the thesis. As the result of such analysis in Chapter III a baseline that represents the current state of the art is proposed, so in the rest of the thesis all the experiments are done comparing the proposals against the baseline.

Chapter IV is focused on how texture information is extracted and represented in a primate's brain. This chapter studies the different parts of the visual system and specially looks at the study of the primary visual cortex, its neural structure and functionality, as it is in charge of basic texture information extraction and representation. It also presents the existing functional and computational models of the cortex, and it shows the proposal of several models which mimic the real effects generated in a macaque's visual cortex.

Chapter V presents how the proposed cortex models can be used to make a new implementation of the MPEG7 standard, and more particularly the Edge Histogram Descriptor. A detailed proposal of integration is exposed and then a fine tunning of the biological parameters is performed leading to a visual descriptor that achieves better results than the standard MPEG7 and the state of the art.

Chapter VI looks at a different point in the image annotation chain, and more particularly it is focused in the label transference models. In this chapter a novel tag transference algorithm is proposed which uses visual information and textual information to propagate the tags. In addition a fast and adapted training algorithm is proposed, obtaining an accurate result with less overhead in real time queries.

Finally, Chapter VII sets out the final conclusions and main contributions of this thesis as well as proposals for future works.

II. Estado del Arte en anotación de imágenes

II.1 Introducción

En esta tesis se aborda la temática de la anotación automática de imágenes en escenarios en los que el volumen disponible de éstas es muy alto. Un ejemplo de este tipo son los buscadores de imágenes, los cuales trabajan con un gran número de ellas, y es necesario anotarlas de forma textual para que los motores de búsqueda sean capaces de localizarlas. Para poder realizar esta anotación, es primordial analizar las propias imágenes mediante tecnologías de Visión Artificial, de modo que se logre conocer el contenido de las mismas.

Las técnicas de Visión Artificial para el análisis de imágenes permiten obtener su representación matemática, de forma que posteriormente se interpretará esta información de bajo nivel para encontrar su correspondencia con conceptos semánticos de alto nivel. Este es el principal objetivo de los investigadores en la actualidad y se le denomina *semantic gap*.

Desde el comienzo de las investigaciones en el campo de la anotación de imágenes no estaba claro si era posible superar el *semantic gap*. Diferentes estudios del cerebro humano han investigado cómo se representan los objetos en el córtex visual [SERR07] [DICK09], pero no se podía vislumbrar cómo se realiza el cambio a una representación semántica de la imagen formada en la retina [TOUS12]. En cuanto a la posibilidad de realizar la interpretación semántica por parte de sistemas automáticos de procesamiento, no ha sido hasta 2011 cuando se ha demostrado que cuanto más visualmente similares son dos imágenes, mayor es su similitud semántica [DESE11].

La barrera del *semantic gap* es crítica a la hora de realizar la anotación de imágenes, y se ha abordado desde diferentes puntos de vista: utilizando técnicas estadísticas para mapear características de bajo nivel a vocabularios no estructurados de alto nivel [CSUR04], mediante representaciones intermedias compartidas entre varios objetos [BERG14], usando jerarquías semánticas de vocabularios limitados que aportan un mayor grado de riqueza semántica [GAO10], o mediante la utilización de técnicas complejas de procesamiento de lenguaje natural para definir no sólo el contenido de las imágenes sino las relaciones existentes entre el propio contenido [LEE10].

El contenido de la imagen es muy importante a la hora de anotarla para su posterior recuperación. Pero hay situaciones en las que la información visual no aporta todo el

contenido de la propia imagen, y es evidente que la combinación de la información visual con otra información textual es beneficiosa para los sistemas de anotación [GARC11]. Para lograr una mayor riqueza de anotación, existe una línea de investigación a la que cada vez se están dedicando más esfuerzos y será muy relevante en el futuro: la "contextualización de la imagen". En este área, se parte de la asunción de que una imagen estará localizada dentro de uno o varios documentos, por ejemplo páginas web, que tratarán sobre temas relacionados con ella. De esta forma, la contextualización de la imagen analiza los propios documentos, no sólo la imagen, y busca obtener un mayor conocimiento del contenido de la imagen además del que se pueda extraer de la propia fotografía. Por ejemplo, si se presenta en una página web una imagen del cuadro de "Las Meninas", el sistema de procesamiento visual sabrá que hay un perro, un espejo, varias personas, incluso podría llegar a saber quiénes son las personas. Pero, del texto que rodea la imagen se puede extraer información sobre quién fue el autor o cuando se pintó, pudiendo reconocer que se trata del cuadro original pintado por Velázquez o la reinterpretación de Picasso, ambos presentados en la Figura II-1.

Figura II-1: Dos versiones diferentes del cuadro de "Las Meninas", el original de Velázquez (izquierda) y la reinterpretación de Picasso (derecha)

Esta última aproximación queda fuera del alcance de esta tesis, puesto que el objetivo es analizar el propio contenido de las imágenes para realizar la anotación, y no tanto analizar todo el documento donde se encuentra la imagen. A pesar de ello, el concepto de hibridar la información visual con datos textuales se analizará en detalle.

En los siguientes apartados se verán una serie de algoritmos que definen el estado actual de la técnica de anotación automática de imágenes. En primer lugar, se describirán las

tecnologías base de los diferentes pasos necesarios para anotar una imagen, como son los descriptores visuales de imágenes y la búsqueda de los vectores matemáticos similares a un tercero. Posteriormente, se tratarán las diferentes tendencias en el campo de anotación de imágenes. Finalmente, se describirán de forma breve las relaciones que pueden existir en el futuro entre la anotación automática y el análisis textual de documentos relacionados con las imágenes.

II.2 Descriptores visuales de imagen

Un descriptor de imagen se define como una representación matemática de una imagen o vídeo en forma de ristra de números. La principal propiedad de este conjunto de números debe ser que sean lo suficientemente representativos como para contener la *información relevante* de la imagen.

Evidentemente, no existe el descriptor perfecto y en la literatura no aparece un único descriptor, principalmente porque el concepto de *información relevante* es muy dependiente de la aplicación a la que se oriente. Por ejemplo, si el objetivo de una aplicación de visión artificial es detectar una mancha de tinte verde en un proceso de fabricación de telas naranjas, la *información relevante* es el color. Por ello, ante una fotografía será necesario describir el color de la misma o de pequeñas porciones de la misma. Adicionalmente, la *información relevante* puede tener una componente espacial pues no es lo mismo que el descriptor describa una zona local de la imagen (y se le denomine *descriptor local*), o toda la imagen en su conjunto (denominándose *descriptor global*).

Otro ejemplo puede ser detectar que la impresión de un libro esté borrosa. Para ello, la utilización del color como información relevante no es una buena opción, ya que las páginas siempre tendrán color blanco y negro. Si la impresión es borrosa o defectuosa, la distribución del color podría ser la misma, lo que cambia es la textura de la imagen. Una página de un libro posee una textura determinada: fondo blanco y líneas definidas horizontales. Si la impresión no es correcta, las líneas no serán horizontales, o las líneas tendrán zonas vacías, o no existirán líneas y será borroso. Para estos casos es necesario describir la información de textura dentro del descriptor visual.

Además de la textura y el color, los descriptores visuales pueden contener información sobre las formas que aparecen en las imágenes o el movimiento en los vídeos. En estos

casos, la *información relevante* es información de bajo nivel semántico. En otros casos, la información relevante que se quiere representar de una imagen son conceptos de un mayor nivel semántico, como por ejemplo *caras* u *objetos*. En este último ejemplo, la idea de representación se basa en que a partir de una imagen que contiene un objeto, el vector que describe la imagen deberá contener información sobre el objeto que está presente, y no sólo del color o la textura de la imagen completa.

Con estos ejemplos, se intuye la importancia de los descriptores visuales en cualquier sistema de visión artificial, pero la pregunta es ¿por qué son necesarios los descriptores visuales?

Una imagen digital está formada por un conjunto de píxeles, que no son más que un agregado de números representando la cantidad de fotones que ha llegado al sensor de la cámara en un punto espacial determinado. Por construcción, una imagen no tiene información de alto nivel de la escena, sino que sólo da información de la cantidad de luz que llega a la cámara. En cualquier tarea de visión artificial, el objetivo es realizar algún tipo de reconocimiento automático de los elementos presentes en la imagen, y este reconocimiento tiene asociado una semántica proporcionada por los seres humanos, por ejemplo, reconocer a una persona. De esta forma, aparece de nuevo el denominado *semantic gap*: la imagen posee información de cantidad de luz en la cámara y el ser humano quiere información semántica de alto nivel. Para llenar este *gap*, los sistemas de visión artificial utilizan algoritmos de inteligencia artificial que permiten realizar esta traducción. Para ello sintetizan la información de los píxeles en los descriptores visuales y posteriormente infieren información semántica a partir de ellos.

Por tanto, se ve que el objetivo de un descriptor visual es el de agrupar la información de los píxeles de una imagen, dando un sentido semántico que ayude en la tarea final de la visión artificial.

Los descriptores visuales no trabajan de forma individual. En diferentes estudios se ha demostrado que la combinación de características visuales alcanza tasas de acierto mucho mayores que con el uso de características individuales [DESE08]. De esta forma, la propia combinación de las características es un área de intenso estudio.

Este apartado se centrará en mostrar en detalle diferentes descriptores visuales utilizados. Si bien no estarán todos los existentes por ser un número excesivamente

grande, sí lo harán los más utilizados en la literatura. En cuanto a la combinación de características queda fuera del alcance de esta tesis, pero se mostrarán varios métodos en los que se basa un gran número de trabajos del estado del arte.

II.2.1 Momentos de imagen

El descriptor visual más básico en esencia son los momentos de imagen. Se puede describir un momento de imagen como una media ponderada de la intensidad de los píxeles de una imagen. El objetivo de los momentos, como el de todos los descriptores de imagen, es representar con uno o varios números una propiedad que contiene la propia imagen y en este caso se representa la textura de la misma.

La definición matemática básica de *momento*, para una función continua en dos dimensiones genérica *f(x,y)* con una integral finita diferente de cero, es la siguiente:

$$M_{pq} = \int_{-\infty}^{+\infty}\int x^p y^q f(x,y)\, dx\, dy$$

donde M_{pq} es el momento de orden *p*+*q*, y donde *p* y *q* pueden ser cualquier número natural. Una imagen *I(x,y)* no es una función continua, por lo que pasando dicha función al plano discreto, la definición de *momento* queda de la siguiente forma:

$$M_{pq} = \sum_x \sum_y x^p y^q I(x,y)$$

Por ejemplo, el momento M_{00} se corresponde con el área de toda la imagen. Si el descriptor se quiere de toda la imagen y, por tanto, es un descriptor global, o de una zona única, para el caso de descriptores locales.

A esta definición se le denomina *momento geométrico*, pero como se puede ver, no es invariante ni a rotación ni a traslación. Por *invariante* se hace referencia a que si a una imagen se le aplica una transformación, en forma de rotación, traslación o escala, el descriptor debería mantenerse constante. Para lograr una invariancia a la traslación se puede usar el *momento central* [HUAN10].

$$c_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y)$$

Se denomina *momento central* por la resta del centro de la imagen, siendo:

$$\bar{x} = \frac{M_{10}}{M_{00}}$$

$$\bar{y} = \frac{M_{01}}{M_{00}}$$

Esta resta permite representar el momento M_{pq}como si estuviese desplazado al centro de la imagen. Si, independientemente de la posición de la imagen, siempre se vuelve a una representación en base a su centro, este *momento* será **invariante a traslación**.

Si además se desea lograr una invariancia a cambios en la **escala** de la imagen, entonces se puede realizar una normalización del momento central de la siguiente forma:

$$\eta_{pq} = \frac{c_{pq}}{c^{\gamma}{}_{00}}$$

donde

$$\gamma = \frac{p+q+2}{2}$$

$$p + q \geq 2$$

Basándose en esta definición, en 1962 Hu propuso siete nuevos números a partir de una combinación de estos momentos, de tal forma que se convertían en **invariantes a rotación**, además de a escala y traslación. Los siete nuevos números, también llamados momentos, son los siguientes:

$$M_1 = (\eta_{20} + \eta_{02})$$

$$M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta^2{}_{20}$$

$$M_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$M_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$M_5 = (\eta_{30} - 3\eta_{12})^2(\eta_{30} + \eta_{12}) \ \ [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$

$$M_6 = (\eta_{20} - \eta_{02}) \ \ [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{12})$$

$$+(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$M_7 = (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$

$$-(\eta_{30}+3\eta_{12})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]$$

Según su autor, los momentos del 1 al 6 son momentos totalmente invariantes a rotación, escala y traslación, mientras que el 7 no tiene invariancia ortogonal y, por tanto, diferencia imágenes “en espejo”.

Por ejemplo, ante las imágenes de la Figura II-2, los resultados obtenidos de los siete momentos se presentan en la Tabla II-1[KASY13].

Figura II-2: Silueta humana y silueta humana rotada 90º [KASY13]

Tabla II-1: Tabla de momentos invariantes de Hu [KASY13]

	M_1	M_2	M_3	M_4	M_5	M_6	M_7
Original	3.78	8.71	0.69	0.02	-0.003	-0.05	-0.006
Rotado	3.78	8.71	0.69	0.02	-0.003	-0.05	0.001

En la Tabla II-1 se comprueba que los valores de los diferentes momentos ante la rotación son exactamente iguales a excepción del momento 7, que no es invariante. A pesar de este resultado, la realidad es que estos valores no son constantes y fluctúan ante la rotación y escala, como se puede ver en momento M3 (Figura II-3) [HUAN10].

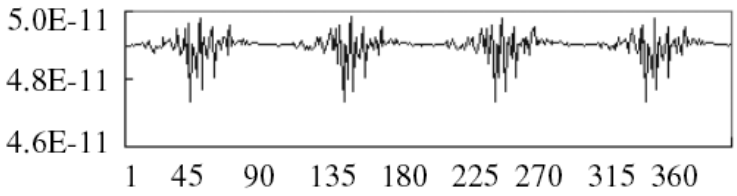


Figura II-3: Fluctuación de M3 al rotar la imagen de prueba entre 1º y 360º[HUAN10]

Además de estos momentos, existen diferentes valores que permiten robustecer los anteriores. Uno de los más utilizados en el estado del arte son los *momentos Zernike*. Un momento Zernike de dos dimensiones de orden *n* con repetición *m* sobre una imagen $I(\rho,\theta)$, definido en coordenadas polares es el siguiente:

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} I(\rho,\theta) V_{mn}(\rho,\theta)$$

donde:

$$\rho \leq 1$$

$$\rho = \sqrt{x^2 + y^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

V_{mn} es una serie de polinomios complejos (polinomios de *Zernike*) definidos dentro del círculo unidad mediante la siguiente fórmula:

$$V_{mn}(\rho,\theta) = R_{mn}(\rho) e^{(jm\theta)}$$

donde:

$$R_{mn}(\rho) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s (n-s)!}{s!\left(\frac{n+m}{2}-s\right)!\left(\frac{n-m}{2}-s\right)!}$$

Los polinomios se pueden ver en la Figura II-4 y se ve necesario el definir un orden de los mismos. Los momentos de orden más bajo contienen información bruta de la forma de los objetos, mientras que los momentos de mayor orden contienen información más

detallada. Un ejemplo es la utilización de 30 valores para representar una imagen, logrando un alto grado de detalle [KASY13].

La principal propiedad de estos momentos es que la magnitud de los momentos de *Zernike* es totalmente invariante a rotación [KHOT90]. Si se quiere hacer estos momentos invariantes a escala y traslación, como los de Hu, es necesario aplicar una normalización específica [KASY13].

Para obtener una invariancia a la traslación se debe modificar la imagen $I(x,y)$ por una trasladada a su centroide $I(x-\bar{x}, y-\bar{y})$. En cuanto a la invariancia a la escala, se puede hacer la siguiente transformación de la imagen:

$$I(x,y) \rightarrow \ I(\alpha x, \alpha y)$$

donde: $\alpha = \sqrt{\frac{\beta}{m_{00}}}$, de tal forma que *beta* es un valor predeterminado.

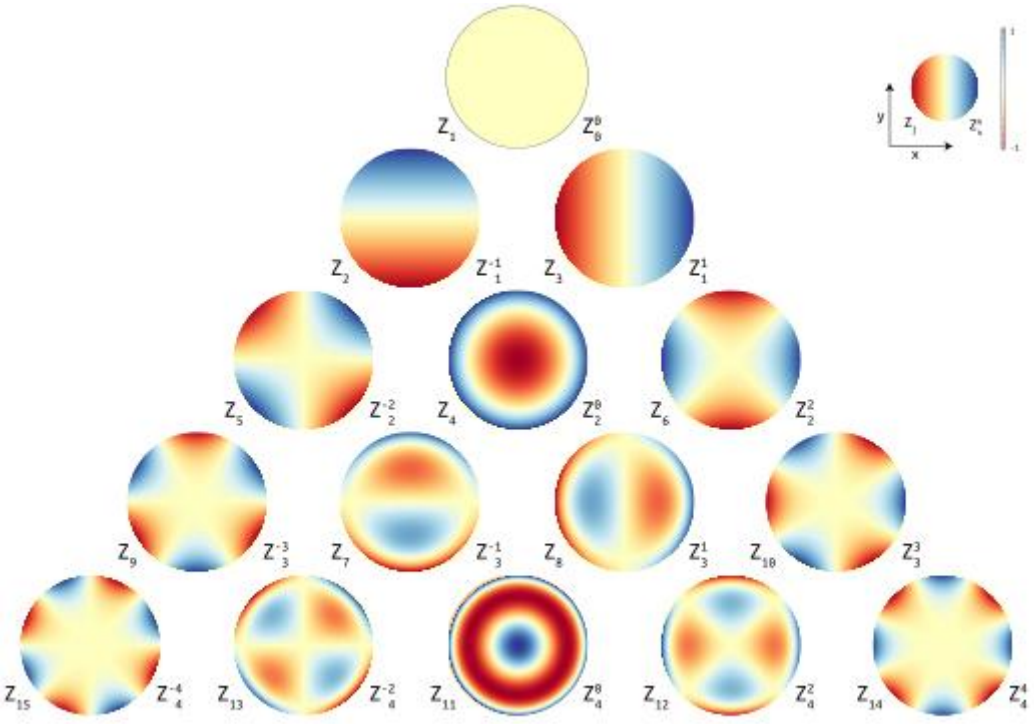


Figura II-4: 15 polinomios de Zernike [ZERN14]

Finalmente, un descriptor visual basado en *momentos* (geométricos, de *Hu* o de *Zernike*) sería un vector que en cada posición almacenaría uno de los valores definidos.

II.2.2 Histograma de color

En el punto anterior se ha visto uno de los descriptores de textura más básico, que es el de los *momentos* de imagen. En ellos, se ha visto cómo se calculan una serie de números y se aglutinan en un único descriptor visual.

Otro de los descriptores más básicos es el histograma. Un histograma computa la frecuencia de repetición de cada uno de sus valores dentro de un conjunto de datos. La forma más simple es, ante una imagen 2D en la que cada píxel representa un nivel de gris, computar el histograma con varios niveles que corresponden a diferentes valores de gris. Este descriptor dará una representación genérica de la textura de la imagen, ya que muestra la distribución de la intensidad. Por ejemplo, en la Figura II-5 se ve un histograma de una imagen dividida en 250 niveles de gris. Se comprueba la presencia de muchos niveles oscuros (valores de intensidad menores de 100), por lo que se trata de una imagen oscura.

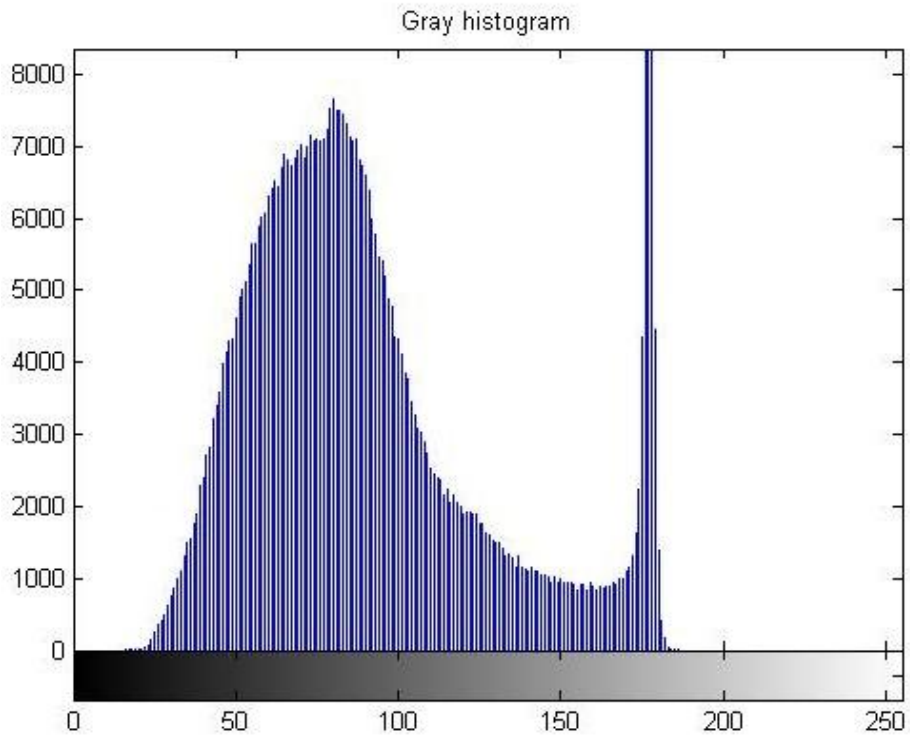


Figura II-5: Histograma de los niveles de gris de una imagen.

Otra opción para representar el color de forma básica es computar un histograma de color de la imagen. Para ello, cada valor del histograma será una discretización del conjunto de colores posibles representados como 3 números, en función del espacio de color utilizado, por ejemplo *Red Green Blue* (RGB) o *Hue Saturation Value* (HSV).

Un caso particularmente especial bastante utilizado en la actualidad son los histogramas basados en el espacio de color *oponente* [VAN10]. El histograma final de la imagen se compondrá de tres histogramas de una dimensión concatenados, donde cada dimensión se representa con las siguientes coordenadas del *espacio oponente*:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$$

Estas coordenadas representan de forma gruesa la representación del color en el sistema visual humano, de tal forma que la dimensión O_3 representa la información de *intensidad*, y la O_1 y O_2 la información de *color*.

Este tipo de histogramas entran dentro de la categoría de descriptores visuales globales, ya que representan a toda la imagen en su conjunto.

II.2.3 Filtros de Gabor

Hasta ahora se han mostrado una serie de descriptores básicos que permiten comprender la funcionalidad de los descriptores de imagen. A pesar de ser descriptores simples, son unos de los más utilizados por su simplicidad de computación y de resultados obtenidos.

Dentro del conjunto de descriptores que permiten describir una textura se encuentran los filtros de Gabor. Estos filtros modelan la forma de las neuronas simples del córtex visual primario [SERR07], por lo que gozan de una gran popularidad.

Un filtro de Gabor se puede definir como una señal portadora sinusoidal modulada por una señal gaussiana. Este filtrado en una dimensión se ilustra en la Figura II-6.

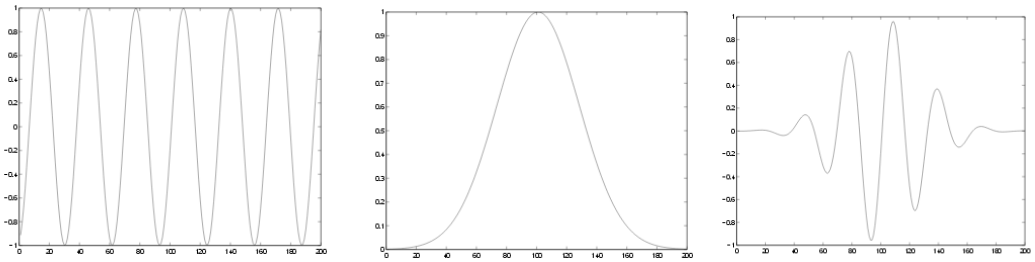


Figura II-6: Función gráfica de Gabor en 1 dimensión. Izquierda: Señal portadora sinusoidal. Centro: Señal moduladora gaussiana. Derecha: Señal de Gabor multiplicando las dos anteriores

De forma matemática, los filtros de Gabor poseen una componente real y otra imaginaria. Esta es la definición matemática de estos filtros en dos dimensiones *(x,y)*:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

siendo la parte real:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

siendo la parte imaginaria:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * sen\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

donde:

$$x' = x\, cos\theta + y\, sen\theta$$

$$y' = -x\, sen\theta + y\, cos\theta$$

Sabiendo que $\frac{2\pi}{\lambda}$ es la frecuencia, θ la orientación del filtro, Ψ el desfase, σ la anchura de la gaussiana envolvente y γ añade elipcidad al filtro.

En la Figura II-7 se muestran tres filtros de Gabor con diferentes parámetros de configuración en dos y tres dimensiones. El filtro superior y el último poseen la misma

frecuencia pero diferente rotación, mientras que el central posee diferente frecuencia pero igual rotación que el primero.

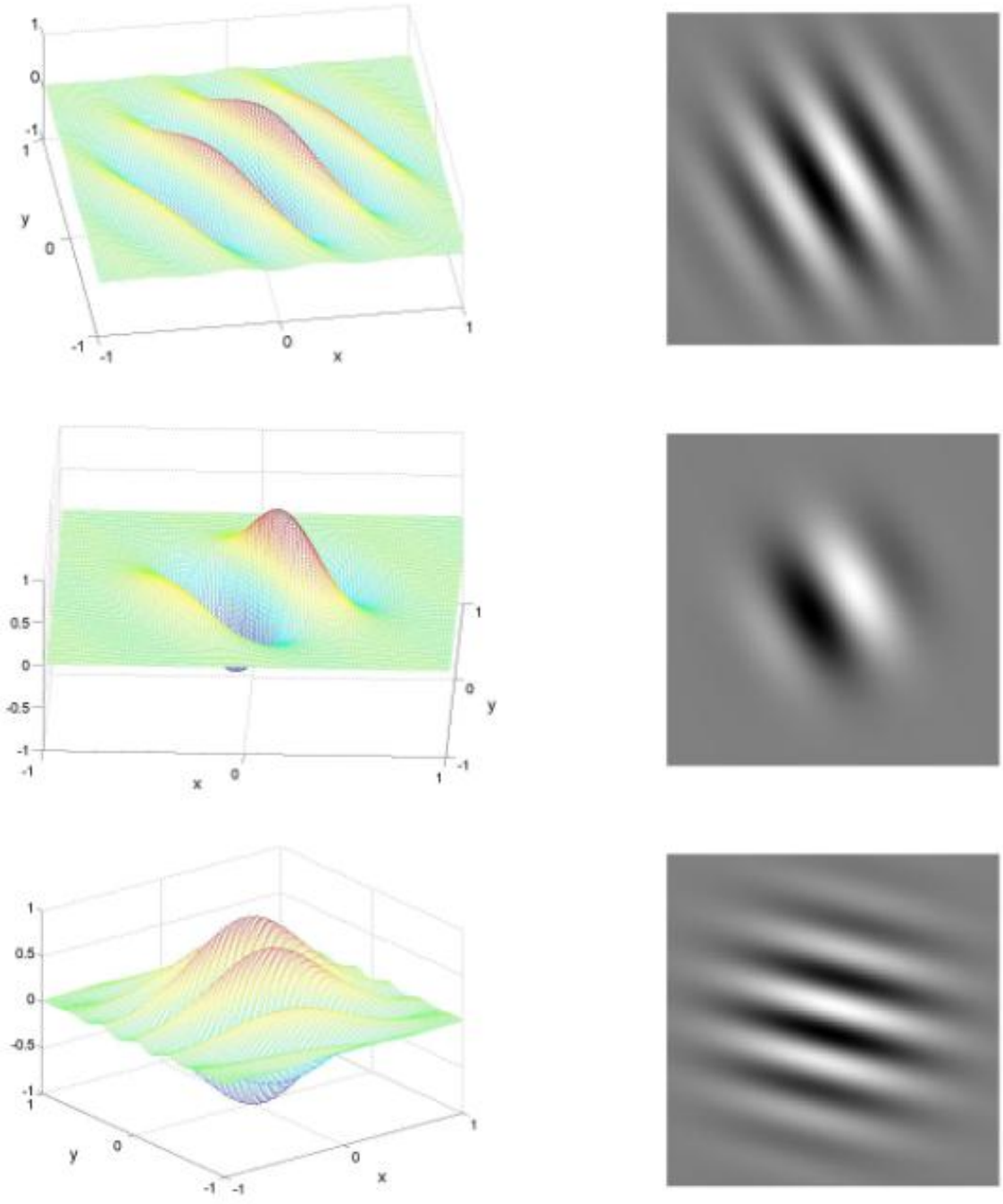


Figura II-7: Representación en 3D y su proyección en 2 dimensiones de tres filtros de Gabor con configuraciones diferentes

En cuanto a la modelización de las neuronas del cortex, en [POLL81] se muestra cómo los pares de células adyacentes en el córtex están en cuadratura, por lo que cada par de neuronas se podría poner como la parte real e imaginaria del filtro de Gabor y trabajar con dicho filtro complejo. Aun así, años más tarde en [JONE87] comprobaron cómo la parte real del filtro de Gabor es la que representa mejor la forma del campo receptivo de las neuronas simples del córtex visual del gato.

A partir de la respuesta de los filtros de Gabor se puede construir un descriptor de imagen que represente la textura de forma bioinspirada. En la literatura aparecen varias aproximaciones a la hora de generar un descriptor de imagen basado en los filtros de Gabor [DESE08]. En general todas ellas se basan en un *banco de filtros*, que se define como un conjunto de filtros Gabor con diferentes parámetros de configuración

Al usar un banco de filtros de Gabor, el descriptor visual más común es el basado en la media y la varianza de la respuesta. Así, ante una imagen de entrada se aplican todos los filtros del banco en cada píxel, y para cada respuesta de los filtros en 2 dimensiones se computan la media y la varianza, concatenando todos los valores para generar el vector de características final. Por ejemplo, para el caso de un banco de filtros de 5 frecuencias y 5 direcciones se obtiene un descriptor de 50 valores (5x5x2), ya que se obtiene la media y la varianza para cada uno de los 25 filtros.

Es evidente que este descriptor dependerá directamente de la rotación de la imagen. Si existe rotación, la respuesta máxima ya no se generará en el mismo filtro del banco, sino que se generará en otro, y el descriptor, que es la concatenación simple de las respuestas, será diferente. Para robustecer este tipo de descriptores existen numerosas aproximaciones, y una de las más sencillas y efectivas es utilizar la *Transformada Discreta del Coseno* de las respuestas [ZHI10].

Para ello, todas las medias de las respuestas se disponen en una matriz rectangular en función de su orientación y escala. Sobre esa matriz se computa la *Transformada Discreta de Coseno,* que es variante a la rotación y sus coeficientes serán los valores a concatenar en el descriptor final, en vez de las medias y varianzas.

Finalmente, la principal recomendación existente a la hora de analizar texturas es probar primero con los bancos de filtros Gabor [RAND99], por ello son estos filtros los más comunes en cuanto a la descripción de textura. A pesar de ello, también es cierto que en

los últimos años se comienza a cuestionar la necesidad de usar bancos de filtros [VAR03] en favor de volver al uso de los niveles de gris puros.

II.2.4 Haar Histogram

Además de los filtros de Gabor, idealmente se podría utilizar cualquier otro filtro para computar su salida y acumularlo en un descriptor de textura. Un grado intermedio entre la complejidad de un filtro de Gabor y la simplicidad de los niveles de intensidad de los píxeles son las *características de Haar*. Estas se derivan de un conjunto de filtros discretos y finitos, que se pueden aplicar de la misma forma que un banco de filtros de Gabor: computar su salida y luego acumularla en un histograma [MAKA10]. La principal diferencia con respecto a Gabor es que sus salidas no son complejas, por lo que el descriptor generado será más compacto.

El wavelet de Haar en una dimension fue propuesto en 1909 por Alfrèd Haar, y se puede ver en la Figura II-8.

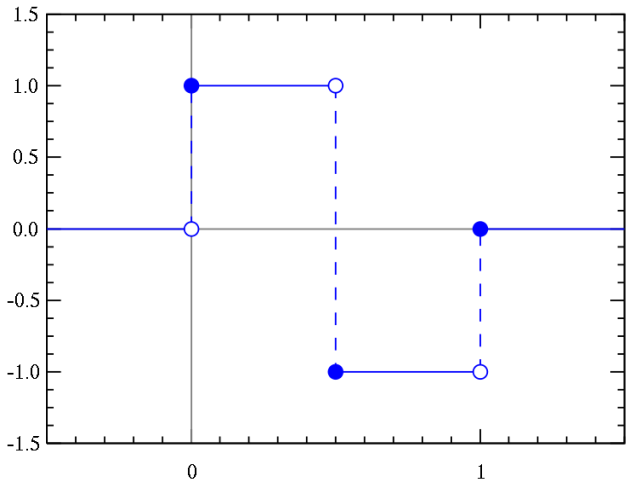


Figura II-8: Wavelet de Haar en una dimension [HAAR14]

Este wavelet se propone como una función no diferenciable, de la siguiente forma:

$$\psi(t) = \begin{cases} 1 & 0 \le t < 1/2, \\ -1 & 1/2 \le t < 1, \\ 0 & en\ otro\ caso \end{cases}$$

Si se extrapola este wavelet a dos dimensiones se obtienen los filtros de Haar, que permiten ser aplicados a las imágenes. Como se puede intuir y ver en la Figura II-9, existen numerosas formas de aplicar estos filtros, los cuales se usaron por primera vez para la detección facial [VIOL04].

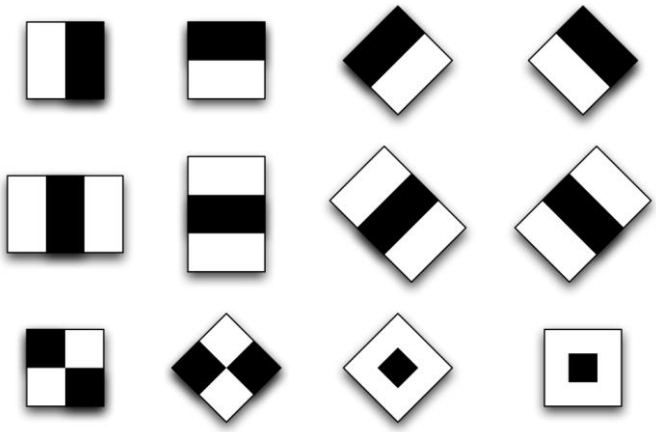


Figura II-9: Filtros de Haar 2D [BERG08]

En estos filtros, la sección negra indica un valor de -1 y la sección blanca un valor de +1, mientras que su extensión puede variar en función del objetivo de la aplicación. De esta forma ante una imagen, este filtrado se aplicará mediante ventana deslizante, o usando una optimización basada en imágenes integrales [VIOL04], y cada filtro generará un valor diferente que se acumulará en el descriptor final. En su modo más compacto, el histograma sólo tendrá dos valores que representarán el signo de la salida [MAKA10].

II.2.5 Histograma de Gradientes Orientados

Siguiendo con los histogramas como descriptores visuales, se encuentra el denominado Histograma de Gradientes Orientados (Histogram of Oriented Gradients, comúnmente HOG). En esencia, HOG computa el gradiente de la intensidad de una sub-imagen para representarlo en un histograma, de forma que dicho histograma contendrá información del contorno, de la silueta, así como cierta información de textura. Por ello, este descriptor tiene un nivel semántico mayor a otros descriptores que se han descrito anteriormente basados en histogramas, ya que no recopila estadísticas de la imagen sino que dichos números tienen una relación con la variación del nivel de gris en los bordes de los

elementos presentes en la escena. Es más, este descriptor se creó para la detección de personas con un clasificador simple [DALAL05], y posteriormente se ha utilizado para detectar todo tipo de objetos [FELZ10].

Para describir una región de la imagen, el primer paso es normalizar la luminosidad o gamma de la misma, evitando, en la medida de lo posible, los defectos en la iluminación. A continuación, para dicha región se computan los gradientes de la imagen.

Gracias a estos gradientes, aparece información semántica importante, como es la silueta y cierta información de textura; aparte de que los valores de los gradientes son resistentes a las variaciones de la iluminación.

El siguiente paso se produce a nivel de *celdas* de la sub-imagen (ver Figura II-10). El objetivo de este paso es codificar la información de los gradientes, de forma que contenga información local de la imagen pero que sea resistente a cambios pequeños de apariencia. Para ello, se usan histogramas de la orientación de los gradientes. Para cada celda de la sub-imagen, se genera un histograma de las orientaciones del gradiente en cada uno de los píxeles de la celda. En este histograma, todas las posibles orientaciones se dividen en un número definido de niveles, que constituirá una de las variables de configuración del descriptor. Así, la orientación del gradiente computado en la fase anterior determinará el nivel del histograma al que agregarse, y su magnitud determinará la cantidad que aportará en ese nivel.

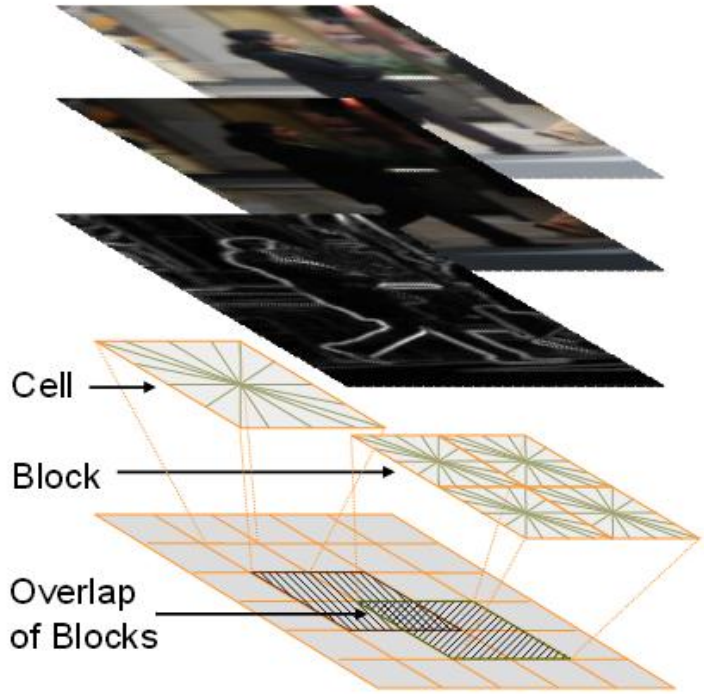


Figura II-10: Pasos para computar el descriptor HOG [DALAL12]

Tras la computación de los histogramas por celda, se procede a realizar un paso de normalización de dichos histogramas. Para ello, se seleccionan grupos de celdas llamados *bloques* y se normaliza el contraste de los histogramas de las mismas, usando la energía acumulada de todas las celdas del bloque. Esta normalización mejora la invariancia a la iluminación, a las sombras y a las variaciones en el contraste de los bordes. Como se ve en la Figura II-10, cada celda se comparte en varios bloques, por lo que para cada bloque se generará y usará una normalización específica, haciendo que la celda aparezca varias veces con diferentes normalizaciones en el descriptor final y, por tanto, mejorando su funcionamiento.

Finalmente, la acumulación de estos histogramas para todos los bloques de una imagen genera el descriptor HOG final, con un tamaño del descriptor definido así:

```
size_hog_features = (size_t)nbins * ( blockSize.width/cellSize.width)
                        * (blockSize.height/cellSize.height) * ((winSize.width
                        - blockSize.width)/blockStride.width + 1)
                        * ((winSize.height - blockSize.height)
                        / blockStride.height + 1);
```

La computación de este descriptor es más compleja que los bancos de filtros o histogramas de color y además, posee varios pasos para robustecerlo frente a cambios de iluminación así como para incluir información semántica relevante de formas y bordes. Por eso, no es de extrañar el uso de dicho descriptor en muchos de los mejores sistemas de detección de objetos del estado del arte [FELZ10].

Asimismo, tampoco es rara la aparición de diferentes modificaciones que hacen que este descriptor sea aún más robusto. Por ejemplo, el descriptor PHOG (Pyramid Histogram of Oriented Gradients) es uno de los casos más diferentes y que mejor resultado ha obtenido [BOSC07]. En ese trabajo, sus autores no modifican HOG, sino que lo utilizan para crear su propio descriptor. En un primer paso, este descriptor detecta los bordes de la imagen usando un detector de Canny y navega a través de ellos computando un histograma de orientaciones del gradiente sólo en los bordes. Además, esto lo computa para diferentes niveles de sub-imágenes (de forma piramidal), y todos los histogramas de estos niveles y sub-imágenes se concatenan para generar el descriptor PHOG final.

II.2.6 MPEG7

Los descriptores anteriores son algoritmos propuestos y utilizados en la literatura de visión artificial, pero no tan implantados en la industria multimedia. Debido a su importancia, en este último campo también se ha trabajado para obtener descriptores que sirvan para un amplio abanico de aplicaciones industriales, generando varios estándares y propuestas, entre los que el más relevante es el estándar MPEG7.

El estándar MPEG7 se estandariza en 2002 como ISO/IEC 15938 [ISO02], como complemento a otros estándares de codificación multimedia del grupo MPEG. Estándares como MPEG4 o MPEG2 se centran en protocolizar la codificación de ficheros de vídeo y audio para su reproducción en cualquier dispositivo que cumpla con dicho estándar. En cambio, MPEG7 se creó como referencia para describir el propio contenido de los documentos codificados, con el objetivo de permitir realizar búsquedas eficientes en base a dicho contenido. Por tanto, se puede definir a MPEG7 como un estándar de descripción de contenido multimedia basado en XML, que favorece la interoperabilidad de los sistemas de búsqueda multimedia.

El estándar consiste en 12 partes, cada una de las cuales cubre un aspecto determinado de toda la especificación, yendo desde la parte de descripción del contenido Visual hasta la parte de especificación de las consultas a dichos descriptores.

En la especificación, el estándar define cuatro herramientas:

- Los esquemas de descripción (Description Schemes – DS)
- Los descriptores (D)
- Un lenguaje de descripción de los esquemas (Description Definition Language - DDL).
- Esquemas, sistemas de interconexión y herramientas para codificar las descripciones de forma binaria, sincronización con otros flujos MPEG y el almacenamiento de los descriptores.

Para el objetivo de esta tesis, el punto más importante son los descriptores. Para más información sobre el estándar MPEG7 se invita al lector a leer este resumen del estándar [CHANG01].

Dentro de MPEG7 el concepto descriptor se refiere a una representación de una característica definida sintácticamente y semánticamente, por lo que un único objeto se podrá describir por varios descriptores.

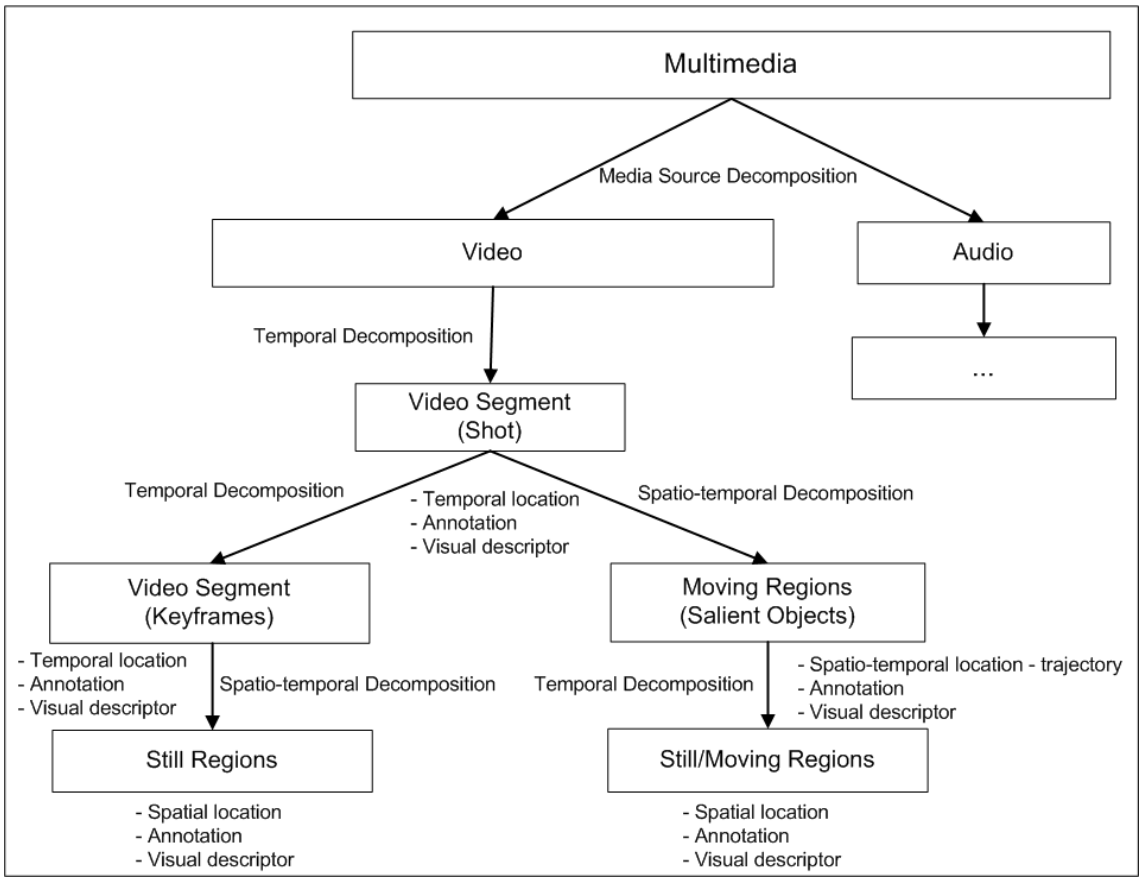


Figura II-11: Perfil de descomposición de descriptores compatible con MPEG7 [BILV14]

En concreto, este apartado II.2.6 va a tratar sólo como la parte visual de MPEG7 (Parte 3 – MPEG7 Visual). En esta parte se definen tanto los descriptores visuales como los esquemas de descripción.

Los descriptores visuales se enfocan en una característica visual (color, textura, forma o movimiento). Por otra parte, los esquemas son elementos de soporte de los anteriores, y definen estructuras (*Grid layout* y coordenadas espaciales), punto de vista, localización o información temporal.

Son varios los descriptores que se especifican en cada una de las categorías, y todos ellos poseen el objetivo de ser computacionalmente ligeros de obtener y de comparar, optimizando el espacio en memoria de las características extraídas [DESE08].

En cuanto a los descriptores de color, MPEG7 detalla los siguientes [OHM01]:

- **Dominant Color Descriptor**: Define un conjunto de colores dominantes en la imagen, así como sus propiedades estadísticas como la distribución y varianza. Su

propósito es proveer una representación de los colores presentes en una imagen (o en una región) lo más compacta, efectiva e intuitiva posible. Así, este descriptor permitirá hacer búsquedas por un color determinado de forma rápida.

- **Scalable Color Descriptor (SCD)**: Se basa en un histograma de color en el espacio de color HSV. Usa la transformada Haar sobre los coeficientes del histograma para codificarlo, permitiendo una representación escalable, pero también permitiendo una escalabilidad en la complejidad de extracción y *matching* del descriptor. Su objetivo es representar la distribución de color de toda la imagen en general, por lo que permitirá hacer búsquedas por similitud o clasificación.
- **Group of Frames/Group of Pictures Descriptor**: Es una extensión del descriptor SCD a un grupo de imágenes que componen un vídeo o una colección de imágenes. Se basa en agregar las propiedades de color de cada imagen.
- **Color Structure Descriptor**: Se basa en histogramas de color, buscando identificar distribuciones de color en zonas concretas de la imagen mediante una operación de ventana estructurada. Para garantizar la interoperabilidad de este descriptor se usa un espacio de color propio de MPEG7 (Hue-Min-Max-Difference – HMMD)
- **Color Layout Descriptor**: Captura la disposición espacial de los colores dominantes en una imagen. La representación de esta disposición se basa en los coeficientes de la *Transformada Discreta del Coseno*. Este es un descriptor muy compacto, que a su vez es muy eficiente para la búsqueda de imágenes, sobre todo para hacer una búsqueda gruesa por similitud, ya que no captura pequeños detalles. Por representar los colores a *groso modo*, este tipo de descriptor es muy útil para las búsquedas tipo *query-by-sketch*, donde se realizan trazos a grandes rasgos de la imagen.
- **Color Temperature descriptor**: Describe la temperatura perceptual sentida de una imagen. Su objetivo es permitir una búsqueda y navegación de imágenes en base a la percepción de la temperatura del usuario (imagen cálida o fría, por ejemplo).

Además de descriptores de color, MPEG7 define descriptores de textura, como:

- **Homogeneous Texture Descriptor**: Caracteriza las propiedades de textura de una imagen basándose en que la textura es homogénea y, por tanto, que las propiedades de la misma no cambian mucho en una región. Para su descripción se

utiliza un banco de filtros Gabor, que permiten extraer un gran número de características.

- **Texture Browsing Descriptor**: Es una herramienta que permite al anterior descriptor generalizar y realizar una solución más escalable para la búsqueda de regiones en imágenes.
- **Edge Histogram Descriptor (EHD)**: A diferencia de Homogeneous Texture Descriptor, EHD se centra en describir la textura en base a los bordes de la misma, para lo que define cinco tipos de bordes (cuatro direccionales y uno no direccional). Consiste entonces en una concatenación de histogramas parciales de este tipo de bordes en regiones de la imagen.

Además de los dos anteriores, el estándar también define descriptores de forma (Region Shape Descriptor, Contour Shape Descriptor, Shape 3D Descriptor), descriptores de movimiento (Camera Motion Descriptor, Motion Trajectory Descriptor, Parametric Motion Descriptor, Motion Activity Descriptor) además de otros descriptores de más alto nivel como el Face Recognition Descriptor.

No todos los descriptores mencionados son útiles para el objetivo de descripción de imagen para su posterior anotación. Para este objetivo, la literatura deja claro que se necesitan descriptores de color y textura [MAKA10] que tengan la suficiente información útil en ellos. Así, por ejemplo, el Dominant Color Descriptor no es válido, ya que sólo almacena información sobre los colores dominantes.

A continuación, se describen en detalle los dos descriptores más relevantes del estándar MPEG7 en relación a esta tesis.

II.2.6.1 MPEG7 – Scalable Color Descriptor

Uno de los descriptores de color definidos en el estándar es el Scalable Color Descriptor (SCD). El objetivo de este descriptor es generar un histograma de color global de la imagen, generando una representación que sea escalable en cuanto al número de niveles del histograma, y que sea preciso en relación al número de bits con los que se representa dicho histograma [DESE08].

El descriptor SCD se basa en el espacio de color HS, representado en la Figura II-12. Éste es un espacio de color perceptual, que emula la percepción humana de los colores y modela cada color en base a su Tono (Hue), Saturación y Valor. El primer número (H)

indica el color puro del que se trata, con un número del 0 al 360, medido en grados, donde por ejemplo 0º es el color rojo y el verde es 120º. El segundo valor (S) muestra cuan blanco es el color, indicando con un 0 la mínima saturación, por tanto la ausencia de pigmentación y, por ello, el color será blanco, y con un 1 la saturación total. Finalmente, el tercer valor (V) se denomina luminosidad e indica cuán negro es el color, así con V=0 el color es negro, disminuyendo hasta que V se hace 1.

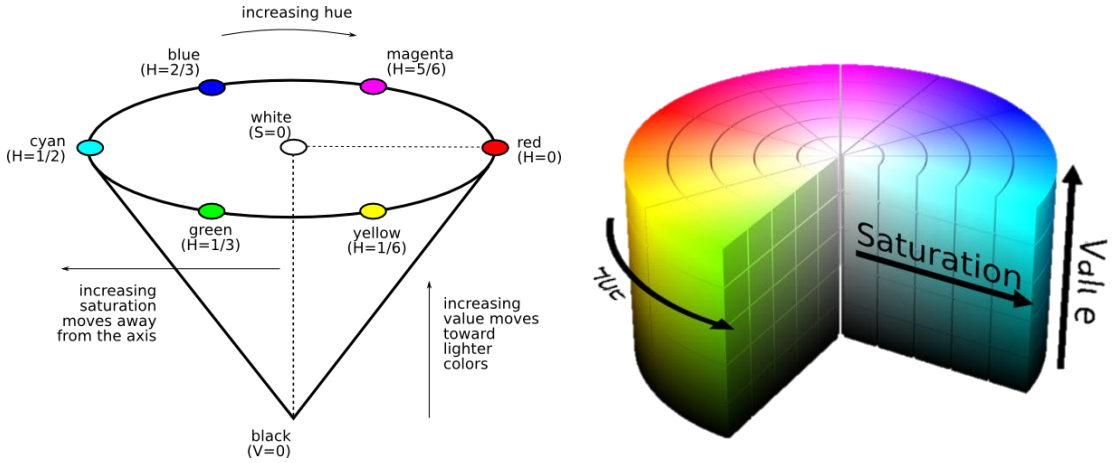


Figura II-12: Representaciones del espacio de color HSV [NEWM14] [HSL14]

Con este espacio de color como base, el descriptor SCD computa el histograma de color de la imagen en HSV y lo codifica con la transformada Haar, mediante los pasos indicados en la Figura II-13.

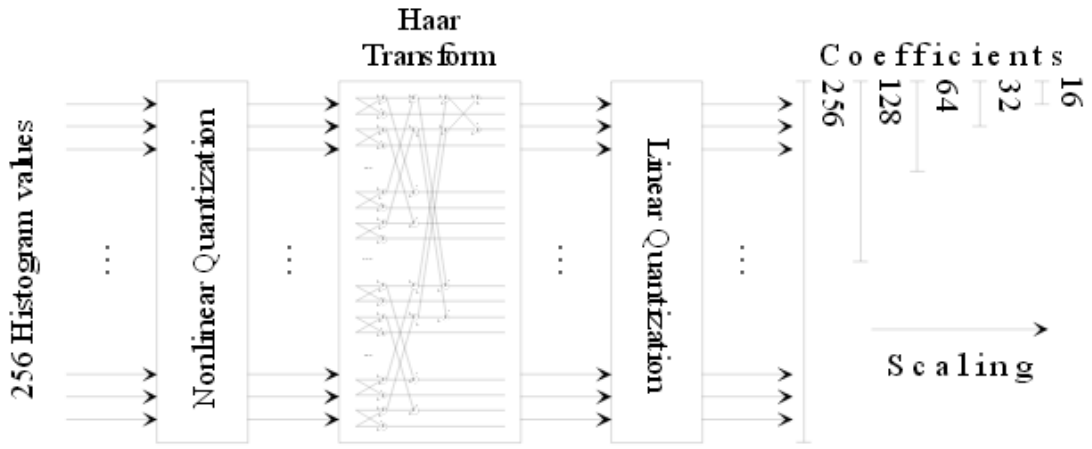


Figura II-13: Diagrama esquemático de los pasos a dar para extraer el descriptor SCD [OHM01]

Para computar el histograma, primero se cuantiza el espacio HSV de forma uniforme en 256 valores. 16 de esos niveles se asignan al valor de H, 4 niveles se asignan a S y otros 4 a V; y con esta cuantización se computa el histograma de la imagen. Para lograr una representación compacta, se cuantiza el espacio de forma no uniforme, y cada nivel del histograma se representa con un entero de 3 bits, dándole más significancia a los valores pequeños que a los grandes.

Es en este momento cuando se aplica la transformada Haar de dos formas diferentes: en modo suma o en modo diferencia.

En su forma más básica, la transformada Haar se puede aplicar sumando los pares de niveles consecutivos en el histograma de 256 niveles. Esta suma implica que en ella estará presente información más genérica de dos grupos de colores diferentes, lo que se puede asemejar a un filtro paso-bajo. Así, a partir de un histograma de 256 valores, se generará otro nuevo de 128 valores que contenga la información más gruesa del color, pero cuya representación ocupa la mitad de espacio. Esta operación suma se puede realizar sucesivamente para tener una representación de 64, 32 o 16 niveles. La escalabilidad de este descriptor es debida gracias a este punto: un sistema que necesite poca información de color puede generar un descriptor de 64 bits, teniendo en cuenta la nueva cuantización de los planos de color especificada en la Tabla II-2, mientras que otro que necesite mucha información puede generar uno de 256 bits. Es más, si es necesario

comparar imágenes de ambos sistemas, es tan sencillo como aplicar dos veces la transformada de Haar para llegar a 64 bits en ambas imágenes.

Tabla II-2: Partición del espacio de color en función del número de niveles del histograma [OHM01]

Número coeficientes	Número de bins H	Número de bins S	Número de bins V
16	4	2	2
32	8	2	2
64	8	2	4
128	8	4	4
256	16	4	4

Por otro lado, SCD también define otro posible descriptor que ayuda a la escalabilidad espacial del descriptor. A la hora de aplicar el filtro paso-bajo de Haar, también es posible aplicar una resta. Al aplicar la resta entre niveles adyacentes, la información obtenida es información del detalle, actuando como un filtro paso-alto. Los valores diferencia se pueden codificar con diferente número de bits, permitiendo obtener una representación compacta, donde en su extremo sólo se guarda la información del signo, ya que pueden ser positivos o negativos. Pero la precisión de la recuperación aumenta en función del número de bits y coeficientes utilizados en la representación [DESE08].

En cuanto a la forma de matching, la sugerencia realizada por el grupo de interés de MPEG indica que se pueden correlar los coeficientes de Haar o el propio histograma usando la distancia L1.

II.2.6.2 MPEG7 – Edge Histogram Descriptor

Otro de los descriptores más importantes del estándar MPEG7 es el Edge Histogram Descriptor (EHD). La idea es agrupar pequeños grupos de píxeles de una imagen y detectar en ellos la presencia de un borde horizontal, vertical, diagonal (45º y 135º), o un borde no direccional. Una vez computado esto para todos los píxeles, se generan varios

histogramas que agrupan toda esta información de bordes y ese es el descriptor final. Un diagrama que recoge estos pasos se presenta en la Figura II-14.

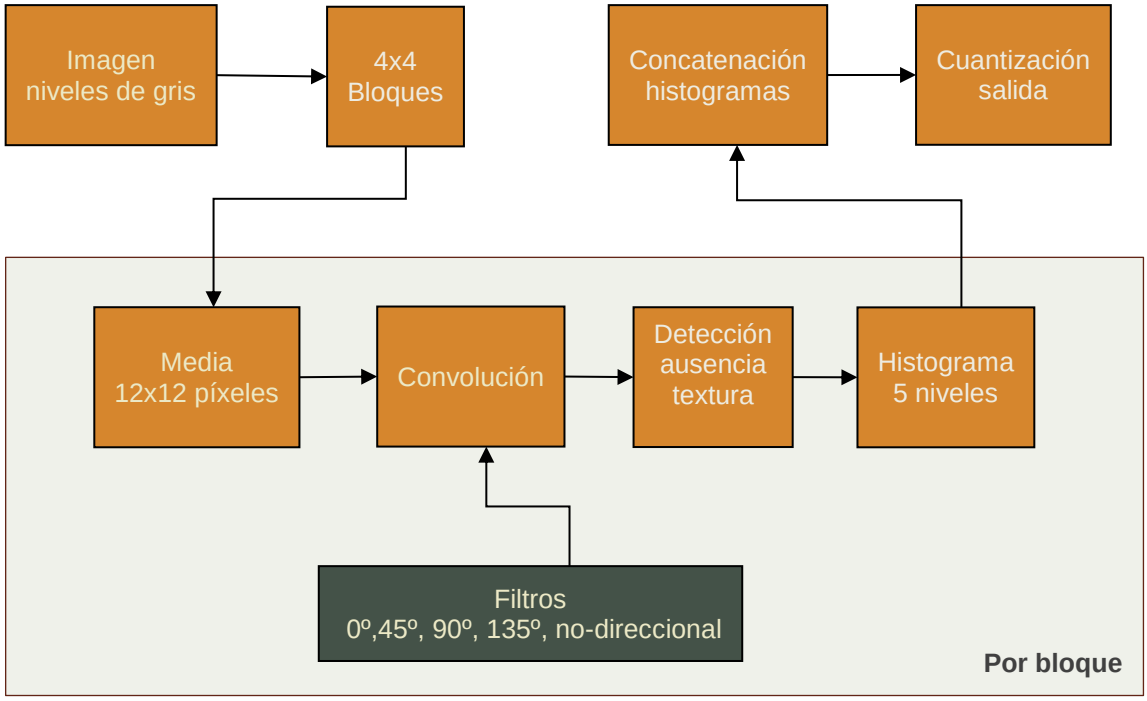


Figura II-14: Pasos necesarios para computar el descriptor MPEG7-EHD

Más detalladamente, el primer paso a dar es representar la imagen en regiones, para lo que primero se divide la imagen en 4x4 sub-imágenes. Cada sub-imagen se vuelve a dividir en regiones más pequeñas de unos pocos píxeles, y en esas sub-regiones se analiza si el borde presente es vertical, diagonal (45º y 135º), horizontal o un borde no direccional. Para ello, el primer paso realizado en la implementación de referencia del estándar es seleccionar vecindarios de 12x12 píxeles y computar su valor medio. A continuación, seleccionando 2x2 salidas de vecindarios adyacentes, se aplican 5 filtros lineales que se corresponden con cada uno de los cinco tipos de bordes (Figura II-15). De los cinco valores de salida para cada grupo de píxeles, se selecciona el mayor de ellos y es éste el que se asigna a esa zona como borde presente.

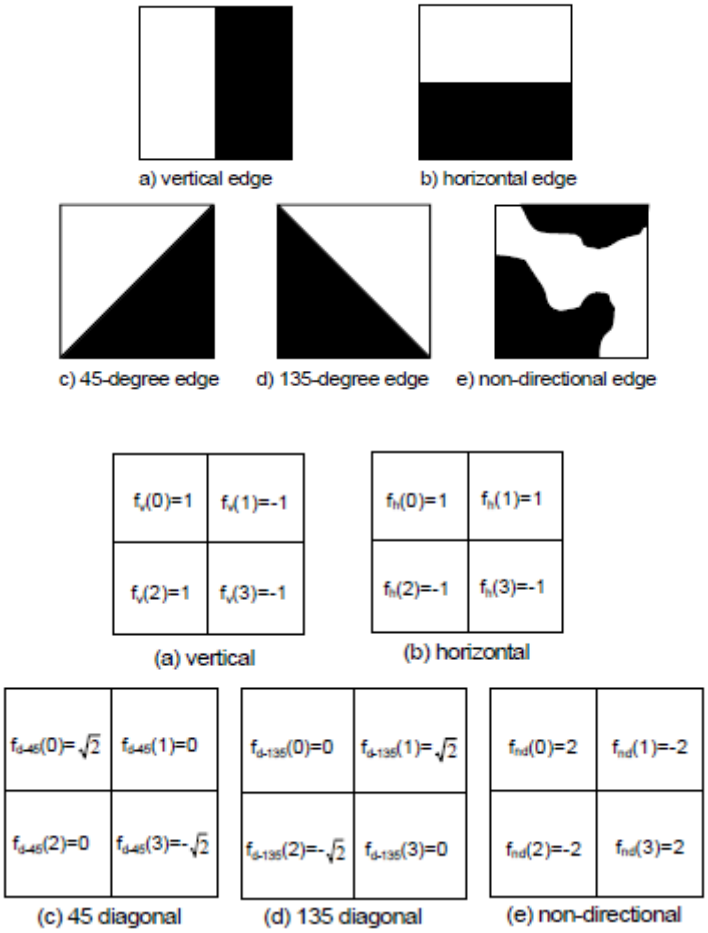


Figura II-15: Filtros lineales para especificar cada tipo de borde. Representación visual (arriba) y matemática (abajo) [CHEE02]

Para cada sub-región se computa la presencia de uno de estos bordes y contribuye a un histograma de 5 niveles en el que se contabilizan los bordes presentes en toda la sub-imagen. Además de los cinco bordes, también se especifica un umbral, de forma que si ninguna de las respuestas pasa de dicho umbral, se considera que no existe borde. Este umbral deberá ser específico de la aplicación, pero se recomienda un valor de 11. Como se puede ver, los filtros propuestos son totalmente arbitrarios, con el filtro de *borde no direccional* como ejemplo claro. El estándar MPEG7 sugiere que este filtro puede ser sustituido por otro, o incluso utilizar un segundo umbralizado para reconocer si existe borde no direccional o no existe borde [CHEE02].

El proceso descrito se repite para las 16 sub-imágenes y sus salidas se concatenan, obteniendo un descriptor de 5x16=80 coeficientes. Para compactar la representación del

mismo, el estándar propone el uso de una cuantización no lineal, de tal forma que se usen sólo 3 bits por coeficiente, tal y como muestra la Tabla II-3.

Tabla II-3: Tabla de cuantización del estándar MPEG7-EHD donde se presentan los valores representativos del bin junto con el valor en binario [CHEE02]

Bin count	Valores para bordes verticales	Valores para bordes horizontales	Valores para bordes de 45 grados	Valores para bordes de 135 grados	Calores para bordes no direccionales
000	0.010867	0.012266	0.004193	0.004174	0.006778
001	0.057915	0.069934	0.025852	0.025924	0.051667
010	0.099526	0.125879	0.046860	0.046232	0.108650
011	0.144849	0.182307	0.068519	0.067163	0.166257
100	0.195573	0.243396	0.093286	0.089655	0.224226
101	0.260504	0.314563	0.123490	0.115391	0.285691
110	0.358031	0.411728	0.161505	0.151904	0.356375
111	0.530128	0.564319	0.228960	0.217745	0.450972

II.2.7 Color and Edge Directivity Descriptor

El estándar MPEG7 fue propuesto en 2002 y, desde entonces, han sido varios autores los que han intentado proponer nuevos descriptores visuales basándose en conceptos propuestos por el estándar. Uno de los más satisfactorios [IAKO14] es el descriptor Color and Edge Directivity Descriptor (CEDD) [CHAT08].

La estructura de este descriptor se inspira en las estructuras y conceptos de MPEG7, por lo que comienza dividiendo la imagen de entrada en 1.600 sub-imágenes rectangulares. Cada sub-imagen se clasifica en base a su color y textura. Para extraer la información de color, CEDD propone un procedimiento de dos fases basado en histogramas difusos, que une toda la gama de colores de entrada en un histograma de 24 colores pre-configurados.

Por otra parte, de la misma sub-imagen también se extrae información de textura utilizando los mismos filtros que MPEG7-EHD (Figura II-15), de cuyas salidas se selecciona la respuesta máxima normalizada y se umbraliza para componer un vector de textura único.

Para finalizar, todos estos sub-vectores se agrupan para poder obtener el descriptor visual final, el cual es tan compacto como el de MPEG7, pero obtiene mejores resultados [CHAT08].

II.2.8 Scale-Invariant Feature Transform

Todos los descriptores detallados hasta ahora se encontraban dentro de la categoría de descriptores *globales* de imagen, ya que su objetivo era representar la información visual de la imagen completa mediante un único descriptor.

En el lado opuesto se encuentran los descriptores *locales* de imagen, los cuales se centran en detectar puntos característicos y repetitivos de una imagen, y representar una zona local. Dentro de estos descriptores locales de imagen, el descriptor más utilizado es el Scale-Invariant Feature Transform (SIFT) [LOWE04]. SIFT se basa en una serie de pasos para lograr una representación de los puntos locales característicos de una imagen de forma invariante a escala, rotación y traslación. Dichos pasos son los siguientes:

1. Como los elementos presentes en una imagen pueden contener diferentes tamaños, el primer paso es **generar un espacio de escalas** (Figura II-16), asegurándose invariancia a escala.

Figura II-16: Espacio de escalas para una imagen

2. El siguiente paso es la **computación de los puntos de interés potenciales** en una imagen. Para ello, se utiliza la operación de “Laplacian of Gaussian (LoG)”,

donde a partir de una imagen se computan las derivadas de segundo orden de esa misma imagen en una escala inferior. A pesar de que esta operación es computacionalmente muy pesada para aplicarlo a todas las escalas, en SIFT se propone una solución basada en restar imágenes de escalas consecutivas para simular la LoG.

3. A partir de este último cálculo, es necesario **detectar los puntos de interés**, que se encontrarán en los máximos y mínimos de la resta de las escalas. En SIFT se busca mucho más detalle, detectando la posición exacta de ellos con precisión sub-píxel, mediante la expansión de Taylor de la imagen en ese punto para aproximar el punto. En la Figura II-17 se ve un ejemplo de este tipo de detección sub-píxel: Los puntos rojos en la imagen anterior marcan los píxeles de una imagen y la curva indica el nivel de gris. Se comprueba que el máximo del nivel de gris (punto verde) puede encontrarse fuera de un píxel de la imagen (puntos rojos).

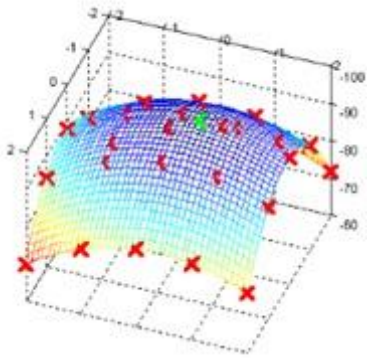


Figura II-17: Detección sub-píxel

4. El paso anterior genera un gran número de puntos característicos, y SIFT realiza una **limpieza** de ellos eliminando aquellos que se detecten en bordes de la imagen (computando el gradiente de la imagen en el punto) y aquellos puntos en regiones con poco contraste (en base a un umbral).
5. Con los pasos anteriores se han conseguido puntos característicos estables e invariantes a escala. Para conseguir la **invariancia a la rotación** se computan la magnitud y la orientación del gradiente de la imagen en base a las siguientes fórmulas:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = \tan^{-1}\Big(\big(L(x,y+1) - L(x,y-1)\big)/\big(L(x+1,y) - L(x-1,y)\big)\Big)$$

Calculando estos valores para diferentes puntos de una región limitada de la imagen se computa un histograma de orientaciones añadiendo en cada nivel la magnitud computada. Gracias a este histograma de orientaciones todos los cálculos posteriores serán descritos en función de la orientación principal y, por tanto, serán invariantes a rotación.

Una vez computados los puntos, se extrae el propio descriptor SIFT, que busca ser una firma de la región alrededor de cada punto. Para ello, la región se divide en 4x4 bloques, y estos a su vez en 4x4 sub-bloques (Figura II-18). Dentro de cada bloque se computa la magnitud y orientación del gradiente, agregando todo en un histograma de 8 niveles de orientaciones. A cada orientación se le suma la magnitud del gradiente, ponderado por una distancia al centro del punto, y de esta manera, los gradientes más alejados tendrán menos influencia. Concatenando los 4x4x8 valores se genera el vector de características final de 128 dimensiones.

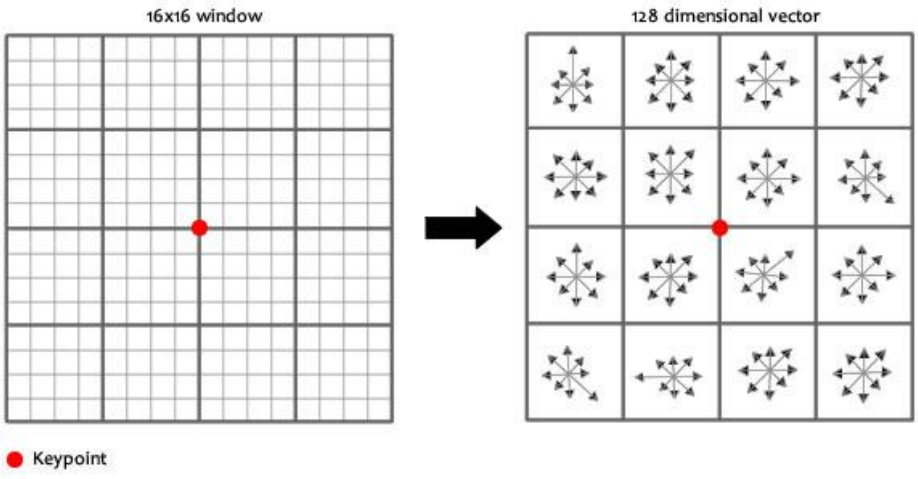


Figura II-18: Regiones para el cálculo del descriptor SIFT alrededor de un punto

La sencillez del proceso y su alto nivel de invariancia han hecho de SIFT uno de los descriptores más utilizados en el estado del arte. Evidentemente, el número de modificaciones de SIFT ha crecido exponencialmente con el paso de los años, teniendo

variantes como Dense SIFT [BOSC06], que no realiza una detección de puntos, sino que describe toda la imagen por zonas, o SURF [BAY08], basado en la detección rápida de las características en base a transformadas Haar y sucesivas aproximaciones del problema.

Toda la algoritmia propuesta en SIFT se diseñó pensando en imágenes de un único canal que define el nivel de gris de la imagen. Dentro de las variantes propuestas, también existen otras variantes para aplicar SIFT a imágenes de color como HSV-SIFT, HueSIFT, OpponentSIFT, C-SIFT, *rg*SIFT, Transformed color SIFT o RGB-SIFT, entre otros [VAN10]. La ventaja de utilización de variantes de SIFT sobre imágenes en color ha quedado probada, y se estima en un 7-8% [VAN10]. Una recomendación realizada es utilizar la variante OpponentSIFT, que aplica SIFT a los 3 canales oponentes (ver apartado II.2.2) obteniendo un descriptor de 3x128 valores.

II.2.9 Local Binary Pattern

Los Local Binary Pattern, conocidos como LBP, son descriptores locales que tratan de describir la textura existente en una zona concreta de la imagen [OJAL96] considerando imágenes en niveles de gris. La aplicación básica del algoritmo de LBP es seleccionar una vecindad de 8 píxeles alrededor del punto de trabajo. Si el valor de uno de esos 8 píxeles es mayor que un umbral, entonces se indica con un 1, y si es menor ese píxel se marca con un 0. Esta operación de umbralizado se ilustra en la Figura II-19.

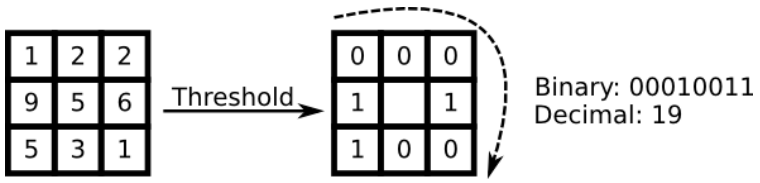


Figura II-19: Operación de umbralizado del LBP

El siguiente paso es transformar la firma de 8 bits en un número entero (entre 0 y 255). Todos estos valores se agregan en un histograma, y ya es posible tener un descriptor de dicha región basado en LBP.

II.2.10 Combinación de características

En sistemas de anotación automática de imágenes, así como en otros campos de visión artificial, lo importante es tener una idea de la imagen en su conjunto y, por tanto, integrar toda la información posible en un único vector que describa el contenido de la misma.

Los descriptores que se han mostrado anteriormente se centran en la descripción de una componente de la imagen: ya sea información del color, de la textura, de la silueta o de elementos locales existentes. Como se intuye, la combinación de características posee un resultado potencial mucho mayor que el uso de características individuales. Por ejemplo, en el caso del estándar MPEG7, se indica que para las aplicaciones de recuperación de imágenes se puede incrementar su rendimiento si el descriptor de bordes MPEG7-EHD se combina con otros descriptores como, por ejemplo, el histograma de color SCD [DESE08]. En este apartado se muestran diferentes aproximaciones de combinación de descriptores que existen en el estado del arte.

II.2.10.1 Concatenación y normalización

La forma más sencilla de utilizar varios descriptores es la concatenación de los mismos. Así, por ejemplo, ante un histograma de color y otro descriptor de textura, como por ejemplo HOG, la concatenación de ambos dará como resultado un vector único que combine textura y color. Esta concatenación no puede ser realizada de forma directa, ya que los rangos de cada uno de los vectores individuales son diferentes, y puede ser que un algoritmo de Inteligencia Artificial que utilice ese vector concatenado tienda a hacer más caso a unas características que a otras.

Para evitar este fenómeno existen los métodos de estandarización de características. El método más habitual es estandarizar los vectores para que cada característica del vector posea media cero y varianza unidad. Con este objetivo, dado un conjunto de datos, es posible calcular la media de cada una de las características y la varianza de las mismas. Con ellas se puede aplicar la siguiente transformación a cada característica del descriptor:

$$X' = \frac{X - \bar{X}}{\sigma}$$

Así, todas las características del vector agregado, independientemente de su origen, estarán normalizadas y tendrán el mismo peso.

Otra estandarización posible es desplazar todo el rango de datos a un rango determinado (normalmente entre 0 y +1 o entre -1 y +1), que permite ser más robustos frente a pequeñas desviaciones estándar de las características, así como mantener aquellas que son cero; cosa que no pasa en el caso anterior. Esta transformación es igualmente sencilla y tiene la siguiente forma:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

II.2.10.2 Bag of Visual Words

Como se ha descrito, la agrupación de descriptores heterogéneos es una práctica muy extendida en el estado del arte, principalmente con descriptores globales de imagen. En el caso de la utilización de descriptores locales esta combinación es más compleja, ya que en una imagen existe un gran volumen descriptores de puntos locales. Si se combinan mediante una concatenación de todos ellos, se puede tener un descriptor muy grande que sea inoperable e ineficaz.

En el caso de descriptores locales, la aproximación de Bag of Visual Words es la más utilizada en el estado del arte [CSURK04] [LAZEB06] [JEGO10] [ZHOU13] [YU14]. El concepto *Bag of Words* fue introducido inicialmente en el campo de análisis de documentos textuales en [HARR54], y sus etapas se muestren en la Figura II-20.

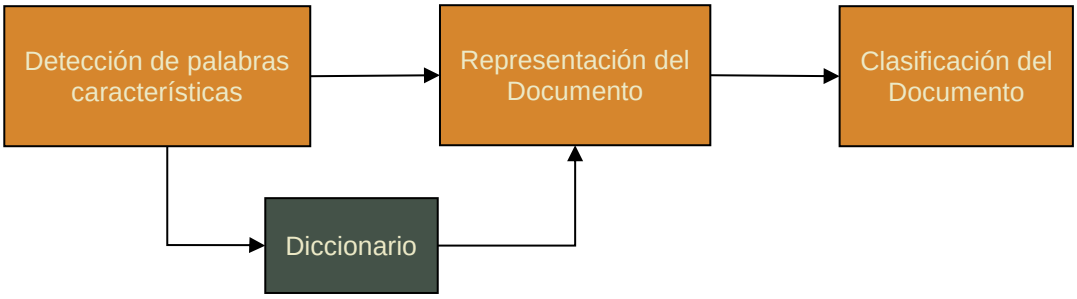


Figura II-20: Pasos correspondientes a la representación de un documento textual mediante el marco de Bag of Words

Esta idea trata de representar un documento textual como un único vector matemático y, para ello, el primer paso es seleccionar un diccionario de palabras. Este diccionario contendrá todas las palabras únicas que existan en un conjunto de documentos amplio (de forma que se maximice la capacidad de representación), e idealmente sería todo el diccionario de la lengua de los documentos. Así, con este diccionario y un documento se puede generar un histograma de palabras, de forma que para cada palabra del diccionario se cuenten el número de repeticiones de la misma en el documento de interés. Este histograma se puede ver como una bolsa de palabras, ya que no tienen ningún orden específico ni relación gramática entre elementos, pero permite representar el documento de texto de una forma compacta que permita diferenciar documentos de temáticas diferentes (Figura II-21).

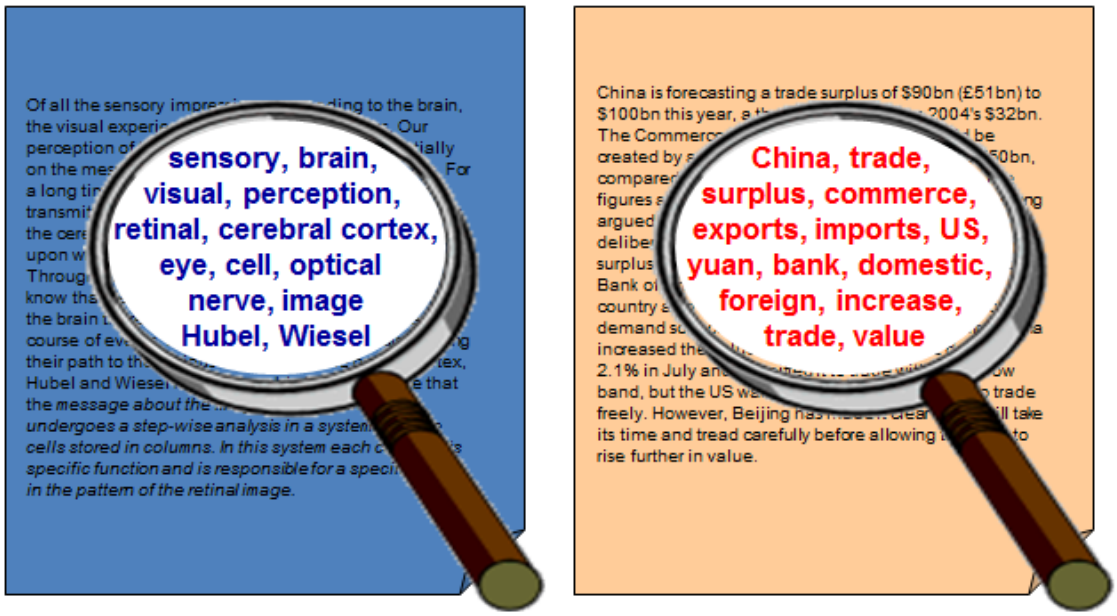


Figura II-21: Dos documentos diferentes sobre temáticas diferentes no tendrán en común la mayoría de las palabras

Mediante este método sencillo se pueden clasificar documentos en diferentes categorías, y un ejemplo clásico de su utilización ha sido el filtrado de *Spam* (usado por primera vez en 1996).

Uno de los primeros trabajos que introdujo este concepto en el campo de la visión artificial fue [CSUR04]. En el caso de una imagen no se disponen de palabras discretas, por lo que

el objetivo es detectar zonas repetitivas en imágenes, a las cuales se les denominará palabras visuales o simplemente palabras.

Idealmente, los elementos que compondrían el diccionario de palabras visuales serían elementos reales (como “ojo”, “rueda”,…) como se muestra en la Figura II-22, pero en el campo de la visión artificial no es posible tener tal grado de precisión.

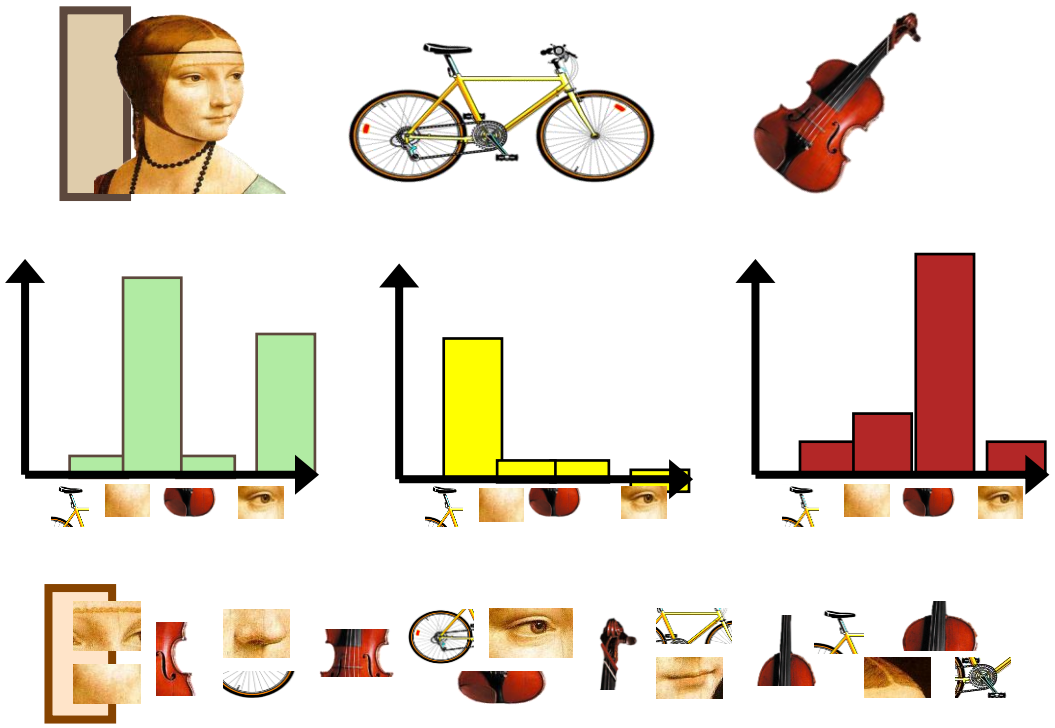


Figura II-22: Representación del marco Bag of Visual Words. Arriba: se presentan tres imágenes diferentes. Abajo: Se muestran diferentes regiones relevantes de las imágenes. Centro: Se muestran tres histogramas representando a cada imagen para un diccionario común

Esto hace que se utilice la detección de zonas relevantes de la imagen y cada zona relevante se describe con un descriptor local. Como la realidad dice que la representación visual de estas zonas relevantes no es repetitiva (ante diferentes iluminaciones, la representación variará ligeramente), el objetivo será cuantizar este conjunto de elementos, es decir, generar patrones comunes en base a técnicas de agrupación para generar el diccionario. Este nuevo proceso se ve en el diagrama de la Figura II-23. Los bloques de la derecha e inferior representan los mismos bloques existentes en el modelo

original. Los bloques de la izquierda representan los pasos adicionales necesarios para la adaptación de esta metodología al campo de la visión artificial.

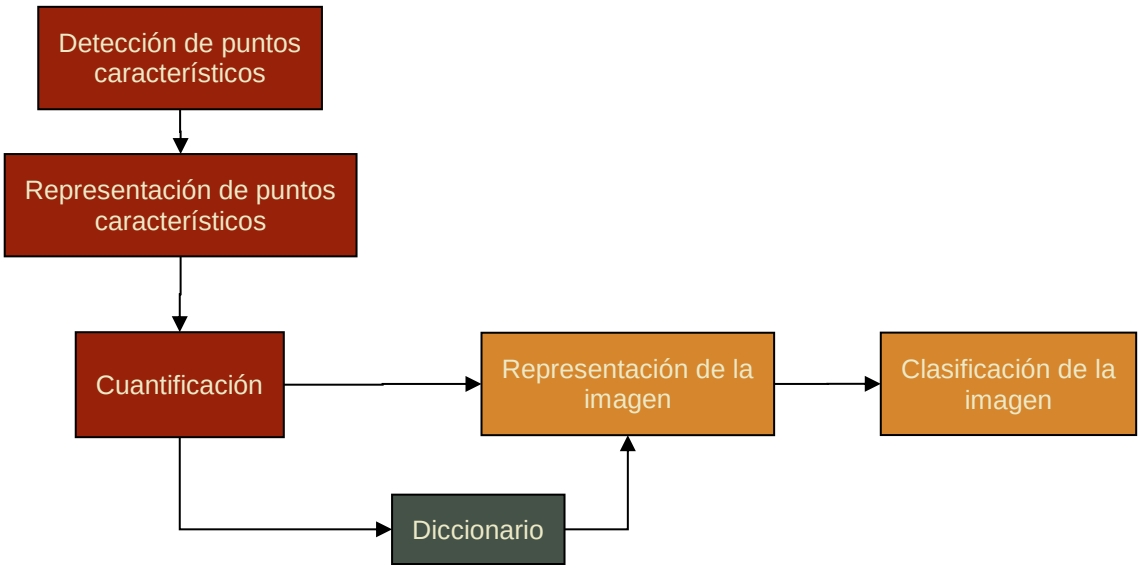


Figura II-23: Diagrama de bloques de Bag of Visual Words.

En [CSUR04], el primer paso dado era la detección y representación de los puntos característicos mediante el algoritmo *SIFT*, ya que se ha visto como comprende estos dos pasos fundamentales: detección de puntos representativos y descripción de los mismos. Con los diferentes descriptores extraídos para un grupo de imágenes, usando un algoritmo de agrupación *K-Means* con un número concreto de centroides (optimizado mediante técnicas de validación cruzada) se generaba el diccionario de palabras visuales. Con este diccionario ya era posible representar una imagen mediante su histograma. Para ello, ante una imagen a representar se computaban y representaban sus puntos característicos mediante el algoritmo SIFT. Para cada punto, se encontraba qué palabra del diccionario es la más cercana mediante el cómputo de la distancia euclidea y se sumaba una unidad a ese nivel del histograma. Haciendo esto para todos los puntos SIFT se obtenía el vector final que representa a la imagen, y éste podría ser utilizado en un sistema de recuperación de imágenes o de clasificación de imágenes.

Esta metodología se ha utilizado ampliamente y es el estado del arte en campos como la detección de objetos. Por ejemplo en la competición PASCAL VOC 2010 el grupo ganador utilizaba una agregación de características y Bag of Words (Figura II-24).

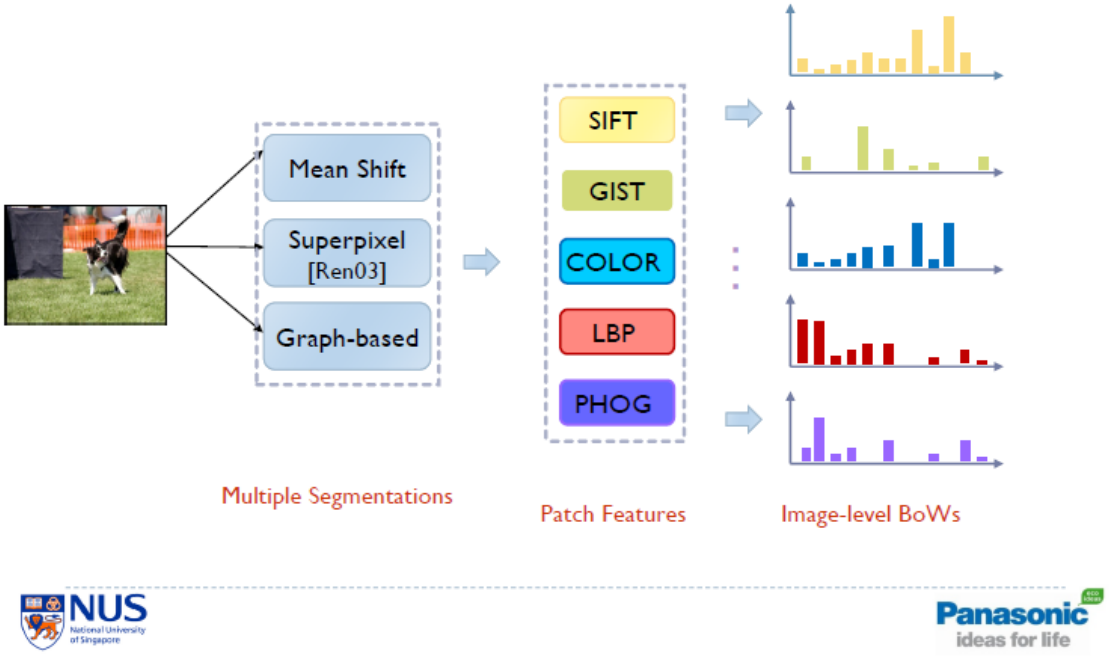


Figura II-24: Representación de una imagen en base a múltiples características visuales de regiones de la imagen (segmentadas mediante tres métodos). Fuente: PASCAL VOC 2010

II.2.11 Descriptores globales de medio y alto nivel

Los anteriores descriptores mostrados se han centrado en representar la información de bajo nivel de una imagen, como son el color o los bordes de la misma. Pero esta información no es la única que se puede utilizar para una representación, ya que una combinación de dicha información puede dar un valor mayor que ella misma. Esta combinación se puede dar de forma explícita como en el punto II.2.10. Pero también se puede combinar esta información en los pasos iniciales de la computación del descriptor.

Este es el caso del descriptor GIST, uno de los descriptores más utilizados en la actualidad en el campo del análisis de escenas por la calidad de su representación y por lo compacto de la misma [DOUZ09]. El descriptor GIST fue propuesto en [OLIV01] como un descriptor compacto que representa la "esencia" de una imagen. Este descriptor se basa en la idea de generar una representación de baja dimensionalidad de toda la escena, por lo que este descriptor global representará el contenido genérico de la misma.

A diferencia de los otros casos, la representación final no se hará con características de bajo nivel (como color o textura), sino que se hará de información de medio nivel. Los autores proponen un conjunto de dimensiones perceptuales humanas que representan la estructura espacial que predomina en dicha escena, siendo estas dimensiones las siguientes: cantidad de naturaleza presente, apertura de la imagen, tosquedad de la imagen, expansión y aspereza. En su artículo, los autores demuestran que estas características pueden ser estimadas de forma precisa en base a propiedades espectrales y localizadas de la imagen.

En cuanto al procesamiento de la imagen, ésta se divide en 4x4 segmentos y en cada uno de ellos se computan histogramas de orientación, de una forma muy semejante a SIFT, computando finalmente un vector, que de forma genérica posee 960 dimensiones.

Otro grupo de descriptores que dan información global sobre una imagen son aquellos que basan su representación en información de *clasificación* [BERG14]. La idea de estos descriptores es representar una imagen utilizando características de bajo nivel, pero realizando su representación a un mayor nivel semántico.

Uno de estos tipos de descriptores es *Classemes* [TORR10]. En este descriptor, se utilizan varias características de bajo nivel para representar la imagen: Color GIST, HOG, SSIM [SHEC07] y SIFT (combinado en base a BoW). Para hacer la representación final de la imagen en su conjunto, se toma como base un grupo de *C* clases definidas directamente como objetos reales obtenidos del conjunto de entrenamiento. Así, el descriptor final estará compuesto por la salida de cada una de los *C* clasificadores entrenados como 1-vs-the-rest. En este trabajo, el clasificador implementado es el LP-Beta [GEHL09], por ser uno de los que mejores resultados obtiene en el estado del arte, y no es más que una combinación lineal de varias SVM no lineales, cada una de ellas entrenada con una característica específica de bajo nivel. Estos autores también logran un descriptor mucho más compacto, binarizando la salida de los clasificadores LP-Beta y, por tanto, teniendo un descriptor binario.

Otro descriptor de este tipo es el presentado en [BERG11] y llamado *PiCoDes*. En este caso, las características usadas son las mismas que en el caso anterior. La principal diferencia es la utilización de un conjunto de clases-base diferentes para representar la imagen, y por tanto la modificación de todo el sistema de aprendizaje. En este caso, el

objetivo es evitar utilizar clases que hayan sido seleccionadas arbitrariamente a mano. Para ello, se ha propuesto un objetivo del algoritmo de aprendizaje tal que consiga generar un conjunto de clases base de forma que se maximice el resultado final. Para hacer que este descriptor sea sencillo y eficiente, se buscan clases cuya respuesta ante una combinación lineal sea máxima y, por tanto, el objetivo de aprendizaje se modifica para tal fin. Otra modificación es que no se mapean de forma directa las clases aprendidas y el vector de salida final, sino que es ese algoritmo de aprendizaje el que hace la conversión. Esto les permite lograr un tamaño variable del descriptor sin modificar el tamaño óptimo de clases base.

II.3 Tecnologías de búsqueda de vectores similares

Como se ha visto en el apartado II.2, cuando se desea analizar una imagen de forma automática, el primer paso es representarla mediante un descriptor visual, que no es más que un vector numérico que representa el contenido de la misma.

Tras este paso comienza el trabajo que se desea realizar sobre la imagen, que puede ser una clasificación del contenido de la misma, la detección de elementos presentes en ella u otros. Este apartado se trata específicamente de la búsqueda de imágenes similares, ya que es el necesario de cara al alcance de esta tesis.

En general, la búsqueda por similitud hace referencia en la búsqueda de elementos lo más parecidos posible al objeto de consulta en el espacio N dimensional. En el caso de las imágenes, se buscan aquellas que dentro de una base de datos sean visualmente parecidas a la imagen de consulta.

Este procedimiento de búsqueda trata de comparar los descriptores visuales de las imágenes de la base de datos con el descriptor visual de la imagen de entrada. Para realizar esta comparación hay varias aproximaciones relevantes en el estado del arte:

- La aproximación más general es computar la distancia entre el vector de consulta o test y los vectores de la base de datos. Para ello, se utiliza una métrica de distancia que generará los vecinos más cercanos y, por tanto, los visualmente más similares.
- Una modificación de esta aproximación es calcular los vecinos más cercanos, utilizando una métrica específicamente aprendida para cada base de datos. Intuitivamente se puede comprender que esta aproximación logra obtener mejores resultados, ya que está adaptada a la base de datos concreta.

- Por otra parte, existe una aproximación igualmente popular y válida, especialmente indicada para bases de datos de gran tamaño. Esta aproximación propone computar los vecinos más cercanos de forma *aproximada* utilizando técnicas mucho más eficientes. Evidentemente, el cálculo será mucho más rápido, pero no será tan preciso como en las otras aproximaciones donde se hace una búsqueda exhaustiva.

En los siguientes sub-apartados se describirán las diferentes métricas y algoritmos que se utilizan en estas tres aproximaciones. Primero, se describirán las métricas de distancia más utilizadas en este campo, luego se mostrarán varios algoritmos de aprendizaje de métricas, para finalizar explicando las técnicas del estado del arte de computación de similitud vectorial mediante técnicas rápidas.

II.3.1 Funciones de distancia para búsqueda por similitud

Dados dos vectores genéricos, es posible calcular la distancia entre ellos utilizando una función, llamada "función distancia" o "métrica". Estas funciones se aplican a un espacio N-dimensional para calcular un valor numérico que da una idea de cuán lejos o cerca están dos vectores en ese espacio. Como se puede intuir, el concepto de "lejos" y "cerca" está directamente relacionado con los conceptos de "diferente" y "similar", respectivamente.

Matemáticamente, la definición de una función de distancia *d* en un conjunto de elementos *Y* es la siguiente:

$$d: YxY \rightarrow \mathbb{R}$$

que satisfice las siguientes condiciones para todo *x, y, z* en *Y*:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0$ si y sólo si $x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Todas las funciones *d* que satisfagan las condiciones previas se denominan métricas, pero estas condiciones también se relajan, generando otro grupo de definiciones de métricas. Los dos grupos más comunes son las *pseudométricas* y las *quasimétricas*.

En las *pseudométricas*, se deben cumplir las siguientes condiciones:

1. $d(x, y) \geq 0$
2. $d(x, x) = 0$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

La principal diferencia está relacionada con la segunda condición, de forma que es posible obtener un valor *d(x,y)*=0 para un *x* diferente de *y*.

En el caso de las *quasimétricas*, sólo es necesario satisfacer tres de las condiciones, que son las siguientes:

1. $\mathrm{d}(\mathrm{x}, \mathrm{y}) \geq 0$
2. $\mathrm{d}(\mathrm{x}, \mathrm{y}) = 0 \quad \text{si y sólo si } \mathrm{x} = \mathrm{y}$
3. $\mathrm{d}(\mathrm{x}, \mathrm{z}) \leq \mathrm{d}(\mathrm{x}, \mathrm{y}) + \mathrm{d}(\mathrm{y}, \mathrm{z})$

Volviendo a la definición original de *métrica*, en la literatura existen numerosas distancias para calcular la separación entre dos vectores. La métrica más común es la distancia Euclídea, pero también existen otras funciones que se pueden usar para el caso de vectores visuales, y que se presentan en los siguientes puntos.

II.3.1.1 Distancia Euclídea

La definición de distancia Euclídea, entendida como la distancia entre un par de puntos *x* e *y,* es la siguiente:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

II.3.1.2 Distancia Manhattan

Una version más relajada de la distancia euclídea es la distancia *Manhattan*, también conocida como *Taxicab*. Esta métrica es mucho más ligera, computacionalmente

hablando, ya que no requiere calcular los cuadrados ni la raíz cuadrada de la distancia Euclídea. Por tanto, el resultado es una distancia mucho más rápida de computar.

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

II.3.1.3 Distancia Minkowski

Dadas las dos definiciones anteriores, es posible generalizar una distancia común aplicada al espacio Euclídeo. Esta distancia se llama *Minkowski* o *Lp*, y se define como:

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Se puede ver que la distancia *Manhattan* se obtiene con un coeficiente *p=1* y la distancia Euclídea con un coeficiente *p=2*.

II.3.1.4 Distancia Coseno

La distancia Coseno mide el coseno del ángulo que forman los dos vectores. La definición de esta métrica es de la siguiente forma:

$$d(\bar{x},\bar{y}) = 1 - \cos(\theta) = 1 - \frac{x\, y^t}{\sqrt{(x\, x^t)(y\, y^t)}}$$

Como se puede ver, esta métrica no es muy eficiente en cuanto a su computación, pero permite tener un sentido de la dirección relativa de ambos vectores de entrada. Por ello, esta distancia es muy popular, a pesar de que no satisface la condición del triángulo y, por tanto, no es una métrica completa.

II.3.1.5 Distancia Chi cuadrado

Otra de las distancias más utilizadas en el estado del arte de visión artificial es la distancia Chi cuadrado. Esto se debe principalmente a que está muy relacionada con la medición de similitudes entre dos histogramas, y la mayor parte de las representaciones en visión

artificial se hacen en base a histogramas (por ejemplo, ver el apartado II.2.10.2 Bag of Visual Words).

Esta distancia se basa en el Test estadístico Chi-cuadrado, que evalúa la diferencia entre dos grupos de frecuencias o distribuciones de probabilidad. El Test Chi-Cuadrado se define de la siguiente forma:

$$\chi^2(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|^2}{x_i}$$

El problema de este test es que no es simétrico, por lo que para generar una distancia válida, es necesario generar la versión simétrica. Así, la definición formal de la Dsitancia Chi cuadrado es la siguiente:

$$d(x,y) = \frac{1}{2}\sum_{i=1}^{n} \frac{|x_i - y_i|^2}{x_i + y_i}$$

II.3.1.6 Earth Mover Distance

La denominada Earth Mover Distance (EMD) es un tipo especial de métrica *Wasserstein* y, por tanto, enfocada a la comparación de dos distribuciones de probabilidad dada una función de distancia base. La idea básica de esta distancia es que a partir de un vector visto como un montón de tierra, la EMD mide un valor proporcional a la cantidad de tierra que se necesita mover para generar el otro vector.

La computación de EMD se realiza calculando la solución al *transportation-problem* [HITC41], y una descripción detallada se puede encontrar en dicha referencia.

II.3.1.7 Distancia Hamming

La primera aparición de esta distancia fue en el campo de Teoría de la Información con el objetivo de medir los errores cometidos al recibir una ristra de bits durante una

transmisión. Esta medida se calcula en base a los cambios de bit producidos entre el origen y el destino de la comunicación. Así, para el siguiente ejemplo:

- Señal binaria transmitida: 10110
- Señal binaria recibida: 11111

El número de errores en esta transmision ha sido 2, así que la distancia Hamming entre estos vectores binarios es 2.

Actualmente, esta distancia se ha generalizado y se utiliza de la misma forma, pero aplicado a vectores visuales binarios.

II.3.1.8 Distancia de Mahalanobis

Las distancias definidas previamente sólo tienen en cuenta información de los dos vectores a comparar. Pero existen casos en los que utilizar más información sobre la distribución espacial de ellos puede dar más información sobre las distancias entre los puntos. En general, la distancia de Mahalanobis se utiliza en el contexto en el que todas las posibilidades de los datos se componen por diferentes subconjuntos de datos, donde cada uno tiene unas características determinadas. En el ejemplo de la Figura II-25 se tiene una distribución de probabilidad de la aparición de puntos en el espacio de una dimensión. Esta distribución de probabilidad está, a su vez, compuesta por otras dos distribuciones.

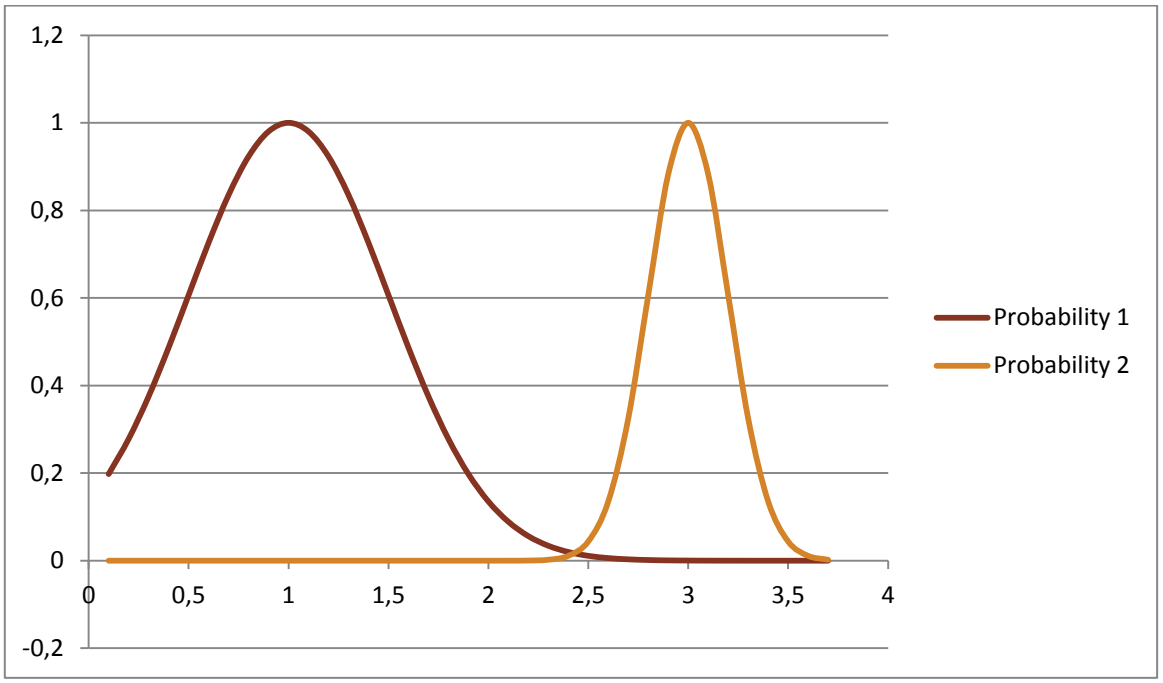


Figura II-25: Distribución de probabilidad de puntos en 1 dimensión. Cada curva representa una componente de la distribución final.

De esta forma, cuando se computen distancias entre dos puntos en este espacio de 1 dimensión, con esa distribución, sería interesante añadir información de la propia distribución. Supongamos el siguiente caso, donde se quiere medir la distancia entre cuatro puntos de una dimensión: A(X=1), B(X=1.5), C(X= 2.75) y D(X=3).

Calculando la distancia Euclídea, se ve que los puntos C y D son más cercanos entre sí que los puntos A y B. Pero si se observa la distribución de los puntos se ve que realmente ambos pares de puntos están a la misma distancia "conceptual", en función de la probabilidad de aparición de los puntos. La idea que subyace en esta afirmación es que la desviación estándar de uno de los conjuntos de puntos es mayor que la otra, por tanto, la influencia en la medida de distancias deberá ser diferente. Esto hace que esta distancia sea ideal cuando se trata de problemas de clasificación, proponiendo la siguiente función:

$$\frac{x-\mu}{\sigma}$$

Esta es la versión normalizada de la distancia entre un punto *x* y el centro de su conjunto de datos X_i definido por su media (μ) y su desviación estándar (σ). La versión generalizada de esta distancia es la siguiente:

$$D_Y(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}$$

que mide la distancia entre un vector *x* y un subconjunto cualquiera de datos definido por su media μ y covarianza *S*.

Pero esta función no está en la forma de las anteriormente presentadas, ya que no mide la distancia entre dos puntos en el espacio. Para ello, la distancia Mahalanobis se redefine de la siguiente forma:

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

Donde *x* e *y* son dos vectores de la misma distribución con una matriz de covarianza *S.*

II.3.1.9 Distancia Joint Equal Contribution

La distancia Joint Equal Contribution (JEC) fue propuesta por primera vez en [MAKA10]. El objetivo de esta distancia es combinar métricas diferentes, o combinar la misma métrica aplicada a diferentes espacios vectoriales en un único valor.

Supongamos que existe un vector que representa una imagen, que está compuesto por dos sub-partes: $x = [x_a, x_b]$. Supongamos que cada sub-parte se ha generado con un descriptor visual diferente, p.e. x_a posee información de color y x_b posee información de textura. Para medir la distancia entre dos vectores en este espacio de características combinado, sería interesante medir la distancia entre los primeros sub-vectores y luego entre los segundos. La distancia JEC se encarga de hacer esta operación de forma que se equilibra cada una de las características individuales, así que todas las sub-partes tendrán una contribución equitativa en la distancia final.

La formulación de esta distancia, para un número arbitrario de sub-características de los vectores es la siguiente:

$$d(x,y) = \frac{d_a(x_a, y_a)}{\max(d_a)} + \frac{d_b(x_b, y_b)}{\max(d_b)} + \cdots$$

Donde la función $max(\cdot)$, indica el valor máximo de esta sub-distancia en un conjunto de entrenamiento.

II.3.1.10 Distancia ImageNet

Además de las distancias puramente numéricas, existen diferentes propuestas en el estado del arte que tratan de combinar información textual con los vectores. Un ejemplo de esto es la distancia ImageNet basada en Histogramas de Categorías, propuesta en [DESE11].

Para calcular la distancia entre dos imágenes I_i e I_j, representadas mediante sus descriptores visuales, primero se determinan sus vecinos visualmente similares N_i y N_j en la base de datos *ImageNet* [DENG09]. Una vez conocidas las imágenes más similares, calculan un histograma h_i de las categorías visuales de los vecinos N_i, haciendo esto para ambas imágenes. Estos histogramas de categorías capturan la distribución conceptual de los vecinos, por lo que si las dos imágenes sobre las que se quiere calcular su distancia son conceptualmente similares, también lo serán las distribuciones de los conceptos.

Para calcular un valor numérico para esta similitud, se propone utilizar la distancia Chi cuadrado sobre los histogramas de categorías:

$$d(x,y) = \sum_S \frac{|h_x(S) - h_y(S)|^2}{h_x(S) + h_y(S)}$$

Donde $h_x(S)$ es el histograma de vecinos de la imagen x en todas las categorías S.

II.3.1.11 Kullback-Leibler Divergence

En el campo de la estadística de la teoría de la información existe otra distancia muy utilizada para analizar las diferencias entre dos distribuciones de probabilidad. Esta distancia es la *Kullback-Leibler divergence* (o KL-Divergence), y en general la pérdida de

información cuando se intenta aproximar una de las dos distribuciones con la otra. En el campo de medida de distancias entre dos histogramas, gran parte de la formulación matemática de esta distancia se puede omitir, de tal forma que el cálculo de la misma queda de la siguiente forma:

$$d(x,y) = \sum_{i=1}^{n} x_i log\left(\frac{x_i}{y_i}\right)$$

II.3.2 Aprendizaje de métricas para búsqueda por similitud

En la sección II.3.1 se han mostrado funciones de distancia que son ampliamente utilizadas en la comunidad científica para calcular la similitud entre dos vectores. Estas métricas asumen que las imágenes se han descrito en un espacio de características "bueno", que permite describir el contenido de la imagen de una forma apropiada.

En la mayoría de los casos esto no es real. Muchas veces, los espacios en los que se describen las imágenes son buenos describiendo una propiedad determinada, pero no el conjunto de la imagen. Es más, ante un vector determinado de un espacio de características, puede ser posible que unas partes de dicho vector sean más relevantes que otras, y las distancias anteriores las consideran igualmente relevantes.

Para mejorar y corregir estos problemas, los algoritmos de aprendizaje de métricas buscan modificar el espacio de características para adaptarlo a la naturaleza de los vectores de entrada. Esto significa que estas aproximaciones cambiarán la configuración del espacio para modificar la importancia de las características de los descriptores. Por ejemplo, en el caso de un problema de clasificación, el aprendizaje de la métrica modificará el espacio de entrada para agrupar las muestras que pertenezcan a la misma clase.

En general, la gran mayoría de algoritmos utilizan la distancia de Mahalanobis (II.3.1.8) como base:

$$d(x,y) = \sqrt{(x-\mathrm{y})^T S^{-1}(x-\mathrm{y})}$$

Un algoritmo de aprendizaje de la métrica buscará aprender una matriz *S* definida positiva, que logrará modificar la distancia clásica y adaptarla al problema que se intenta resolver.

Con esta finalidad se han propuesto muchos algoritmos a lo largo de los años. En los siguientes sub-apartados se muestran los algoritmos más relevantes para la temática de la búsqueda por similitud.

II.3.2.1 Information-Theoretic Metric Learning

Information-Theoretic Metric Learning (ITML) [DAVI07] se basa en la información proporcionada por otra función de distancia básica. A partir de esta información, se crea una matriz de inicialización S_0 y busca minimizar la divergencia entre la matriz objetivo S y la anterior. A continuación, se muestra la divergencia a minimizar con las restricciones establecidas por un conjunto de parejas de imágenes similares y otro conjunto de parejas diferentes:

$$\min_{S>0} d_{ld}(S, S_0)$$

s.t.

$$d_A(x, y) \leq u \quad (x, y) \in S,$$

$$d_A(x, y) \geq l \quad (x, y) \in D,$$

donde $d_{ld}(S,S_0)=tr(AA_0^{-1})-\log \det(AA_0^{-1})-d$, *S* y *D* son conjuntos de parejas similares y diferentes, y *l*, *u* son dos valores arbitrarios grande y pequeño respectivamente.

II.3.2.2 Metric Learning to Rank

En el trabajo [MCFE10] los autores cambian la perspectiva de este tipo de problemas e interpretan el problema de aprendizaje de métricas como un problema de recuperación de información. Esta es la principal diferencia con otros autores, ya que como un problema de recuperación de información los autores aplican una función de coste a un ranking de

documentos de salida, en vez de utilizar funciones de optimización basadas en parejas de documentos.

El objetivo, en este caso, es aprender *S*, por lo que utilizan una máquina de vector soporte de salidas estructuradas [TSOC04] para obtener el *S* que maximiza el margen sobre todos los posibles rankings de documentos de salida. Esto permite obtener un resultado muy similar al estado del arte en términos de precisión media (0.445 contra 0.448) mientras que el tiempo de entrenamiento es notablemente diferente, requiriendo 232 segundos frente a los 4968 del estado del arte.

II.3.2.3 Keep it Simple and Straightforward Metric

Uno de los últimos avances en este típo de técnicas aplicadas a similitud de imágenes es el trabajo presentado en [KOST12]. Sus autores proponen un método no iterativo y, por tanto, más rápido que la mayoría del resto de algoritmos para aprender la matriz S. Para evitar el uso de iteraciones, los autores asumen que un espacio de características que representa a los pares similares y diferentes es gaussiano, y a partir de este punto, reformulan la métrica como una diferencia de matrices de covarianza.

En cuanto a sus resultados, éstos son muy positivos. En la tarea de re-identificación de personas consiguen una tasa de aciertos del 80,7% mientras que ITML sólo logra un 74,7%. Y por parte del tiempo de procesado, en la tarea de identificación de personas, para una tasa de aciertos similar, KISSME tarda 0,05 segundos para el entrenamiento mientras que ITML necesita 24,81 segundos en la misma máquina.

II.3.2.4 OASIS

En el artículo [CHEC10] los autores proponen un algoritmo para aprender las similitudes semánticas, con la peculiaridad de que buscan un algoritmo que escale bien cuando miles de imágenes están involucradas. Con este algoritmo han conseguido pasar de 337 minutos de entrenamiento a 0,12 minutos con el mismo hardware y las mismas imágenes y clases. Para lograr esta propiedad, los autores buscan aprender una similitud bilineal no restringida.

Esto significa que dadas dos imágenes, su similitud se mide mediante $x_1^t \, W \, x_2$, donde *W* no necesita ser positiva pero sí simétrica. Como se puede entender, este es un caso en el que no se crea una distancia de Mahalanobis. Para aprender la matriz deseada, los autores tratan de minimizar la pérdida global que acumula todas las pérdidas individuales

sobre todas las posibles tripletas de imágenes del conjunto de entrenamiento. La función de pérdida global es la siguiente:

$$L_W = \sum_{(p_i,p_i^+,p_i^-)\in P} l_W(p_i, p_i^+, p_i^-)$$

donde las pérdidas individuales son:

$$l_W(p_i, p_i^+, p_i^-) = \max\{0, 1 - d_W(p_i, p_i^+) + d_W(p_i, p_i^-)\}$$

De forma que p_i es una imagen, p_i^+ y p_i^- son imágenes similares y diferentes respectivamente, y d_W es la distancia base.

II.3.3 Algoritmos de búsqueda rápida por similitud

Las dos secciones anteriores se han centrado en obtener las imágenes similares a una imagen dada de forma exhaustiva. El término "exhaustivo" se refiere a que se computan todas las posibles distancias con todas las imágenes de la base de datos de forma exacta mediante el uso de una función de distancia concreta. Así, es posible obtener una lista de imágenes similares con un cálculo de la similitud exacto.

Como es posible imaginar, cuando se está en un escenario de grandes cantidades de imágenes, ese procedimiento de cálculo exhaustivo será computacionalmente muy caro. Para evitar tener que hacer este cálculo, existen diferentes aproximaciones que permiten obtener una lista de imágenes similares logrando un aumento de la velocidad de computación y, por tanto, de la respuesta del sistema.

Una aproximación para aumentar la velocidad de computación es el uso de técnicas rápidas de cálculo exacto. Uno de los algoritmos más populares en este ámbito es **KDTree**. Para computar la lista exacta de imágenes similares, KDTree construye un árbol en memoria, de tal forma que puede hacer las búsquedas y calcular las distancias exactas sin necesidad de recorrer toda la base de datos de forma exhaustiva. Este método es muy efectivo con bases de datos pequeñas en las que la dimensionalidad de los vectores también es baja, pero cuando se involucran una gran número de imágenes o una alta dimensionalidad, el tiempo de computación para recorrer el árbol degenera hacia el mismo tiempo que una búsqueda exhaustiva.

Por otra parte, están los algoritmos que buscan reducir el tamaño de los vectores y la computación de distancias aproximadas. Uno de los algoritmos más populares es el Locallity Sensitive Hashing (**LSH**) [SLAN08]. La idea de este algoritmo es, ante un vector de entrada, generar un vector binario de menor dimensionalidad, de tal forma que los vectores de entrada similares pertenezcan al mismo conjunto en el espacio de los hashes. Así, dado un vector de entrada, el sistema computará su hash y chequeará el conjunto al que pertenece, de forma que la lista de imágenes similares será la lista de imágenes que están en el mismo conjunto.

Como éste es un método muy popular, se han propuesto numerosas variantes del mismo. Por ejemplo, el Kernelized Locallity Sensitive Hashing (**KLSH**) [KULI12] busca utilizar kernels para la computación de los hashes. Gracias a ello permite embeber correlaciones no lineales entre las imágenes, por lo que es especialmente adecuado para espacios de búsqueda no lineales.

En ambos métodos, la computación de los hashes se hace mediante proyecciones aleatorias de los vectores en varias direcciones del espacio N-dimensional. Pero se ha mostrado que si se usan ciertas proyecciones pre-aprendidas se puede mejorar el rendimiento del sistema de búsqueda, tal y como se muestra en el método KSH propuesto en [LIU12].

II.4 Anotación automática de imágenes

En esta tesis el problema abordado es la anotación automática de imágenes. El planteamiento de este problema es sencillo: dada una imagen (llamada imagen de test o de *query*), el objetivo es devolver aquellas etiquetas textuales que describan de la mejor forma posible el contenido de dicha imagen.

El primer problema que se plantea son las propias etiquetas a devolver, ya que hay infinitas posibilidades de descripción de una fotografía, como por ejemplo, describir los colores de la misma de forma semántica, describir los objetos genéricos presentes, como *edificio* o *coche,* o conceptos más específicos que se pueden corresponder con instancias individuales del caso anterior, como *Edificio Empire State* o *coche Peugeot*. El tipo de etiquetas, por tanto, dependerá del objetivo de la aplicación. El otro gran problema es la descripción matemática de la imagen mediante algoritmos de visión artificial, conceptos vistos en apartados anteriores.

En la última década, éste ha sido un campo en el que un gran número de investigadores de diferentes ámbitos han estado muy activos [ZHAN12]. El principal motivo ha sido la aparición de numerosas bases de datos públicas de imágenes anotadas, lo que ha sido debido primero a la digitalización de las imágenes y de los dispositivos de captura, el auge de internet, la omnipresencia de las cámaras fotográficas debido a los dispositivos móviles y, por último, la aparición de numerosos servicios de intercambio y publicación de fotografías. Todo ello unido ha hecho que la necesidad de organizar ese conocimiento mediante etiquetas textuales que puedan ser descubiertas por motores de búsqueda tradicionales sea inmensa, y atraiga a un gran número de grupos de investigación y empresas privadas.

El trabajo realizado en este campo se puede clasificar en tres grupos de modelos utilizados para la anotación [ZHAN10]: modelos generativos, modelos discriminativos y modelos basados en vecinos. Los dos primeros modelos se han explorado muy activamente, especialmente los modelos discriminativos, donde para cada etiqueta textual posible se entrena un clasificador específico. Sin embargo, varios estudios muestran que los modelos basados en vecinos son mucho más apropiados para la anotación de imágenes cuando se trabaje con cientos de etiquetas [TSAI11][DENG10].

En los siguientes apartados se desgranará cada uno de estos modelos y se mostrará el estado del arte en dichos campos, centrándose en los modelos discriminativos por ser los más trabajados en el estado del arte, y en los modelos de vecinos por ser los más adaptados a la problemática presentada en esta tesis.

II.4.1 Modelos Generativos

Los modelos generativos tratan de anotar una imagen en base a las correlaciones estadísticas existentes entre las características visuales de las imágenes y los conceptos semánticos a anotar.

Así, un primer tipo de modelos generativos son los modelos basados en *mezcla*. En ellos se define una distribución estadística conjunta sobre las características de las imágenes y sobre las etiquetas que se usarán para anotar. Dada una imagen a anotar, estos modelos extraen las características visuales de la imagen y calculan la probabilidad condicional de que aparezcan las etiquetas, dadas las características visuales extraídas. Un trabajo muy representativo de este tipo de modelos es [JEON03]. En él sus autores proponen el

concepto de *Cross-Media Relevance Model*, el cual estima la probabilidad conjunta de existencia de las etiquetas de anotación en base a la información visual y en base a la información semántica a partir de las imágenes de entrenamiento. Para ello, se asume que una imagen está compuesta por diferentes regiones, generadas en base a la agrupación de sus características visuales. Adicionalmente, se propone que cada región de una imagen se pueda describir en base a un conjunto limitado de palabras. Así, dado un conjunto de imágenes con sus anotaciones correspondientes, los autores muestran que un modelo probabilístico permite predecir la probabilidad de generar una palabra determinada dadas, un conjunto de regiones presentes en una imagen.

Por otro lado, existe un tipo de modelos que tienen un mayor grado de abstracción y se denominan modelos generativos basados en *temáticas*. Un modelo generativo basado en *temáticas* anota una imagen suponiendo que dicha imagen es una muestra de una combinación específica de *temáticas*, y no tanto de etiquetas. Este mapeo se realiza a nivel de *temáticas*, definiéndose el término *temática* como una distribución estadística sobre las características visuales de la imagen y sobre las etiquetas que se usarán para la anotación.

Estos modelos son los que más se han trabajado a lo largo de los últimos años, por ser capaces de afinar mejor los resultados. Intuitivamente, el primer tipo de modelos debe generar una distribución conjunta para un gran número de imágenes y para un gran número de etiquetas. En cambio, el uso de temáticas intermedias facilita esta tarea y mejora el resultado final.

Algunos ejemplos de este tipo de algoritmos son el *latent Dirichlet allocation* [BARN03], que parte de la suposición de que un documento está compuesto por un conjunto de temáticas concretas, y a través de analizar rasgos específicos (texto e información visual) es posible obtener esa temática; y el *probabilistic latent semantic analysis* [MONA04], desarrollado inicialmente para descubrimiento de temáticas en documentos textuales, donde cada documento está representado por la frecuencia de aparición de sus palabras.

Aun usando modelos basados en temáticas, los modelos generativos se enfrentan a una gran amenaza: el aumento exponencial del número de imágenes de entrenamiento disponible y la necesidad de aumentar el número de palabras a anotar. Esto hace que los modelos probabilísticos tengan que absorber excesiva información y que las

distribuciones existentes sea muy complejas. Existe una línea de investigación que trata de paliar esta problemática, y donde su principal exponente es el método WSABIE [WEST11]. Éste método propone un modelo que permite aprender a representar las imágenes y las anotaciones de forma conjunta en un espacio de menores dimensiones que los espacios completos usados en otro tipo de modelos. Así logran escalabilidad en cuanto a tiempos de entrenamiento y testeo, pero también logran disminuir los requisitos de uso de memoria. En términos de precisión, el resultado de anotación a 10 etiquetas es de 1,48% frente al 1,26% del estado del arte comparado por los autores. Pero la diferencia en uso de memoria y tiempo de ejecución es drásticamente menor: se pasa de 19 días de computación y 8,2GB de memoria usados a 6,5 días y 82 MB.

II.4.2 Modelos Discriminativos

Los modelos discriminativos para la anotación de imágenes se basan en modelar de forma individual las etiquetas que pueden ser predichas usando clasificadores entrenados para ello. El flujo general de un modelo discriminativo se puede ver gráficamente en la Figura II-26.

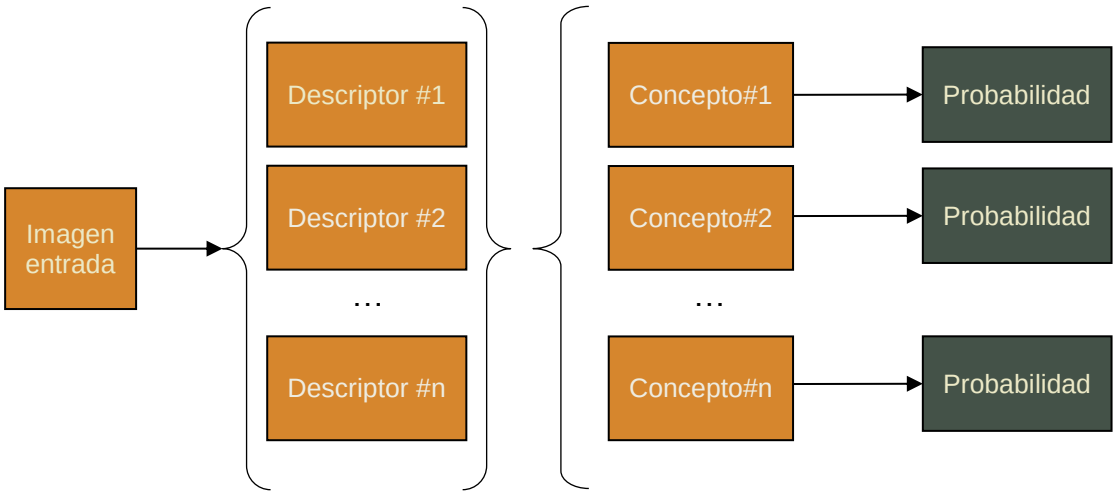


Figura II-26: Modelo discriminativo básico de anotación automática de imágenes

El primer paso a dar es la extracción de las características visuales de la imagen. Esta extracción se realizará en base a algoritmos y descriptores visuales como los presentados en el apartado II.2 y pueden extraerse un único tipo o varios tipos de información visual.

Tras ello, esta información se aplicará, en el entrenamiento y en la prueba, a los clasificadores de etiquetas y ellos determinarán la presencia o ausencia de las mismas. Además de estos pasos sencillos, pueden existir diferentes pasos intermedios para permitir una mejor unión de toda la información. Por ejemplo, si se extraen varios tipos de descriptores visuales, pueden existir métodos de agrupación de los mismos antes de entrar en los clasificadores. Además, la información de las salidas de los clasificadores se puede agrupar para que posean influencia unas sobre las otras, y generar el resultado final. También es posible añadir información adicional (como información de contexto) que ayude a los clasificadores a tomar mejor su decisión.

El objetivo final de los modelos discriminativos es lograr modelar las etiquetas de forma efectiva, ya sean objetos, conceptos genéricos o elementos abstractos. Existen muchas aproximaciones para modelar las clases de conceptos y, en general, se pueden agrupar en: modelos basados en componentes globales y modelos basados en componentes locales. La diferencia fundamental es que los segundos realizan una detección de puntos o regiones características y modelan la imagen en base a esas regiones, que en todas las instancias del concepto a detectar deben ser parecidas. Sin embargo, en los modelos basados en componentes globales se procesa la información de toda la imagen en conjunto para realizar el modelo (no se procesan regiones de forma discreta).

En los siguientes apartados se van a tratar la mayor parte de las líneas de investigación existentes en este área. En primer lugar, se tratarán los modelos de imagen que utilizan información global de la imagen para modelar conceptos. En segundo lugar, se describirá los avances en el modelado de imágenes utilizando componentes locales. En la siguiente subsección se describirá cómo se está utilizando actualmente información que rodea a los conceptos principales para obtener una mejor tasa de detección. También se hablará del modelado de objetos 3D para una detección desde cualquier punto de vista, mientras que también se tratará la detección de objetos mediante su silueta, puesto que los métodos anteriores se basan principalmente en la forma.

II.4.2.1 Modelos basados en componentes globales

La primera línea de investigación que se va a tratar dentro de los modelos discriminativos de anotación de imágenes es la detección de conceptos basada en componentes globales.

Una de las primeras investigaciones sobre la que se han basado posteriormente otros sistemas es la propuesta en [PAPA00]. En ella, proponen utilizar un modelo descriptivo para definir la clase de los objetos, de tal manera que permita describir cualquier posible forma, pose, color y textura de un objeto; y al mismo tiempo, que sea lo suficientemente genérico como para poder modelar cualquier clase de objetos. Para ello, las imágenes son procesadas mediante los wavelets de Haar, y las salidas de estos filtros son los que se utilizaban junto con una SVM para modelar los objetos y generar la clase.

Los wavelets de Haar han sido utilizados en otros sistemas de detección de objetos posteriores, como [VIOL01][VIOL04], que es uno de los algoritmos más populares con una tasa de detección de caras reportada del 93,7%.

Este algoritmo propuesto por Viola y Jones, tiene como característica que, más de 10 años después de su creación, es uno de los más rápidos existentes en el ámbito de la detección de objetos. Para conseguir esta velocidad, se basaron en una nueva representación de las imágenes (llamada imagen integral) que permite un procesamiento rápido de los wavelets de Haar. Además, utiliza un algoritmo de boosting que permite una selección de las características más destacadas para describir un objeto en base a la respuesta de los wavelets. De esta forma, cada clase, en vez de tener información sobre todas las salidas de estos filtros, sólo tiene la información más discriminativa, por lo que aumenta el rendimiento del sistema.

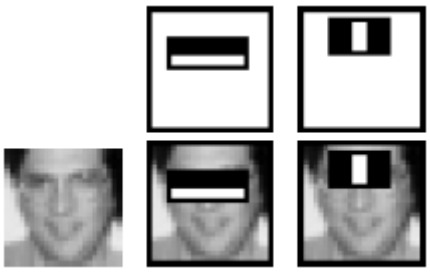


Figura II-27: Detección de componentes en una cara [VIOL04]

Una mejora de este sistema se ha planteado en [LIEN02], donde se propone la utilización de un conjunto diferente de wavelets de Haar. Este conjunto incluiría los wavelets propuestos en [VIOL01], pero también añade más, centrándose en que tengan una rotación de 45º, ya que así se logra una ligera mejora sobre el algoritmo inicial.

Otro caso de detección utilizando información de todo el objeto es la propuesta en [SERR05] y [SERR07]. Aquí proponen la utilización de ciertas características inspiradas en el córtex visual humano para la detección de objetos. De esta forma, los autores han creado un nuevo conjunto de características para la detección robusta de objetos, y utilizan una serie de cuatro etapas que simulan una parte del proceso de reconocimiento de objetos del ser humano. Los resultados que han obtenido son muy satisfactorios, obteniendo tasas de detección de caras del 98,2%, de motocicletas del 98% o de aviones del 96,7%, a pesar de que el tiempo de computación es muy grande.

El problema existente con este tipo de algoritmos es que se centran en la detección de objetos “monolíticos” que posea una cierta estructura común (como pueden ser las caras o los coches). Si el objeto es más deformable (como una persona), entonces este tipo de sistemas presentan más problemas.

Para solucionarlo, otro tipo de investigaciones se ha centrado en analizar información local de los objetos agrupándola de forma global en base a histogramas. La primera propuesta ha sido en [DALA05], que se basa en el uso de histogramas locales de las orientaciones de los gradientes de la imagen. La idea básica es que la apariencia y la forma locales de un objeto pueden ser caracterizadas por la distribución de los gradientes locales o por las direcciones de los bordes, incluso sin un conocimiento preciso de la posición de los gradientes o bordes.

Los autores implementan esta idea dividiendo la imagen en pequeñas regiones (llamadas “celdas”), y computando el histograma de las direcciones del gradiente además de tener en cuenta las posibles variaciones en la iluminación. De esta forma, generan los descriptores HOG que alimentarán a una SVM que será la que generalice y detecte a una clase de objetos, y los resultados generados fueron los más satisfactorios hasta ese momento, teniendo una tasa del 100% de aciertos para personas.

En [ZHU06] se ha mejorado el sistema anterior, de tal forma que se aumenta la velocidad de procesamiento gracias al uso de algoritmos de boosting. Este aumento ha conseguido según sus autores bajar de los 500ms de análisis en [DALA05] a 26ms en su aproximación. En [ZHAN07] han aplicado su framework propio de detección modelando los objetos mediante HOG, y en [FELZ08] lo han aplicado a un modelo de partes deformables.

Otra aproximación para la detección de objetos mediante la apariencia global es el uso de textons [SHOT06] [SHOT09]. Estos descriptores permiten dar información tanto de la forma del objeto como de la textura (de color) de los objetos. La generación de un diccionario de textons es sencilla: se aplica un banco de 17 filtros (Gaussianos, derivadas de Gaussianas y Laplacianos de Gaussianas) en todas las imágenes de entrenamiento y, posteriormente, se realiza un agrupamiento de *K-means*. De esta forma, cada píxel en la imagen se asigna al clúster más cercano, obteniéndose el "texton map". Esto permite no solo detectar los objetos sino que también segmentarlos. Gracias a ello, la tasa de acierto media de la segmentación es del 88,6%, pero lo más reseñable es la disminución de 30 a 1 segundos la velocidad de detección.

II.4.2.2 Modelos basados en componentes locales

La mayoría de las investigaciones en el campo de la detección de objetos se basan en modelarlos en función de sus componentes locales. Conceptualmente este modelo es mucho más sencillo de entender que el anterior: una clase de objetos está formada por una serie de características, las cuales aportan información local sobre el objeto y además están relacionadas entre sí físicamente.

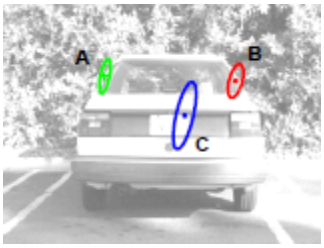


Figura II-28: Tres posibles características locales de una imagen a anotar con el concepto "coche" [WEBE00]

Como explicación, podemos ver la Figura II-28. En ella se ven tres puntos que definen tres características que supongamos definen la clase "coche". Estas características serán definidas en base a cierta información de la imagen que rodea su centro, pero esta definición dependerá del método utilizado. Además, la relación entre ellas también se puede modelar, por ejemplo a la posición relativa de las mismas. Así, una clase se definirá tanto por las características como por su relación. Por consiguiente, cuando

aparezca un nuevo coche, si el método para extraer las características es lo suficientemente robusto, se obtendrán las mismas tres características (o parecidas) y en una posición semejante; por lo que se podrá decir que ese objeto es un coche.

Dentro de esta línea de investigación hay diferentes métodos que se pueden diferenciar más o menos de la explicación anterior. Uno de los casos más claros es el modelo de "constelación de partes" propuesto en [WEBE00]. En él, los autores proponen modelar una clase de objetos mediante el uso de constelaciones flexibles de partes rígidas (donde a las partes rígidas las llaman "características"). El método que proponen se basa en que automáticamente el sistema identifica las partes distintivas de un objeto de entre las imágenes de entrenamiento. Para ello, se aplica un operador sobre dichas imágenes, que es el encargado de extraer ciertos patrones de ellas. A partir de estos patrones, se aplican técnicas de clustering que serán las encargadas de detectar las partes de un objeto de forma autónoma. Una vez se tienen las partes, el sistema extrae el modelo estadístico, mediante el algoritmo de "expectación-maximización", que lo utilizan en la fase de detección.

En este mismo principio se basó [FERG03], pero buscaba un sistema que fuese invariante a escala y a ligeras transformaciones. Para ello, también propusieron la modelización de los objetos como "constelaciones" flexibles de partes. La diferencia radicaba en que todos los aspectos del objeto (forma, apariencia, oclusión y escala relativa) se modelaban probabilísticamente, lo que aporta una mayor información de la clase y, a priori, un mayor grado de acierto. Con este método se consiguen tasas de detección que rondan el 90%, siendo especialmente destacable la tasa de acierto de peatones (96,8%) o el 94% de acierto en la detección de bicicletas. Comparativamente con [WEBE00] el resultado es mejor, donde por ejemplo [WEBE00] obtenía una tasa de detección de coches del 86,5% mientras que [FERG04] lograba un 90,4%.

El problema que posee este tipo de modelos, es que para su entrenamiento se requiere un gran número de imágenes. Por ejemplo, en [FERG03] se habla de que se han necesitado entre 200 y 800 imágenes de entrenamiento para cada clase. En general, esto era un problema, ya que la captación de tantas imágenes puede era posible para ciertas clases hasta hace unos pocos años. Para solucionar este problema, en [FEIF07] utilizan el mismo concepto de constelación de partes, pero con un enfoque diferente. En este caso, se propone la utilización de un modelo de aprendizaje incremental, de tal forma que

se permite al sistema aprender un modelo muy general a partir de pocas imágenes. Como prueba, muestran que con el entrenamiento de 1 imagen logran obtener alrededor de un 70% de aciertos, mientras que con el uso de 15 imágenes suben hasta el 80% de aciertos de media. Además, recalcan que el menor uso de imágenes de entrenamiento consigue una reducción del tiempo total de entrenamiento, lo cual siempre es un aspecto positivo.

Una aproximación más genérica que la "constelación de partes" es la detección de objetos mediante puntos característicos locales, sin modelar las relaciones geométricas entre ellos. En estos métodos, al igual que antes, se aplica un operador que detecta puntos característicos de la imagen para, posteriormente, describirlos con algún tipo de descriptor. Normalmente, este descriptor describirá la región local cercana al punto de interés extraído, por lo que la detección de un objeto se basa en comparar si los puntos extraídos y las regiones cercanas se parecen al modelo previamente aprendido.

Uno de los trabajos en esta línea es el propuesto en [AGAR02]. En este caso se define el concepto de "vocabulario", de tal forma que una clase de objetos se modelan en base a un vocabulario que define las partes de la misma. Así, todas las instancias de una clase de objeto vendrán definidas por un conjunto de palabras visuales que estén dentro de ese vocabulario visual. En la generación de este vocabulario se utiliza un operador de interés, que extrae la información importante de las imágenes de entrenamiento. Así, un vocabulario se formará utilizando todas las características obtenidas durante el entrenamiento. Como se puede ver, los vocabularios contendrán mucha información que en general modele a las instancias, siendo poca la información que sea común para toda la clase de objetos. Este concepto es muy semejante a los "bag of keypoints" o "bag of words" propuesta en [CSUR04] y [CSUR07].

Para evitar que exista tanta información dentro de uno de esos vocabularios, en [DORK05] se propone utilizar un clasificador que de todas las "partes" que se han obtenido durante el entrenamiento, sólo se seleccionen aquellas que discriminen de la mejor forma posible a la clase. Así, en vez de tener un gran número de "partes", cada clase se modela con un reducido grupo de ellas que representan de forma óptima a la clase. Esto hace que se logren tasas de acierto muy altas, en especial la detección de caras del 99,08% aunque falla en otros objetos como la de peatones, que sólo obtiene el 88% de tasa de aciertos.

Otra forma de ser más eficiente con la representación de las clases es la reutilización de las características. Este es el caso propuesto en [MIKO06], donde se permite el entrenamiento de varias clases en un mismo modelo, que es invariante a la rotación y escala. El reconocimiento de las clases se fundamenta en la utilización de un codebook. Para cada clase existe un codebook que define todas las características posibles que va a tener la clase. Este concepto es más o menos similar a los anteriores, pero la aportación de este método es que los codebooks no son disjuntos entre ellos. Es decir, las características que existen en los codebooks de las clases no son totalmente diferentes entre ellas, sino que las características se comparten en los diferentes codebooks en función de los modelos probabilísticos aprendidos durante la fase de entrenamiento. Suponiendo una tasa de acierto del 94,7% en la clase coche, mejorando el resultado de otros algoritmos del estado del arte.

Una aproximación muy semejante es la propuesta en [OMME10]. En este caso, el objeto se descompone en diferentes parches de forma no supervisada. La diferencia en este caso es que esos parches se agrupan para generar partes que componen el objeto. Es decir, primero trocean el objeto, para luego reagrupar, de tal forma que pueden lograr una mayor generalización, ya que la comparación se hace a nivel de grupos de parches y no a nivel de parches. Estas partes son las que dan una idea de la forma global del objeto, y permiten obtener un modelo estadístico que detecte los objetos. Además, utilizando ideas propuestas en otros trabajos, estas partes se pueden compartir para la detección de otros objetos, por lo que aporta ventajas a la hora de modelar varias clases con el mismo algoritmo.

Otra de las líneas de investigación existentes en la actualidad es el modelado de objetos en base a las regiones que los componen. En los casos descritos anteriormente, las partes que componen los objetos se detectan de forma automática y no tienen porqué coincidir con regiones del objeto. En este caso, las regiones son las que realmente componen el objeto, por ejemplo, en [GU09]. Aquí se propone utilizar partes reales de los objetos, que son previamente segmentadas mediante un algoritmo específico. Según ellos, las características basadas en regiones son mejores porque contienen información natural del objeto (las partes que lo componen) y además están poco influenciadas por el ruido de fondo.

Un nivel más alto de modelización de los objetos se basa en las partes reales que componen los objetos (se podría decir que son sub-objetos). Por ejemplo, en [MOHA01] detectan el cuerpo humano realizando primero una detección de brazos, piernas y cabeza, cuyos detectores están basados en las características de Haar ya mencionadas. Una de las ventajas que aporta este tipo de detección es su robustez frente a oclusiones, logrando una tasa de detección del 92%. Esto se debe a que si una parte no está visible, es posible inferir que existe, si el resto de partes están presentes en la escena. Un algoritmo más elaborado es el propuesto en [FELZ05], donde además de definir un objeto en base a los componentes que lo forman, también especifica que los componentes están unidos entre sí en una configuración deformable, a modo de "visagra". A este concepto lo denominan "Pictorical Structures", y ha sido utilizado como base para otros sistemas. De esta forma, para la detección del modelo generado en la imagen de entrada, se busca minimizar la función de energía que mida tanto el coste de relacionar cada parte como el coste de la deformación para cada par de partes conectadas. Dicho de otro modo, el coste o energía de una configuración particular depende tanto en lo bien que encaje cada parte del modelo en la imagen, como lo bien que encajen las localizaciones relativas de las partes en el modelo deformable entrenado. El punto fuerte del trabajo propuesto se encuentra en que desarrollaron un algoritmo eficiente que puede encontrar el mínimo global de la función de energía sin necesidad de inicialización.

En la actualidad, existen muchas líneas de investigación que siguen explorando algunas de las anteriores, pero que tratan de mejorar ciertos aspectos. Por ejemplo, en [LI09] se propone un nuevo modelo de aprendizaje incremental que mejora el aprendizaje sobre un gran número de imágenes. En este caso, el modelo de clase que se utiliza es el de bag-of-words, aunque dicen que es posible utilizar su algoritmo de aprendizaje sobre un modelo cualquiera gracias a él reportan una tasa de aciertos media del 74.82% cuando [FERGUS03] tenía 72%.

En [LEVI10], también se busca mejorar los sistemas de modelado de clases en base a conjuntos característicos. En general, como se ha visto, los puntos que caracterizan a una clase son seleccionados durante el entrenamiento y son utilizados en la detección. Sin embargo, en ese trabajo proponen un algoritmo que modifica los puntos que definen una clase a medida que se va utilizando. Así, durante el entrenamiento se eligen unos puntos

clave, y estos van cambiando en función de la variabilidad que se presenta durante la detección.

Otra forma que también se utiliza para mejorar es la utilización de varios modelos de los descritos anteriormente de forma conjunta para modelar una clase [VEDA09], o incluso la utilización y combinación de diferentes tipos de características [OPEL06] [GEHE09].

En el caso del concepto de "bag-of-features", también es uno de los que más se ha intentado mejorar. Este es el caso de [YANG08], donde se seleccionan las características que forman la "bag of words" de forma manual, y se integra la representación como la clasificación, permitiendo un aumento de la tasa de acierto media cercana al 20%. También en [JEGO10] tratan de mejorar el sistema para un funcionamiento óptimo en grandes bases de datos. Para ello, proponen que a cada característica del objeto se le asigne una firma binaria, por lo que se mejorará la comparación entre visual words. Además, una vez se realice la comparación de descriptores, se filtran en función de si cumplen o no ciertas restricciones geométricas.

También existen otras aproximaciones que buscan la generación de codebooks compactos, para optimizar el modelizado y la detección de los objetos [GEME10]. Y otras que proponen mezclar la segmentación de los objetos, a la par que se utilizan codebooks para una primera localización de características [LEIB08].

Por último, en la actualidad también se están dando pasos en otras direcciones diferentes a las que se han venido explorando en los últimos años. Así, por ejemplo, en [ENGE10] se ha propuesto extender el modelo de elementos finitos a la detección de objetos. En este caso, el modelo de los objetos se basa en las partes que lo componen. El modelo representa la forma de estas partes en base a los modos de vibración de los elementos finitos (finite element vibration modes). Gracias a esto, su sistema de detección de objetos se basa en buscar la forma deseada por medio de la evolución, permitiéndole ser más inmune a la oclusión y variaciones estructurales que otros métodos basados en formas. Gracias a la utilización de este modelo, el sistema requiere un entrenamiento de las posibles variaciones en la forma, pero sólo puede tener variaciones en 2 dimensiones, ya que no es muy robusto frente a cambios de orientación.

Otro ejemplo de estas nuevas investigaciones es el de [DESE10], donde se propone utilizar el concepto de "Self-Similarity" para la detección de objetos. Esta es una

propiedad de las imágenes que permite ver cuánto se parecen unas a otras. En la tarea de reconocimiento de objetos se ha estudiado recientemente, pero es en este trabajo donde se da un nuevo paso adelante, apostando por la "self-similarity" de las regiones locales de la imagen contra la imagen global (lo definen como Global Self Similarity). Para ello, en vez de comparar imágenes de objetos, compara un único parche del objeto contra todo el objeto. De esta forma, se obtiene un mapa de las zonas más similares a ese parche y, en base a esa información, realiza la detección de objetos. Además, en ese trabajo se combina el concepto de Global Self Similarity con otros conocidos como HOG [DALA05], bag of words [CSUR04] y el GIST [TORR04], de tal forma que se complementan para generar un mejor detector.

II.4.2.3 Utilización del contexto

Como se ha dicho anteriormente, la detección de las "partes" o "características" que componen un objeto es una de las líneas de investigación más destacadas en la actualidad, pero aun así hay problemas que no se pueden solucionar con este tipo de información local. Por ejemplo, dada una imagen, mediante las características locales de dos objetos, puede darse el caso en el que exista una situación de ambigüedad entre ello.

Figura II-29: Dada una imagen de satélite, observando sólo las características locales, las dos zonas recuadradas se podrían identificar como un coche. Si se atiende al contexto, se pueden identificar mejor [HEIT08]

Tal y como se ve en la Figura II-29, atendiendo únicamente a las imágenes locales de los dos elementos recuadrados, no podremos diferenciar cuál de los dos es un coche. Debido a este problema de ambigüedad, se ve que es necesario introducir otro factor que permita

diferenciar los objetos. Este factor adicional puede ser el contexto en el que se encuentra el objeto a detectar. Así, por ejemplo, si se ha detectado la presencia de un objeto (con unas características determinadas), y un sistema no puede saber si es un coche o una cafetera, el contexto en el que se sitúa la imagen le permitirá decidir cuál de los dos es. Por ejemplo, si la imagen es de satélite, se ve que el objeto “coche” tiene que estar sobre una carretera, no sobre un edificio.

Existen numerosas aproximaciones para la utilización del contexto en la ayuda a la detección de objetos. Una de las más referenciadas es la propuesta por Torralba et al. en [TORR03] [TORR04] [TORR07]. En estos trabajos, los autores buscan identificar localizaciones familiares en las imágenes y usar esta información para detectar los objetos presentes en la misma. Para llevar a cabo este análisis global, los autores han creado una representación global, de baja dimensionalidad, de la imagen. Esta representación permite obtener la información relevante que permita realizar un reconocimiento aproximado del lugar. A esta representación la denominaron la “esencia” (o “gist”) de la imagen, y buscaron poder obtenerla sin necesidad de identificar regiones u objetos dentro de ella. Para evaluar esta representación, los autores utilizaron varias clases de bojetos y escenas y mientras que la peor tasa de detección fue para la detección de personas de un 75% de aciertos, la mejor tasa de detección ha sido para los edificios con un 90% de acierto.

Además de esta información global, en [MURP06] se propone la unión de la información global con la información local obtenida por otros métodos mencionados anteriormente. La utilización de ambos métodos permite, tanto la detección de presencia de un objeto, como su la localización.

Una opción parecida es la propuesta en [HEIT08]. En ella, se realiza un segmentado de las regiones de la imagen y se trata de comprobar si es posible o no que los objetos detectados sobre estas regiones existan ahí. Esto genera un incremento de la tasa de acierto del alrededor del 3%, a excepción de la detección de personas, donde se empeora la tasa de detección en 1%.

Otra opción diferente para utilizar el contexto es la propuesta en [CARB04]. En él, se tiene en cuenta la relación existente entre los objetos presentes en una imagen. Básicamente, los autores buscan realizar varias hipótesis sobre todos los objetos presentes en una

imagen y, a partir de esas hipótesis y de la información aprendida, consideran si es probable o no que ciertos objetos estén cerca de otros o no, ayudando así al proceso de detección.

De forma similar opera [LEE10]. Este método utiliza las relaciones entre objetos conocidos para identificar otros objetos desconocidos presentes en la escena. Es este modelo de relaciones el gran aporte de este trabajo, ya que gracias a la utilización de semántica de alto nivel, el modelo propuesto permite aprender nuevas categorías de objetos basándose en categorías previas conocidas. Este aspecto permite aprender categorías de forma no supervisada, aunque también habría que tener en cuenta cómo se aprenden las categorías iniciales.

Al igual que esta opción, hay otras que también comienzan a utilizar la semántica de una forma activa para el proceso de detección de objetos. Es el caso de [GUPT08], donde se trata de formalizar las relaciones entre los objetos. Para ello, propone no sólo la utilización de “nombres” para definir las clases de los objetos sino que propone la utilización de “preposiciones” y “adjetivos comparativos”. Es decir, ante una escena, no dice que hay “carretera” y “coche”, sino que define la relación entre ellos como “coche” “encima de” “carretera”.

Otro tipo de contexto que se puede utilizar son las palabras que rodean a la imagen a analizar. Así, [HWAN10] se centra en mejorar la detección de objetos en las imágenes anotadas. Así, en función de varios parámetros de las etiquetas (como su posición y relación entre ellas) mejora la detección de los objetos.

De forma parecida es la propuesta de [JAMI10], donde tienen en cuenta los pies de foto de las imágenes. Así, además de realizar una detección de puntos característicos, éstos son asociados a las palabras que existen en los pies de foto. Con varias imágenes (y sus pies de foto correspondientes) el sistema es capaz de identificar qué palabras corresponden con las características y objetos presentes en la imagen.

Además de éstas, son otras muchas las que utilizan el contexto de diferentes maneras; como [CHOI10] donde las categorías y sus relaciones se modelan en un árbol; [GALL10] donde la información utilizada es semántica, regiones vecinas y límites de los objetos; o [BEHJ10] donde proponen un modelo de aprendizaje activo basado en las relaciones contextuales de los elementos.

En definitiva, son muchas las opciones de detección de objetos mediante componentes locales y, como se comprueba, esta es la línea de investigación que más peso ha tenido en esta temática.

II.4.2.4 Detección de objetos 3D/multivista

En general, la mayoría de las investigaciones se han centrado en la detección y modelado de objetos en una única vista, o con un rango limitado de vistas. Normalmente, a partir de un modelo válido para una vista, se realiza el entrenamiento para varias vistas. De esta forma se consigue detectar un objeto en un rango de posiciones.

A pesar de ello, estos métodos no son efectivos, ya que implica realizar un entrenamiento excesivo y repetitivo que no saca ventaja de información que relacione las vistas.

En relación a esto, son bastantes las investigaciones que están aprovechándose de estas relaciones para modelar de forma real los objetos en 3 dimensiones. Un claro ejemplo es [FERR04] donde se busca establecer una relación entre las diferentes vistas del entrenamiento, lo que reportan es muy beneficioso llegando a aumentar en 3 veces el resultado que obtenían con un detector base. Una idea parecida es la seguida por [SAVA07], en donde proponen un modelo compacto de clase en el que extraen características de las diferentes vistas de entrenamiento y las relacionan en su modelo. Sus autores reportan incrementos en la detección de hasta el doble, y en algunos casos superior, como el caso de la detección del objeto “bicicleta”, donde su sistema de detección basado en apariencia obtiene una tasa de acierto del 30% y el sistema completo 3D reporta un 80% de aciertos.

II.4.2.5 Modelo de objetos en base a silueta

Mientras que la mayor parte de los algoritmos actuales se basan en la apariencia de los objetos para su categorización, existe otra línea de investigación que aprovecha la información del contorno de los objetos. Este es el caso de [BELO02], donde se propone modelar el contexto de los puntos característicos de la forma del objeto; [OMME09] extrae puntos característicos del borde de los objetos con la finalidad de modelarlos; [FERR10] modela los objetos directamente mediante su forma, pero centrándose en realizar el matching de formas correctamente para gestionar la variabilidad intra-clase; [SHOT08] utiliza un codebook de fragmentos del contorno para el modelado de las clases.

II.4.2.6 El nuevo paradigma: Deep Learning

Dentro de este apartado II.4.2 se ha visto cómo los modelos discriminativos son capaces de detectar objetos independientes de forma satisfactoria dentro de las imágenes. Se ha mostrado cómo existen numerosas aproximaciones que tratan de representar lo más fielmente las formas, los colores, las texturas, la profundidad, etc, de los objetos, de tal manera que son capaces de lograr tasas de acierto muy cercanas a los humanos.

La problemática de estas aproximaciones es que no existe una única que sirva para todo tipo de objetos, sino que para cada problema concreto se debe utilizar una u otra. Esto ha comenzado a cambiar con la entrada en escena del concepto de "arquitectura profunda" dentro del campo de la inteligencia artificial. La teoría básica detrás de estas arquitecturas es que para que un algoritmo automático pueda aprender conceptos con un nivel de abstracción muy alto, como son los objetos presentes en una imagen, es necesario utilizar arquitecturas que se compongan de muchas capas y, por tanto, sean capaces de representar funciones altamente no lineales [BENG09]. Una arquitectura tipo relacionada con este área son las Redes Neuronales como la representada en la Figura II-30.

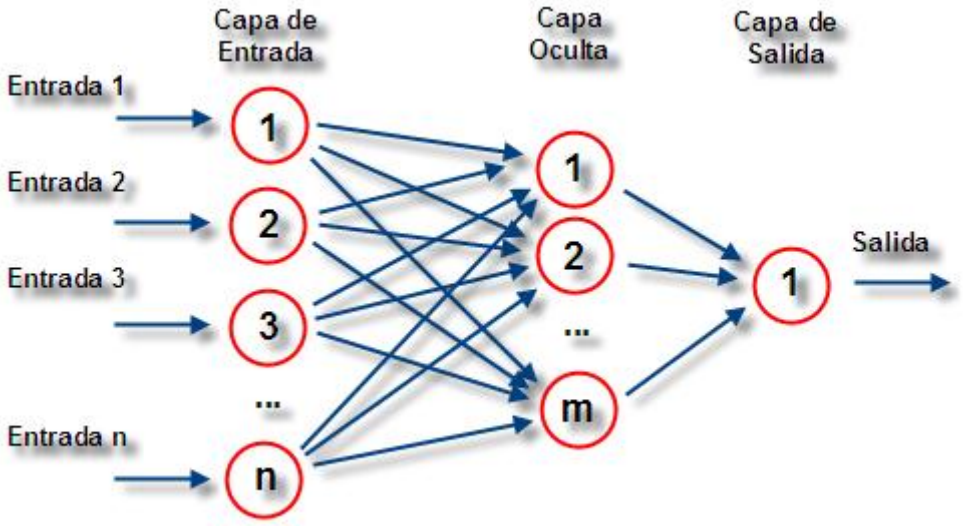


Figura II-30: Red neuronal, perceptron multicapa [RED14]

Como se puede intuir, este concepto no es nuevo, sino que se remonta hasta el comienzo de las redes neuronales en los años 80. En esos comienzos, se propusieron diferentes arquitecturas de redes neuronales con diferentes objetivos, y la forma más habitual de

entrenar a estas redes ha sido la llamada Backpropagation [RIED93]. Backpropagation se basa en calcular el gradiente de una función de error sobre los pesos de la red, y utilizar el error cometido y dicho gradiente para actualizar los propios pesos.

Este algoritmo se utilizó a lo largo de los años 90 en multitud de aplicaciones que utilizaban Redes Neuronales de pocas capas. Pero como se ha descrito, la teoría dice que se necesitan arquitecturas profundas, pero la utilización del algoritmo Backpropagation en estas arquitecturas genera un error mayor que usando arquitecturas con menos capas, lo que hizo que no se dedicase tanto esfuerzo en aquellos años a las arquitecturas profundas.

No ha sido hasta los años 2000 donde se ha visto que la utilización de Backpropagation en las arquitecturas profundas presenta varios problemas más graves [HONG10]:

- El gradiente se “diluye” en las últimas capas. Dicho de otra forma, significa que a partir de las primeras capas, la corrección del algoritmo en las sucesivas capas es inapreciable, por tanto no hay entrenamiento efectivo de dichas capas.
- El algoritmo Backpropagation se queda estancado en mínimos locales muy rápidamente cuando se usan arquitecturas de muchas capas.
- Es muy importante el uso de datos correctamente anotados, y en los años 90 no existía tanta información anotada como hoy en día.

Para abordar esta problemática, numerosos estudios han propuesto soluciones que se basan en:

- **Fase I**: un paso de entrenamiento no supervisado, también llamado pre-entrenamiento.
- **Fase II**: fase de refinamiento supervisado.

Esta aproximación tan “simple” ha sido la que ha hecho que las arquitecturas profundas logren muy buenos resultados, dando lugar al área del conocimiento que se conoce como Deep Learning, donde se busca aprender diferentes niveles de abstracción de los datos que, en global, den una información útil de los mismos.

Dentro de este área son muchos los tipos de arquitecturas propuestas, cada una con un objetivo determinado. La base de muchas de ellas son los autoencoders [BENG09], cuya forma básica posee dos capas. El primer paso es representar los datos de entrada en la

capa intermedia de una forma compacta. El segundo paso es regenerar los datos de entrada a partir de la representación anterior en la capa final. El objetivo del aprendizaje es que el error entre la entrada y la salida sea el menor. A partir de este concepto se ha creado diferentes arquitecturas como los Stacked Autencoders [BENG06].

Otras técnicas relacionadas son las Restricted Boltzman Machines [SMOL86], que son redes neuronales generativas capaces de aprender distribuciones de probabilidad, y su extensión, las Deep Belief Networks [HINT06], que permiten aprender representaciones jerárquicas de los datos de entrada de forma probabilística.

Además de éstas, otra arquitectura que está siendo muy utilizada en el campo del procesamiento de imagen es la Convolutional Neural Network (Figura II-31). Esta arquitectura se puede ver como una red neuronal clásica en la que los pesos de las neuronas de una capa se comparten. Esto quiere decir que en vez de tener un peso determinado para combinar ciertas características, todas las combinaciones de una capa se hacen con los mismos pesos. En el caso del análisis de imágenes esto es similar a la aplicación de un filtro único mediante la convolución de dicho filtro con la imagen.

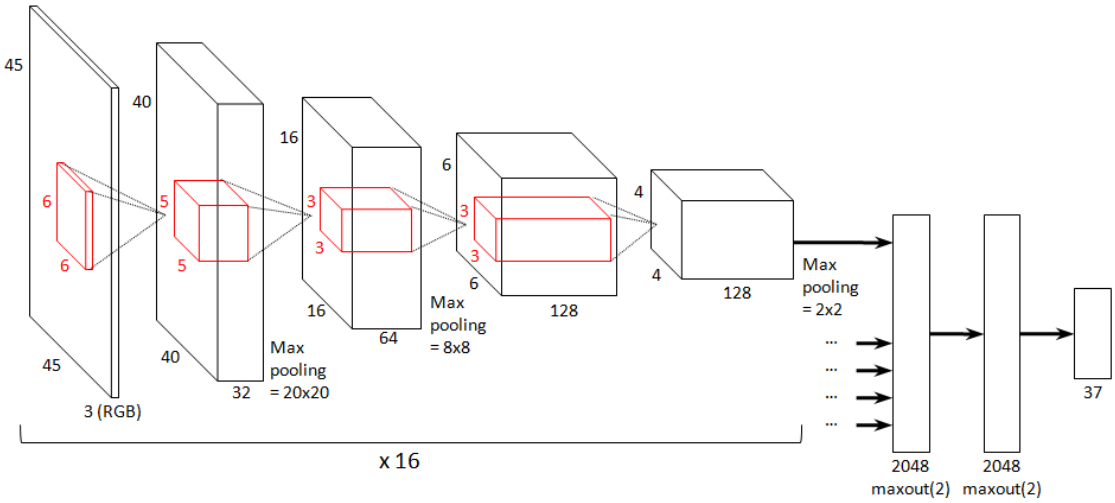


Figura II-31: Aspecto típico de una Convolutional Neural Network [WEB1]

Como se ha visto, este tipo de arquitecturas tienen sus raíces en los años 80, pero hasta el siglo XXI no se han generado los algoritmos adecuados de entrenamiento. Pero la principal motivación del uso de este tipo de algoritmos en la actualidad se debe a que ha

quedado claramente demostrado que los resultados generados en la detección de objetos a gran escala con estas arquitecturas no supervisadas supera ampliamente a lo desarrollado hasta el momento. Uno de los primeros trabajos en este sentido fue [KRIZ12], donde mostraban un mejor resultado que el estado del arte en la base de datos ImageNet 2010, logrando una tasa de fallos del 15,3% comparado con el 26,2% del segundo participante. Otro gran resultado fue el uso de una red muy similar en la competición ImageNet 2012, donde de nuevo se obtuvieron los mejores resultados hasta la fecha, reportando una mejora del 60% sobre cualquier otro sistema probado hasta la fecha en la tarea de 21.000 categorías [DEAN12]. Es más, con la misma arquitectura, Google realizó una serie de experimentos privados y reveló que dicha arquitectura tenía el doble de aciertos que cualquier otro algoritmo que ellos habían probado. Es más, en 2014 el mismo grupo de Google generó la red GoogleNet con más capas internas y más sencillas de computar [SZEG14], logrando mejorar el resultado en dicha competición con una tasa de fallos del 6,7% y quedándose a sólo un 1% de la tasa de fallos de un humano [RUSS14].

Se puede decir que esta fue la catapulta de este tipo de algoritmos en el procesamiento de imagen, ya que a partir de ese momento se han sucedido numerosos movimientos de importantes investigadores de este campo y empresas privadas: George Hinton, parte de su equipo y la tecnología fue contratado por Google en 2013 [WEB2] y estos sistemas implementados en Google+ y Google Images [WEB3], LeCun fue contratado por Facebook en 2013 [WEB4], Baidu contrató a Andrew Ng en 2014 [WEB5], mientras que otras empresas, como Yahoo/Flickr en 2013 [WEB6] o Pinterest en 2014 [WEB7] comenzaron a comprar startups que trabajaban con esta tecnología.

En el ámbito de las librerías software son muchas las que han aparecido para ayudar a trabajar con estas redes, y pueden ser Theano, CUV-RBM [WEB8] o Caffe [JIA13].

En el ámbito académico, los movimientos han sido también numerosos. Por ejemplo, de apenas estar presente en el congreso más importante de Machine Learning, NIPS 2009, a copar la mayor parte de las ponencias en NIPS 2014.

El futuro de la detección de objetos parece pasar por las técnicas no supervisadas descritas en este apartado II.4.2.6. Pero nada más lejos de la realidad, cada vez son más autores los que están trabajando con redes neuronales similares a las iniciales, y con

algoritmos de entrenamiento supervisados, la gran diferencia es usar funciones de activación nuevas y regularizadores específicos. Esto permite el entrenamiento supervisado de forma eficiente usando el algoritmo clásico de Backpropagation, sin usar los nuevos algoritmos no supervisados anteriores. Un ejemplo de las nuevas funciones de activación usadas es rectified linear units (ReLUs) (que no tienen tantos problemas "disolución" del gradiente) y el mecanismo dropout (permite evitar el overfitting), mostrando que la fase de pre-entrenamiento ya no es necesaria [DAHL13] [SRIV13].

Ésta es una técnica que está a punto de alcanzar su madurez y tendrá mucho que decir en un futuro cercano.

II.4.3 Modelos basados en vecinos más cercanos

Hasta este punto se ha mostrado cómo los modelos generativos son capaces de modelar de forma conjunta las características visuales y las etiquetas de anotación, pero también se ha mostrado cómo el estado del arte ha tendido hacia los modelos discriminativos, por dar una mayor flexibilidad al sistema final.

A pesar de ello, para la anotación de imágenes a gran escala, donde un gran volumen de etiquetas está implicado, existe una aproximación más apropiada [DENG10] [TSAI11], que es la basada en vecinos más cercanos.

Un modelo basado en vecinos más cercanos se define por tener dos etapas principales: en la primera un algoritmo de vecinos más cercanos se encarga de recolectar las imágenes de entrenamiento que sean visualmente similares a la imagen de entrada, también llamado paso K-NN (K-Nearest Neighbors). En la segunda se aplica un algoritmo de transferencia de etiquetas a las etiquetas de las imágenes anteriores, de tal forma que se seleccionen aquellas más aptas para la imagen de entrada. Este proceso queda ilustrado en Figura II-32.

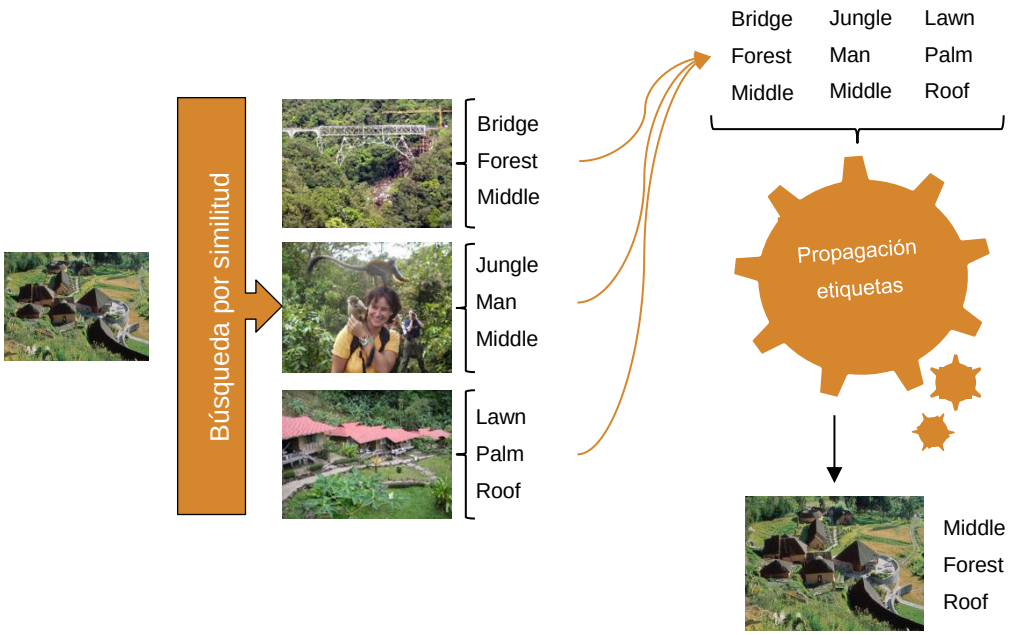


Figura II-32: Gráfico que muestra los pasos esquemáticos de un modelo de anotación basado en vecinos más cercanos

El trabajo más relevante en este ámbito es el realizado por *Makadia et al.* en [MAKA10]. En él, cada imagen se describe en base a varias características visuales y, a continuación, se miden las distancias entre la imagen de entrada y las imágenes de entrenamiento. En la etapa de transferencia de etiquetas se propone un algoritmo en el que se tienen en cuenta principalmente las etiquetas de la imagen más similar, y se añaden más etiquetas de otras imágenes similares en base a la frecuencia de aparición de las etiquetas y su coocurrencia.

Como se puede intuir, esta aproximación puede llegar a ser muy lenta, dependiendo del tipo de distancia computada y de la cantidad de imágenes en la base de datos. Para lograr un aumento de la velocidad de cómputo sin perder la precisión del resultado, *Makadia et al.* proponen la utilización de distancias simples y características simples que permitan la computación rápida del algoritmo K-NN. Las de características visuales, utilizadas son algunas de las más básicas vistas en el apartado II.2:

- **Características visuales de color**: Histograma de color en el espacio RGB, histograma de color en el espacio HSV e histograma de color en el espacio LAB.

Cada histograma se computa en 3 dimensiones, una por cada canal del color, y en cada dimensión hay 16 niveles.

- **Características visuales de textura**: Respuesta de un banco de filtros de Gabor de tres escalas y cuatro orientaciones. En vez de usar un histograma como descriptor, cada respuesta se divide en 16x16 bloques rectangulares, y se calcula el valor medio de la magnitud de cada bloque de las 12 respuestas, concatenando todas ellas para generar el descriptor final. Este mismo proceso se ejecuta para la fase de las respuestas de Gabor, cuantizando sus valores a 8 niveles. Al igual que con Gabor, también se usan tres filtros de Haar (orientación vertical, diagonal y horizontal), para computar información de textura. En este caso la fase de la respuesta es simplemente el signo de la misma, por lo que son dos únicas características.

Todos los descriptores anteriores (o *proto-descriptores*) se concatenan para generar el único descriptor de imagen.

Para la fase de cálculo de la distancia entre pares de imágenes no se utiliza una única distancia para el descriptor final. La propuesta de *Makadia et al.* se centra en calcular la distancia entre dos *proto-descriptores* de ambas imágenes y, luego, se suma de forma ponderada, denominando este método como *Joint-Equal Contribution Distance*. Las distancias utilizadas para las características visuales anteriores son la distancia L1 para todos los *proto-descriptores* y la distancia *Kullback-Leibler-Divergence* para el histograma LAB. Estas distancias han sido elegidas tras realizar evaluaciones con usuarios reales. Además de ello también han buscado rapidez en la computación de las mismas por eso se centran en distancias L1 evitando las distancias cuadráticas, como la Euclídea (ver apartado II.3.1.2).

El siguiente paso del algoritmo es la transferencia de etiquetas. Una vez se poseen un conjunto de 10 imágenes cercanas, se dispone de un conjunto de etiquetas asociado a dicho grupo de imágenes. El procedimiento ahora trata de conocer cuáles serán las etiquetas que deben ir directamente a la imagen. [MAKA10] propone un método simple pero que les ha reportado muy buenos resultados. Los pasos dados son los siguientes:

1. Del conjunto de *K=10* imágenes similares, seleccionar la imagen más similar. El primer paso es establecer un ranking de todas las etiquetas de dicha imagen. A

cada etiqueta se le asigna un valor que es el número de imágenes en el conjunto de entrenamiento en las que aparece dicha etiqueta, denominando a este valor la frecuencia de aparición en la base de datos.

2. Transferir a la imagen de entrada las *n* etiquetas con mayor rango (donde *n=5* es el número final de etiquetas a transferir). Esto sólo es posible si el número de etiquetas presentes en la primera imagen es mayor de *n*.
3. Si es menor que *n*, entonces entran en juego el resto de las imágenes del conjunto *K-1*. Cada etiqueta del dicho conjunto *K-1* se debe ordenar en base al producto de dos factores:
    a. La coocurrencia del conjunto de etiquetas en las imágenes de entrenamiento. Esta coocurrencia se define como el número de imágenes de entrenamiento en las que una etiqueta aparece con otras de este conjunto *K-1*. Evidentemente, los autores normalizan este valor por la suma de las coocurrencias de todas las etiquetas posibles.
    b. La frecuencia de aparición de cada etiqueta dentro de ese conjunto *K-1* de imágenes. Al igual que en el caso anterior, este valor se normaliza por la suma de las frecuencias de todas las etiquetas con las que se está trabajando.
4. Tras la reordenación de las etiquetas, seleccionar todas las necesarias hasta completar el total de *n* etiquetas finales (teniendo en cuenta las transferidas en el primer paso).

Se ve que las únicas características utilizadas por el algoritmo de transferencia de etiquetas son características textuales.

En el artículo [MAKA10] los autores proclaman que esta forma de proceder obtiene los mejores resultados hasta ese año en la anotación de forma automática y, para ello, lo demuestran con tres bases de datos diferentes: COREL [DUYG02], IAPR-TC12 [GRUB06] y ESP [AHN04]. En ellas las ventajas con respecto al estado del arte en aquel momento son claras, y se muestran en la Tabla II-4 y Tabla II-5:

Tabla II-4: Precisión de [MAKA10] en diferentes bases de datos

	COREL	**IAPR-TC12**	**ESP**
SotA en cada base de	0.24	0.21	0.21

datos [MAKA10]			
Makadia et al. [MAKA10]	0.27	0.25	0.23

Tabla II-5: Recall de [MAKA10] en diferentes bases de datos

	COREL	**IAPR-TC12**	**ESP**
SotA en cada base de datos [MAKA10]	0.29	0.14	0.17
Makadia et al. [MAKA10]	0.32	0.16	0.19

A pesar de su simplicidad, se comprueba que las mejoras sobre algoritmos más complejos son claras. Su principal inconveniente está en el espacio de almacenamiento en disco: en una aproximación basada en vecinos más cercanos es necesario almacenar toda la información de las imágenes. En el caso de los descriptores propuestos por *Makadia et al.*, cada imagen está representada por un vector ocupando 169KB por imagen. Si se tiene una base de datos de 1 millón de imágenes, el tamaño de la representación de la misma será excesivamente alto.

El trabajo de *Makadia et al.* se ha tomado como base del estado del arte, y ha inspirado a otros autores a proponer nuevos y mejores sistemas de anotación de imágenes. Este es el caso de TagProp [GUILL09] desarrollado por el grupo LEAR de INRIA. Aquí, en vez de seleccionar un conjunto muy limitado de información visual, se han seleccionado muchas más características de imagen que en el caso anterior. Estas características y descriptores son: GIST, histogramas de color de 16 niveles por cada canal de color en los espacios RGB, HSV, LAB; descriptores Dense SIFT agrupados en un Bag of Words y un descriptor de Tono de color. En cuanto a la etapa de transferencia de etiquetas, se ha propuesto un modelo hibrido con los sistemas discriminativos. Para ello, se han computado las distancias entre imágenes mezclando diferentes distancias (L2 para GIST, L1 para los descriptores de color y Chi-Square para el resto de descriptores), y estas distancias se han usado como pesos en los modelos discriminativos entrenados para cada etiqueta posible. Estos modelos codifican la presencia o ausencia de las etiquetas

de las imágenes y se entrenan específicamente para explotar de forma implícita las dependencias entre etiquetas en la base de datos de entrenamiento. Adicionalmente, en este trabajo se muestra y se ataja una problemática que existe en el trabajo de *Makadia et al*. y es la desviación de la anotación hacia etiquetas que se repiten mucho en la base de datos. Para evitar esta problemática, y dejar "espacio" a las etiquetas minoritarias, en *TagProp* se propone una modulación sigmoidal en cada posible palabra, logrando un aumento en el *recall* de estas etiquetas. Además de todo ello, se propone otra forma de mejorar el resultado final: la aplicación de técnicas de *Metric Learning* (II.3.2) que modifican la búsqueda de las imágenes similares. Gracias a esto se producen unas mejoras de un 16% de precisión y recall sobre [MAKA10] en la base de datos IAPR-TC12, y un 15% de precisión y 8% de recall en la base de datos ESP.

En contra de lo mostrado en [GUILL09], en [FU12] argumentan que el entrenamiento de modelos individuales no es escalable para grandes bases de datos, tal y como sucede con los modelos discriminativos. Por ello, sus autores proponen una metodología diferente para obtener las imágenes visualmente similares y las etiquetas finales sin necesidad de los modelos. El primer paso es recuperar las imágenes similares. Para ello, se propone un *Random Forest* (Figura II-33) entrenado con dos informaciones específicas:

- Información visual a través de las características visuales de las imágenes. Estas características son las mismas que en el trabajo *TagProp*, pero realizan una reducción de dimensionalidad usando *PCA* a 100 dimensiones.
- Información semántica de las etiquetas para mejorar la recuperación de las imágenes relevantes

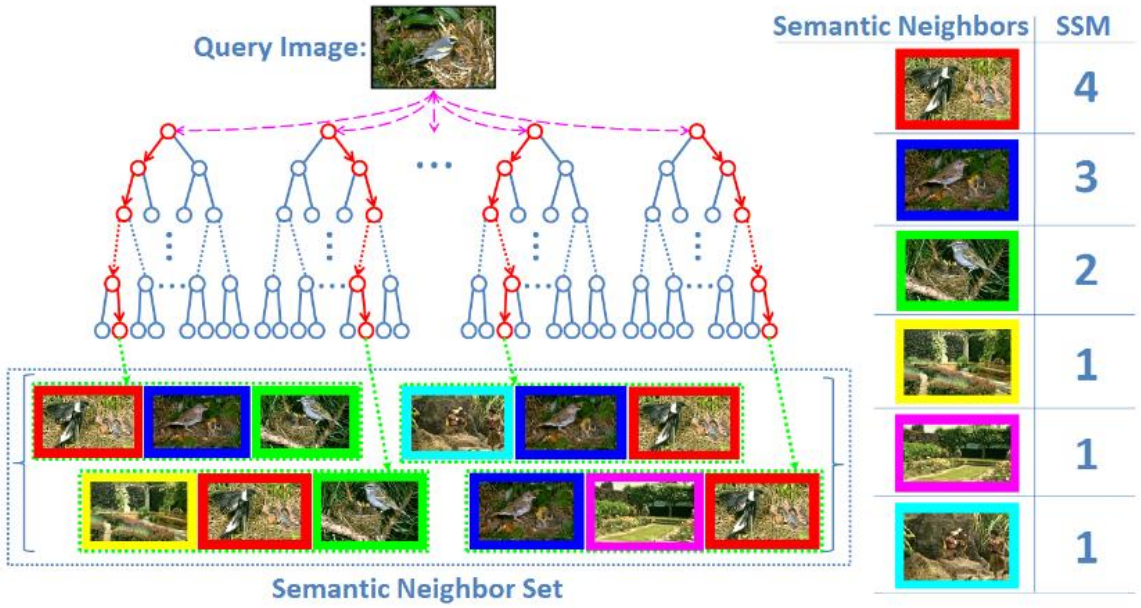


Figura II-33: Análisis de las hojas de los árboles en las que caen la imagen de query [FU12]

En cuanto a la etapa de transferencia de etiquetas, los autores de [FU12] adoptan una estrategia convencional de *tf-idf* para establecer un ranking de las etiquetas de entre todas las existentes en las imágenes similares. Para ello, ante una imagen de entrada se pasa dicha imagen por cada *random tree* hasta que llegue a un nodo hoja del árbol (Figura II-33). En estos nodos ya existen imágenes de entrenamiento, por lo que se denominan *vecinos semánticos* de la imagen de entrada. De todos los árboles del *Random Forest*, se obtienen los vecinos semánticos para generar un *set de vecinos semánticos*. La clave de este trabajo es que si dos imágenes coinciden en muchos *árboles* del *bosque* quiere decir que son imágenes muy similares. El conteo del número de bosques en los que se repiten es el coeficiente llamado *Semantic Similarity Measure* (SSM), y es el usado para la recuperación de etiquetas con el algoritmo *tf-idf*.

Estos trabajos son los que mejor representan el estado del arte actual en anotación de imágenes en base a vecinos más cercanos. En cuanto a las características visuales utilizadas en estos algoritmos, se puede ver cómo la tendencia es usar un gran número de información de descriptores para lograr un mayor grado de representatividad de la imagen.

Por otro lado, también se puede descubrir una evolución en los algoritmos de transferencia de etiquetas. Los primeros algoritmos propuestos, por ejemplo [MAKA10], sólo utilizaban información semántica de las etiquetas presentes en los vecinos más cercanos. Los siguientes algoritmos, por ejemplo [GUILL09], demostraron que, utilizando algo de información visual en la creación de los modelos de etiquetas, era posible mejorar el resultado de la anotación. Otro ejemplo es [FU12], donde tanto información semántica como visual se utiliza para recuperar las imágenes similares, pero esta información explícita no se utiliza durante la etapa de propagación de etiquetas, por lo que éste es un campo potencial a explorar y que se trabajará en esta tesis.

El hecho de disponer de toda la información posible en la base de datos hace de estos modelos los más precisos que puedan existir. Esto se debe a que, mientras los modelos generativos y discriminativos deben realizar modelos y suposiciones, perdiendo eficiencia en el proceso, un modelo basado en vecinos más cercanos tiene toda la información mapeada en el espacio.

A pesar de ello, como es evidente, estos métodos tienen una clara desventaja con respecto a los modelos discriminativos y generativos, y es que se necesita almacenar toda la base de datos para poder hacer el primer paso de computación de los vecinos más cercanos. Para evitar este problema se han propuesto diferentes soluciones, donde una de ellas es la reflejada en [TSAI11]. En ésta, sus autores han procedido a agrupar todas las imágenes del espacio de posibilidades. Esta agrupación se ha realizado en base a características visuales de forma que en un grupo o *synset* existan imágenes visualmente similares, y en base a información textual, de forma que las imágenes que aparezcan estén semánticamente relacionadas; siendo las características visuales varios histogramas de color, wavelets, LBPs y pirámides espaciales de textones sobre varias escalas.

Para ello, para cada una de las etiquetas que disponen, obtienen 1000 imágenes de Google Images, y posteriormente agrupan dichas imágenes en grupos homogéneos en base a su información visual y planteando el problema como un problema típico de agrupación o *clustering*. Con estos dos pasos el concepto no se representa por un único grupo de imágenes (o *synsets*), sino por varios grupos visualmente homogéneos. Además, esto provoca que un único *synset* no represente a un único concepto. Es esta

propiedad la que los autores utilizan de forma que se aprendan unos pesos para cada *synset* que se utilizarán a la hora de la votación para la anotación de una imagen.

Para predecir la anotación, los autores utilizan de nuevo una aproximación más similar a los algoritmos discriminativos: para cada *synset* se entrena una máquina SVM lineal. Ante una imagen de entrada, se evalúan todas las máquinas, y los resultados que pasen de un umbral se utilizan para la votación. De todos los *synsets* aceptados se extraen las etiquetas, y los resultados de los SVM son los que se combinan linealmente, usando los pesos anteriormente calculados, para votar a las etiquetas finales. Esto genera un resultado muy positivo que han comparado contra dos métodos que son uno discriminativo genérico y uno de vecinos más cercanos genérico. Así, el método propuesto obtenía una precisión media del 2,66% mientras que el método discriminativo era de sólo del 0,67% y el método de vecinos más cercanos era de más del doble: 1,87%.

II.5 Más allá de la información visual: la contextualización de las imágenes

A pesar de las mejoras en el procesamiento de imágenes, existe información adicional que puede utilizarse para la anotación de las mismas. En ciertos ámbitos, las imágenes no están solas, sino que forman parte de un documento superior, como puede ser un libro, un artículo o una página web. Entonces, además de la información visual, se puede extraer mucha más información conociendo el entorno que rodea a una imagen y que se denomina “contexto de la imagen”.

Si nos centramos en el entorno web, este contexto es capaz de mejorar la anotación automática de imágenes gracias a que permite desambiguar los conceptos a anotar. En este apartado se mostrarán dos de las tecnologías más relevantes de contextualización web. La primera de ellas se basa en las propias “etiquetas” que están relacionadas con las imágenes, mientras que la segunda trata de un nuevo modelo de contextualización que se basa en el auge de las redes sociales.

II.5.1 Contextualización en base a etiquetas

Uno de los sistemas de contextualización que más se están trabajando en la actualidad es la relación de una imagen con sus etiquetas asociadas en la página web. Un ejemplo claro de este tipo de páginas web es Flickr, donde cada usuario puede subir sus imágenes y etiquetarlas en función de sus intereses personales. Gracias a la importancia

de este sitio web, están surgiendo numerosas iniciativas alrededor de Flickr [ULGE11]. Una de las más innovadoras es el hecho de utilizar la estructura del contenido de la web que proporcionan los usuarios. Por ejemplo, la comunidad Flickr organiza sus fotos en base a los grupos de Flickr. Actualmente, existen más de 200.000 grupos que han sido definidos y relacionados con todo tipo de temas, como “fotos de fiestas” o “fotografía natural”. La utilización de los grupos se centra en la fase de aprendizaje del sistema, donde para cada grupo se genera un modelo. Durante la anotación, la información del grupo se asume como proporcionada por el usuario, es decir, utilizan los grupos como fuente adicional de información que ayuda principalmente en la desambiguación [ULGE11].

Otros autores consideran que los resultados obtenidos de Flickr son muy “ruidosos”, es decir, se obtienen muchas imágenes que no tienen mucho o nada que ver con la imagen anotada. Este fenómeno se puede comprobar en la Figura II-34, donde se ilustran los resultados ante la búsqueda con la palabra “perro”: entre los primeros 36 resultados obtenemos cuatro que no aplicarían, es decir, el 11%.

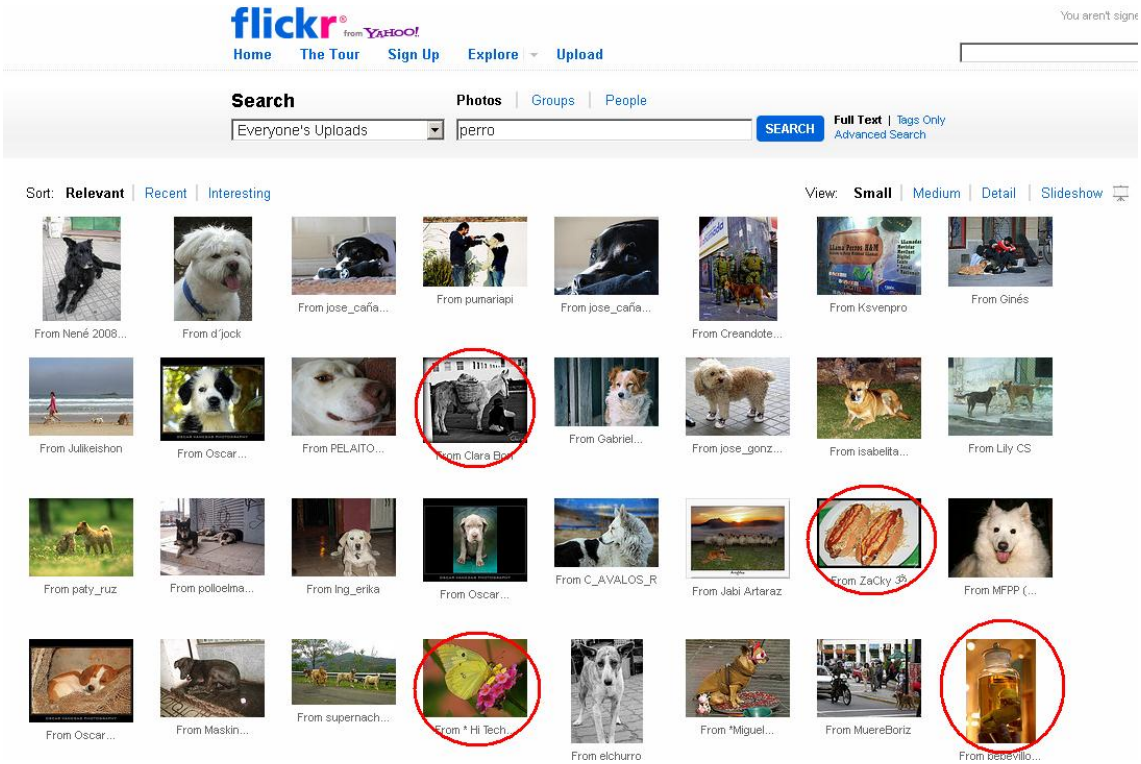


Figura II-34: Resultados en Flickr para búsqueda con la palabra "perro". Se han marcado imágenes donde el sistema falla.

Por ello, cuando se utiliza Flickr para construir una base de datos de entrenamiento para la anotación de imágenes, se hace necesario hacer un filtrado previo, aplicando la máxima de que una imagen es relevante para una cierta etiqueta cuando la etiqueta describa el contenido de una o más regiones de la imagen [TANG11].

II.5.2 Contextualización en base a redes sociales

Otras iniciativas utilizan las redes sociales [ELHA09] para adquirir los metadatos contextuales sociales. Actualmente, las redes sociales son una de las plataformas más utilizadas por las comunidades online para compartir texto, imágenes y vídeos. Las redes sociales se basan en perfiles de usuarios que ofrecen una descripción de cada miembro. Además de las imágenes subidas por un usuario, su perfil contiene comentarios y opiniones positivas/negativas sobre dichos recursos.

Un trabajo de análisis del contexto más complejo es el llevado a cabo en [ELHA11]. En este trabajo se propone un sistema que genera de forma semi-automática anotaciones de imágenes sobre la ontología OntoCAIM, considerando el contexto de una red social y haciendo uso de las anotaciones manuales de imágenes proporcionadas por el usuario más activo. Para ello, generan a este usuario tipo mediante Social Network Analysis y toman sus anotaciones como base para las imágenes objetivo.

La ontología OntoCAIM reutiliza las ontologías FOAF (representa el perfil del usuario de la red social), SIOC (representa comunidades online), EXIF (representa metadatos básicos de fotografías) y WordNet (ayuda en la desambiguación del lenguaje natural), y por lo tanto describe la red social así como las imágenes. Como se ha comentado, este sistema gira en torno al contexto social de una imagen, y dicho contexto estaría formado, entre otros, por la geo-referencia, fecha y hora y la granularidad de las relaciones entre actores de la red social:

- Para obtener el actor central o actor correcto utilizan técnicas de SNA (Social Network Analysis).
- La fecha y hora es utilizada porque se parte de la hipótesis de que fotografías que hayan sido sacadas dentro de un corto espacio de tiempo, seguramente sean muy similares porque la cámara no puede haber estado en localizaciones muy distantes en el umbral de 5 minutos que plantean inicialmente.

- La geo-referencia se utiliza para agrupar fotografías tomadas dentro de un radio concreto, siempre y cuando las fotografías contengan en sus metadatos las coordenadas GPS.

Otra opción de utilización de las redes sociales como ayuda a la anotación es en el caso del reconocimiento facial para la anotación. En el trabajo [STON08] se propone mejorar los algoritmos de reconocimiento utilizando las redes sociales. Así, una imagen facial puede no ser reconocida por diversos motivos (la iluminación o el maquillaje). En este caso, se utilizan imágenes parecidas anotadas de forma manual en las redes sociales y se analizan las personas que aparecen en la misma imagen, identificando grupos de amigos. De esta forma la tasa de anotaciones automáticas de personas aumenta considerablemente gracias al uso del contexto social.

II.6 Sistemas de búsqueda de imágenes comerciales y retos futuros

En anteriores apartados se ha visto cómo es posible anotar de forma automática el contenido de una imagen. El objetivo final de cualquier tipo de anotación de imagen es permitir a los motores de búsqueda la recuperación de las mismas de forma inteligente. Existen numerosos intentos por crear buenos buscadores de imágenes basados en su contenido. Algunos de los más avanzados, en el momento de la escritura, se pueden encontrar en la tabla siguiente.

Tabla II-6: Referencias de motores de búsqueda comerciales de imágenes

Buscador de imágenes	Referencia web
TinEye	http://www.tineye.com
Cydral	http://www.cydral.com
Quintura (y versión Kids)	http://www.quintura.com
Google Similar Images	http://images.google.com
Ithaki	http://www.ithaki.net
LTU	http://www.ltutech.com
Pixlogic	http://www.pixlogic.com

Dentro de ellos hay que destacar el buscador *Google Images* en su versión de búsqueda de imágenes similares. En este caso el usuario le indica al buscador una imagen y éste le devuelve información relevante para ella. Tras realizar numerosas pruebas sobre el buscador (a fecha de 2012), se ha comprobado que es capaz de realizar búsquedas de muy distintos tipos dependiendo de la imagen subida. Por ejemplo, se ha visto cómo si se le indica la imagen frontal de un coche es capaz de reconocer la marca y modelo, así como su color.

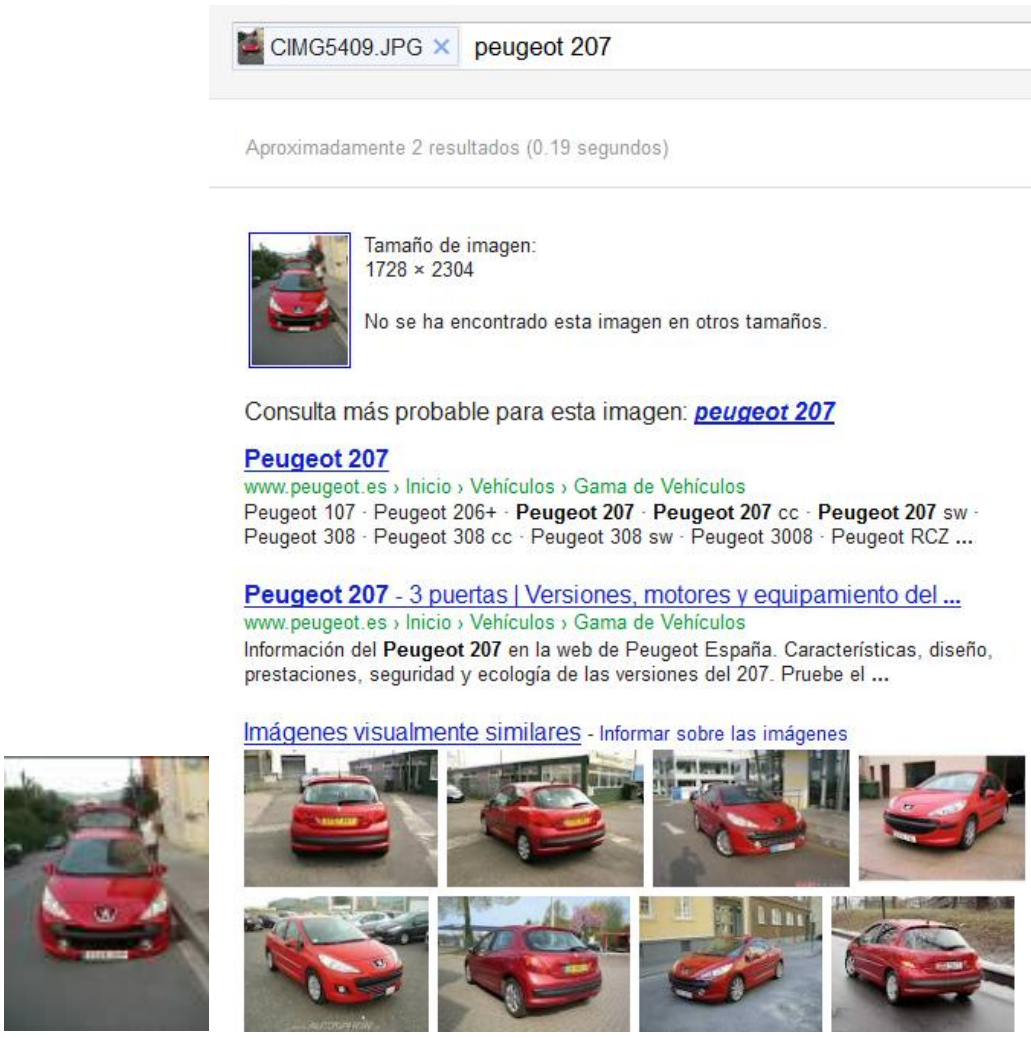


Figura II-35: Búsqueda en Google Images de una imagen de un coche (izquierda). A la derecha se muestra el resultado, detectando la marca y modelo del mismo automáticamente

En cambio, si se le muestra el mismo coche en una vista lateral, sólo intuye que es un vehículo y devuelve imágenes genéricas de vehículos. Esto hace ver que los buscadores comienzan a anotar los contenidos para su búsqueda, pero el resultado no es del todo correcto, ya que existen casos en los que se buscan imágenes de un determinado elemento y aparecen imágenes de otros que no esperamos. Este fenómeno está relacionado con la experiencia de uso, que no es satisfactoria porque el usuario espera otra información.

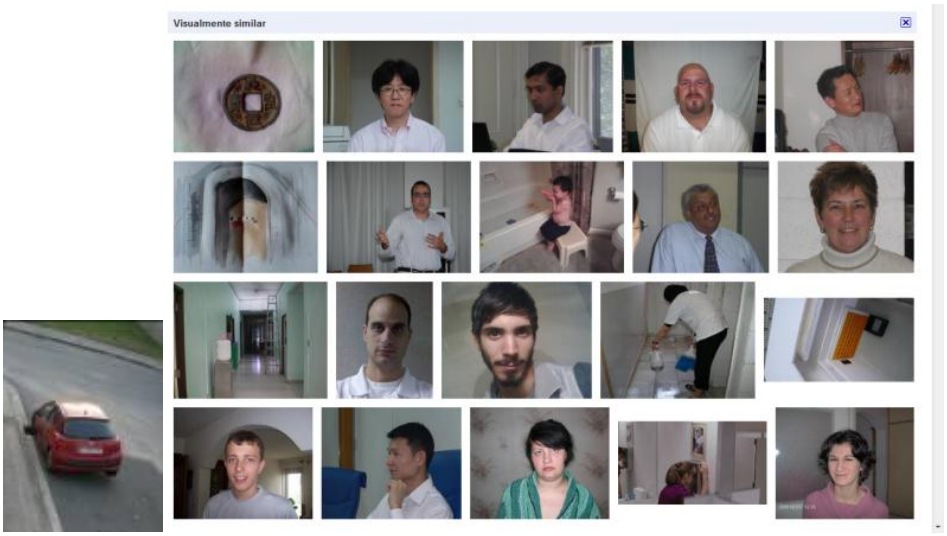


Figura II-36: Búsqueda en *Google Images* de una imagen de un coche (izquierda). A la derecha se ve cómo no ha sido capaz de reconocer la entidad *coche*, y la ha confundido con la entidad *cara*.

Por ello, como retos futuros en el ámbito de recuperación inteligente de imágenes se puede destacar la necesidad de avanzar en herramientas de medición de la satisfacción de la experiencia del usuario, considerando sus expectativas y estado emocional. El resultado de estas mediciones debería influir en el proceso de búsqueda y presentación de los resultados. Asimismo, la anotación de imágenes posee un factor de medición subjetivo, es decir, esperamos que la anotación sea tan buena como aquella que realizaría un ser humano. Por ello, esta subjetividad influye de forma determinante en la recuperación de información, cuanto mejor sea la anotación, mejor será la recuperación de información. Pero, si además se une a la evolución de la capacidad de medición de la calidad de la experiencia - aprendiendo qué es lo que cada usuario espera en cada ocasión, la recuperación inteligente de información daría un paso de gigante.

III. Análisis del Estado del Arte y definición del baseline

III.1 Introducción

En los trabajos de investigación, el estado del arte queda establecido por las publicaciones más recientes que aparecen en las revistas y congresos científicos de mayor impacto. Estas publicaciones pueden ser de dos tipos principales: a) publicaciones que poseen una alta relevancia por la novedad o el avance del método o algoritmo propuesto, con unos resultados ligeramente mejores que los obtenidos hasta la fecha; b) publicaciones que poseen una alta relevancia por lo importante de los resultados mostrados, aun siendo de menor importancia el propio método propuesto. En ciertos campos de la ciencia y la ingeniería, como por ejemplo la neurociencia o la biología, las publicaciones de una línea de investigación determinada son seguidas y generalmente se dan por válidas, y se trabaja a partir de ellas, para refutarlas o aceptarlas.

En el campo de la anotación de imágenes existen numerosos grupos de investigación en todo el mundo trabajando en múltiples líneas de trabajo, paralelas, ortogonales y, en otras ocasiones, complementarias. La proliferación de congresos y revistas de impacto en estos temas hace que no sea posible definir un conjunto de algoritmos punteros acotado y, como se ha visto en el capítulo II, el estado del arte existente es inmensamente amplio. La mayoría de los trabajos buscan la novedad ante todo, y el grueso de la comunidad de investigación busca, año a año, generar un algoritmo nuevo sin analizar en detalle otras propuestas. Esto ha derivado en un campo que tiene multitud de líneas de investigación “quemadas” por publicación de trabajos con grandes resultados un año, pero al año siguiente deja de trabajarse en esa línea. Lo mismo sucede con muchos grupos de investigación, los cuales año a año trabajan en algoritmos completamente diferentes con el mismo objetivo de anotar imágenes.

En este capítulo se pretende analizar en detalle el estado del arte para establecer una base que defina el estado actual de la técnica. En primer lugar, se mostrarán las bases de datos de referencia más comunes en las investigaciones del estado del arte, y se seleccionarán aquellas más adecuadas para esta tesis. A continuación, se especificarán las métricas de validación a utilizar, y que se seguirán tanto en este capítulo como en el resto de la tesis. Posteriormente, se mostrarán diferentes pruebas y resultados con algoritmos del estado del arte, y por último, se definirá la base sobre la que esta tesis pretende mejorar.

III.2 Bases de datos de referencia

En el campo de la anotación de imágenes y detección de objetos existe una gran cantidad de bases de datos públicas de referencia. Cada base de datos tiene unas propiedades y objetivos diferentes, y todas han evolucionado a lo largo del tiempo. Dos de las primeras bases de datos de detección de objetos únicos en una imagen fueron Caltech-101 [FEIF04] y Caltech-256 [GRIFF06]. En ambos casos, se trata de un grupo de imágenes que representan a un objeto concreto en escenas genéricas. La primera tiene 100 objetos y la segunda 255. Algunos ejemplos de imágenes de estas bases de datos se pueden ver en la Figura III-1.

Figura III-1: Diferentes imágenes de la clase "baseball-glove" de Caltech256

Desde estas pequeñas bases de datos, la comunidad ha evolucionado hacia bases de datos más complejas y grandes. En el campo de la detección de “objetos” o “conceptos” únicos en una escena nos encontramos con la base de datos SUN [XIAO10], que posee más de 130.000 imágenes clasificadas en 899 categorías, desde elementos urbanos como “catedral”, hasta localizaciones de interior como “oficina de policía”. Otro ejemplo es la base de datos ImageNet [DENG09], que es la mayor base de datos de imágenes existente. En su versión de 2010 posee más de 14 millones de imágenes con cerca de 22.000 objetos.

Además de estas bases de datos estáticas, existen competiciones de detección de objetos, donde la más destacada es “*The PASCAL Visual Object Classes (VOC) Challenge*”, y desde su finalización en 2012 la gran competición de referencia es “*ImageNet Large Scale Visual Recognition Challenge*”, asociada a la base de datos anterior.

La competición VOC se celebraba todos los años desde 2005 hasta 2012 con diferentes objetivos, entre ellos, la detección de presencia de un tipo de objeto en una imagen. Para realizar esta competición, cada año se proponía una base de datos diferente, que era la que usaban todos los participantes. Esta base de datos estaba compuesta por una serie de 20 objetos, y para cada objeto existía un conjunto de imágenes de entrenamiento, otro conjunto de imágenes de validación, mientras que la competición se realizaba sobre un tercer conjunto de imágenes de test.

El procedimiento de la competición establecía que, antes de la misma, se ponían a disposición de los competidores las bases de datos de entrenamiento y validación, de tal forma que podían optimizar sus detectores de objetos. Por otra parte, la base de datos de test sólo se utilizaba para realizar las pruebas en el momento que se determine en la competición.

En cuanto al conjunto de objetos a detectar, en los años 2005 y 2006 fueron diferentes, aunque a partir del año 2007 se establecieron definitivamente las categorías a detectar, y se muestran en la Tabla III-1.

Tabla III-1: Las clases de objetos a detectar en la competición The PASCAL Visual Object Classes Challenge"

Aeroplane	Bicycle	Bird	Boat	Bottle
Bus	Car	Cat	Chair	Cow
Diningtable	Dog	Horse	Motorbike	Person
Pottedplant	Sheep	Sofa	Train	Tvmonitor

A pesar de que cada año de la competición se definía una nueva base de datos, es una práctica muy habitual en el campo de la detección de objetos el utilizar la base de datos de 2007 como referencia para futuros trabajos, principalmente porque es el año en que se establecieron las categorías de objetos a detectar definitivos y, además, se generó una

publicación explicando la base de datos y comparando los participantes de dicho año [EVER10].

Por otra parte, además de bases de datos con un único objeto en las imágenes, también existen bases de datos que se enfocan a la detección de múltiples objetos en las imágenes, como por ejemplo COREL Database [DUYG02] o ESP Game [AHN04].

Muchos de los trabajos actuales, en cambio, se centran en otras dos bases de datos. La primera de ellas es la colección "*Segmented and Annotated IAPR TC-12*" [GRUB06], mientras que la segunda es MIRFlickr [HUISK08], más específicamente el subconjunto elegido dentro de la competición ImageCLEF 2011. Algunos ejemplos de las imágenes existentes en esta última se pueden ver en la Figura III-2.

Figura III-2: Ejemplos de imágenes de la base de datos MIRFlickr

En el caso de la base de datos SAIAPR-TC12, está compuesta por unas 20.000 imágenes de escenas naturales, deportes, personas, animales, ciudades, etc. Junto a las imágenes de esta base de datos se encuentran disponibles las etiquetas textuales correspondientes a los objetos que aparecen en las mismas y que son un subconjunto de las 275 posibles etiquetas. Estas etiquetas son totalmente genéricas, en idioma inglés, y pueden ir desde medios de transporte como "airplane" hasta otros conceptos como "snow" (Tabla III-2). Además, esta base de datos proporciona la segmentación de las diferentes zonas de la imagen, tal y como se puede ver en la Figura III-3.

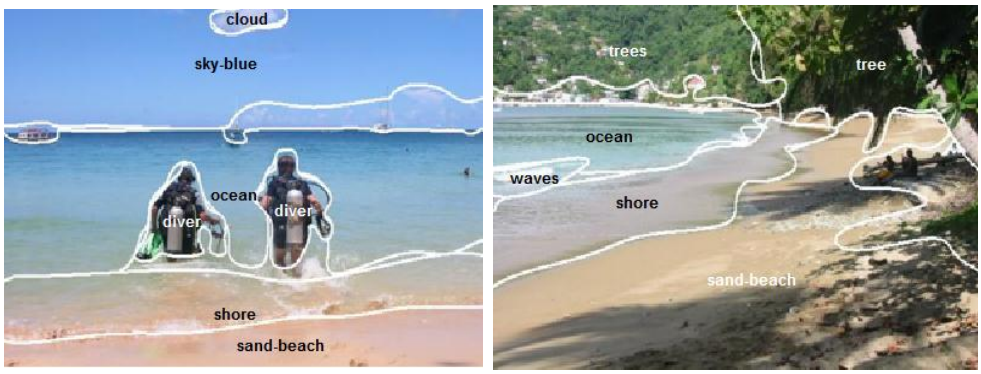


Figura III-3: Imágenes de ejemplo de la base de datos SAIAPR TC-12

Esta base de datos también incluye la definición del conjunto de imágenes de entrenamiento y de test, para que todas las investigaciones se realicen bajo las mismas condiciones, de tal forma que se tienen 17.825 imágenes de entrenamiento y 1.980 de test.

Tabla III-2: Posibles etiquetas de la base de datos SAIAPR TC-12, junto con su identificador numérico

1-; 2-aerostatic-balloon; 3-air-vehicles; 4-airplane; 5-ancent-building; 6-animal; 7-ant; 8-antelope; 9-ape; 10-apple; 11-arctic; 12-astronaut; 13-baby; 14-ball; 15-balloon; 16-beach; 17-bear; 18-beaver; 19-bed; 20-beetle; 21-bench; 22-bicycle; 23-bird; 24-boat; 25-boat-rafting; 26-bobcat-wildcat; 27-book; 28-bottle; 29-branch; 30-bridge; 31-building; 32-bull; 33-bus; 34-bush; 35-butterfly; 36-cabin; 37-cactus; 38-camel; 39-camera; 40-can; 41-canine; 42-cannon; 43-car; 44-caribou; 45-castle; 46-cat; 47-caterpillar; 48-cello; 49-chair; 50-cheetah; 51-child; 52-child-boy; 53-child-girl; 54-chimney; 55-church; 56-church-interior; 57-city; 58-clock; 59-cloth; 60-cloud; 61-column; 62-construction; 63-construction-other; 64-coral; 65-cougar-puma; 66-couple-of-persons; 67-cow; 68-coyote; 69-crab; 70-crocodile; 71-cup; 72-curtain; 73-deer; 74-desk; 75-dessert; 76-dish; 77-diver; 78-dog; 79-dolphin; 80-door; 81-dragonfly; 82-eagle; 83-edifice; 84-elephant; 85-elk; 86-entity; 87-fabric; 88-face-of-person; 89-feline; 90-fence; 91-fire; 92-firework; 93-fish; 94-flag; 95-flamingo; 96-flock-of-birds; 97-floor; 98-floor-carpet; 99-floor-other; 100-floor-tennis-court; 101-floor-wood; 102-flower; 103-flowerbed; 104-food; 105-fountain; 106-fowl-hen; 107-fox; 108-fruit; 109-furniture; 110-furniture-other; 111-generic-objects; 112-giraffe; 113-glacier; 114-glass; 115-goat; 116-grapes; 117-grass; 118-ground; 119-ground-vehicles; 120-group-of-persons; 121-guitar; 122-hand-of-person; 123-handcraft; 124-hat; 125-hawk; 126-head-of-person; 127-hedgehog-porcupine; 128-helicopter; 129-herd-of-mammals; 130-

highway; 131-hill; 132-horn; 133-horse; 134-house; 135-humans; 136-hut; 137-ice; 138-iguana; 139-insect; 140-island; 141-jaguar; 142-jewelry; 143-kangaroo; 144-kitchen-pot; 145-koala; 146-lake; 147-lamp; 148-landscape-nature; 149-leaf; 150-leopard; 151-lighthouse; 152-lion; 153-lizard; 154-llama; 155-lobster; 156-log; 157-lynx; 158-mammal; 159-mammal-other; 160-man; 161-man-made; 162-man-made-other; 163-mandril; 164-marsupial; 165-monkey; 166-monument; 167-motorcycle; 168-mountain; 169-mural-carving; 170-mushroom; 171-musical-instrument; 172-nest; 173-non-wooden-furniture; 174-ocean; 175-ocean-animal; 176-octopus; 177-orange; 178-other-entity; 179-owl; 180-pagoda; 181-painting; 182-palm; 183-panda; 184-paper; 185-parrot; 186-penguin; 187-person; 188-person-related-objects; 189-piano; 190-pigeon; 191-plant; 192-plant-pot; 193-polar-bear; 194-primate; 195-public-sign; 196-pyramid; 197-rabbit; 198-rafter; 199-railroad; 200-reptile; 201-rhinoceros; 202-river; 203-road; 204-rock; 205-rodent; 206-roof; 207-rooster; 208-ruin-archeological; 209-sand-beach; 210-sand-dessert; 211-saxophone; 212-school-of-fishes; 213-scorpion; 214-screen; 215-seahorse; 216-seal; 217-semaphore; 218-sheep; 219-shell; 220-ship; 221-shore; 222-sidewalk; 223-sky; 224-sky-blue; 225-sky-light; 226-sky-night; 227-sky-red-sunset-dusk; 228-smoke; 229-snake; 230-snow; 231-space-shuttle; 232-squirrel; 233-stairs; 234-starfish; 235-statue; 236-steam; 237-strawberry; 238-street; 239-sun; 240-surfboard; 241-swimming-pool; 242-table; 243-telephone; 244-tiger; 245-tire; 246-tower; 247-toy; 248-train; 249-trash; 250-tree; 251-trees; 252-trombone; 253-trumpet; 254-trunk; 255-turtle; 256-umbrella; 257-vegetable; 258-vegetation; 259-vehicle; 260-vehicles-with-tires; 261-viola; 262-violin; 263-volcano; 264-wall; 265-water; 266-water-reflection; 267-water-vehicles; 268-waterfall; 269-waves; 270-whale; 271-window; 272-wolf; 273-woman; 274-wood; 275-wooden-furniture; 276-zebra;

Por otro lado, la base de datos MIRFlickr y, concretamente, el subset de ImageCLEF2011 está enfocada a proporcionar un gran conjunto de datos para la anotación multi-etiqueta. La metodología de evaluación de esta base de datos se basa en un conjunto estándar de test, formado por 10.000 imágenes que contienen anotaciones de 99 conceptos visuales. Estos conceptos son de muy diversa índole, ya que existen grupos de etiquetas abstractos como “Underexposed”, que trata sobre las condiciones de toma de la propia imagen, y otros conceptos más físicos como “ship”. En cuanto al conjunto de entrenamiento, se compone de 8.000 imágenes de todo tipo con anotaciones manuales.

Figura III-4: Imágenes de ejemplo de la base de datos de ImageCLEF 2011

Las 99 posibles anotaciones de esta base de datos son las de la Tabla III-3.

Tabla III-3: Posibles etiquetas del conjunto ImageCLEF2011 junto con su identificador numérico

0-Partylife; 1-Family_Friends; 2-Beach_Holidays; 3-Building_Sights; 4-Snow; 5-Citylife; 6- Landscape_Nature; 7-Sports; 8-Desert; 9-Spring; 10-Summer; 11-Autumn; 12-Winter; 13-Indoor; 14-Outdoor; 15-Plants; 16-Flowers; 17-Trees; 18-Sky; 19-Clouds; 20-Water; 21-Lake; 22-River; 23-Sea; 24-Mountains; 25-Day; 26-Night; 27-Sunny; 28-Sunset_Sunrise; 29-Still_Life; 30-Macro; 31-Portrait; 32-Overexposed; 33-Underexposed; 34-Neutral_Illumination; 35-Motion_Blur; 36-Out_of_focus; 37-Partly_Blurred; 38-No_Blur; 39-Single_Person; 40-Small_Group; 41-Big_Group; 42-No_Persons; 43-Animals; 44-Food; 45-Vehicle; 46-Aesthetic_Impression; 47-Overall_Quality; 48-Fancy; 49-Architecture; 50-Street; 51-Church; 52-Bridge; 53-Park_Garden; 54-Rain; 55-Toy; 56-MusicalInstrument; 57-Shadow; 58-bodypart; 59-Travel; 60-Work; 61-Birthday; 62-Visual_Arts; 63-Graffiti; 64-Painting; 65-artificial; 66-natural; 67-technical; 68-abstract; 69-boring; 70-cute; 71-dog; 72-cat; 73-bird; 74-horse; 75-fish; 76-insect; 77-car; 78-bicycle; 79-ship; 80-train; 81-airplane; 82-skateboard; 83-female; 84-male; 85-Baby; 86-Child; 87-Teenager; 88-Adult; 89-old_person; 90-happy; 91-funny; 92-euphoric; 93-active; 94-scary; 95-unpleasant; 96-melancholic; 97-inactive; 98-calm

III.3 Métricas de evaluación de las pruebas

A lo largo de esta tesis se van a presentar muchos resultados conseguidos en base a las métricas presentadas en este apartado. Para generar los resultados de una prueba, las bases de datos anteriores poseen un conjunto de *entrenamiento* y otro de *test*. Así, con el primero se entrenarán los parámetros necesarios y con el segundo se obtendrán los resultados con el segundo conjunto.

Para mostrar y comparar los resultados, las métricas más comunes son *precision* y *recall*, cuya definición es la siguiente:

$$precision = \frac{numero\_positivos\_correctos}{positivos\_totales\_predichos}$$

$$recall = \frac{numero\_positivos\_correctos}{positivos\_totales\_reales}$$

$$F-measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Entonces, para cada etiqueta se realizan ambas medidas, y el resultado de toda la base de datos es el promedio de todos los valores sobre el conjunto de las imágenes. Con estos valores es posible comparar varios algoritmos para seleccionar aquel que posea mayor *precision*, *recall* o *F-measure*, dependiendo del objetivo. Por defecto todos los algoritmos poseen un parámetro de configuración que permite ajustar la sensibilidad de detección (bien sea en forma de umbral u otros). Debido a esto el *precision* y *recall* varía en función de ese parámetro. Para poder comparar sistemas de forma más fiable, se generan estos dos valores para múltiples valores de configuración, generando las curvas de *precision-recall*.

Además de estos tres valores, también existen otras métricas que se usan comúnmente, y es necesario definir para poder comparar los valores generados con los del estado del arte. Así, las medidas propuestas en esta base de datos son:

- **MAP** (Mean Average Precision): es la media para todas las etiquetas del área bajo la curva *precision-recall*. Cuanto mayor sea su valor significa que es más cercano a la curva ideal con valor máximo de 1.
- **EER** (Equal-Error Rate): es el punto en el que el número de aceptados y rechazados es el mismo. Esta definición hace que cuanto menor sea su valor, el resultado posee menos errores y por tanto es mejor.
- **AUC** (Area Under the Curve): el área bajo la curva ROC (Receive Operating Curve).

Por último, cabe mencionar dos valores que son utilizados en ciertas ocasiones, sobre todo en los modelos discriminativos:

- **FAR** (False Aceptance Ratio): indica la cantidad de veces que se dice que un objeto está presente en la imagen cuando no lo está.
- **FRR** (False Rejection Ratio): indica la cantidad de veces que se dice que un objeto no está presente y realmente lo está.

III.4 Pruebas y análisis de resultados

Como se ha visto en el capítulo del estado del arte, existen dos grandes alternativas para la anotación de imágenes. Por un lado, se pueden usar los modelos discriminativos, que entrenan un clasificador por cada etiqueta posible y luego ejecutan todos ellos para las imágenes de test. Por otro lado, está la aproximación basada en vecinos más cercanos, donde primero se hace una búsqueda de imágenes más similares y posteriormente se ejecuta un algoritmo de propagación de etiquetas.

Dependiendo de la aplicación concreta, la comunidad tiende a trabajar en una línea o en otra. En este apartado se buscará ver los resultados generados por los algoritmos más ampliamente aceptados y utilizados del estado del arte, para poder decidir en el siguiente apartado cuál es realmente la base más útil para el objetivo de anotación automática de imágenes.

Como se ha visto en el capítulo II, cada artículo ha utilizado bases de datos diferentes, por lo que las comparativas entre métodos son complicadas. En este apartado se seleccionará una base de datos común para cada prueba a realizar en esta tesis y se re-implementarán los algoritmos del estado del arte. Con esto se pretenderá hacer una comparativa eficaz.

III.4.1 Análisis de modelos discriminativos

Los algoritmos discriminativos se basan en el entrenamiento de un modelo visual por cada etiqueta a detectar. En el capítulo II se ha visto que el algoritmo *Bag of Visual Words*, descrito por Csurka et al. en 2004 [CSUR04], es uno de los más usados en el estado del arte, ya que su simplicidad, velocidad y modularidad permiten hacer múltiples variaciones que mejoren el algoritmo.

En este apartado se realizará un análisis de este tipo de algoritmos mediante diversas pruebas. El objetivo de las mismas es comprobar la situación de dicho algoritmo con respecto a otros algoritmos del estado del arte. Los algoritmos con los que se comparará serán los presentados a la competición VOC 2007, por lo que la metodología de las pruebas realizadas deberá ser similar a la metodología utilizada en dicha competición.

En líneas generales, el algoritmo de *Bag of Visual Words* consta de 2 bloques principales: descripción de las imágenes en base a diccionarios (o *wordbooks*), y clasificación de las imágenes.

En el primero de los bloques, el algoritmo [CSUR04] utiliza un elemento denominado *wordbook* para la descripción de una imagen, mientras que esta descripción es clasificada mediante el segundo de los bloques. Estos dos bloques son los que se han testeado, y cuyos resultados se analizan en este apartado.

De cara a las pruebas no se ha utilizado la implementación original de 2004, sino que en esta tesis hemos incluido dos modificaciones que son el estado del arte actual, que es el uso de descriptores visuales SURF, así como el uso de máquinas de vector soporte (SVM).

La metodología de pruebas está compuesta por los siguientes pasos:

1. Selección de bloques funcionales, y entrenamiento previo del algoritmo de detección mediante la base de dato de “entrenamiento”.
2. Optimización del algoritmo de detección mediante la base de datos de “validación”.
3. Obtención de resultados finales del algoritmo ya optimizado mediante la base de datos de “test”. Los resultados que se necesitan para realizar la comparación son la curva *precision-recall* y su área debajo de dicha curva, denominado *Average Precision* (AP).

Para la realización de los pasos 1 y 2 se ha utilizado una metodología sistemática para las 20 clases de objetos que conforman la competición VOC (ver Tabla III-1), de tal forma que se ha optimizado un detector individual para cada clase de objeto. Esta metodología es la siguiente:

1. El primer paso ha sido la selección del tamaño del vocabulario. Para realizar esta selección se ha llevado a cabo una búsqueda incremental de dicho tamaño, que

obtenga el mejor resultado cuando se aplica sobre la base de datos de “validación” y se utiliza un clasificador SVM con kernel lineal.

2. Una vez ha tenido el tamaño de vocabulario óptimo, es necesario hacer una optimización de los clasificadores. Para ello se han seleccionado tres clasificadores posibles, que son el clasificador Naive Bayes, un SVM con kernel lineal y un SVM con kernel gaussiano. En esta parte también se han optimizado los parámetros de los mismos con la base de datos de “validación”.
3. Por último, se han obtenido las gráficas de *precision-recall* sobre la base de datos de “test” para cada uno de los tres clasificadores utilizados así como su AP.

Con esta metodología se ha realizado el análisis del algoritmo y de las clases en los siguientes sub-apartados.

III.4.1.1 Resultados de los modelos discriminativos

En este sub-apartado se detallan las pruebas realizadas y los resultados obtenidos. Ya que éstas pruebas son similares para todos los objetos y debido a que el análisis se extendería demasiado, se ilustrarán los pasos con los resultados intermedios de dos de las clases más relevantes, que son la clase “*train*” y la clase “*potted_plant*”, por ser las que mejor y peor resultado obtuvieron en VOC2007 respectivamente..

El primero de los pasos definidos en la metodología es la selección del vocabulario óptimo. Para ello, se ha ejecutado el algoritmo de clusterización *k-means* con diferente número de centroides para generar diccionarios de diferentes tamaños. Con cada uno de los diccionarios se ha ejecutado la clasificación mediante un clasificador base compuesto por una Máquina de Vectores Soporte (SVM) con kernel gaussiano. Este clasificador ha sido entrenado con la metodología *one-vs-rest* para diferenciar entre la clase concreta deseada (C1) y otra clase genérica compuesta por el resto de objetos (C2).

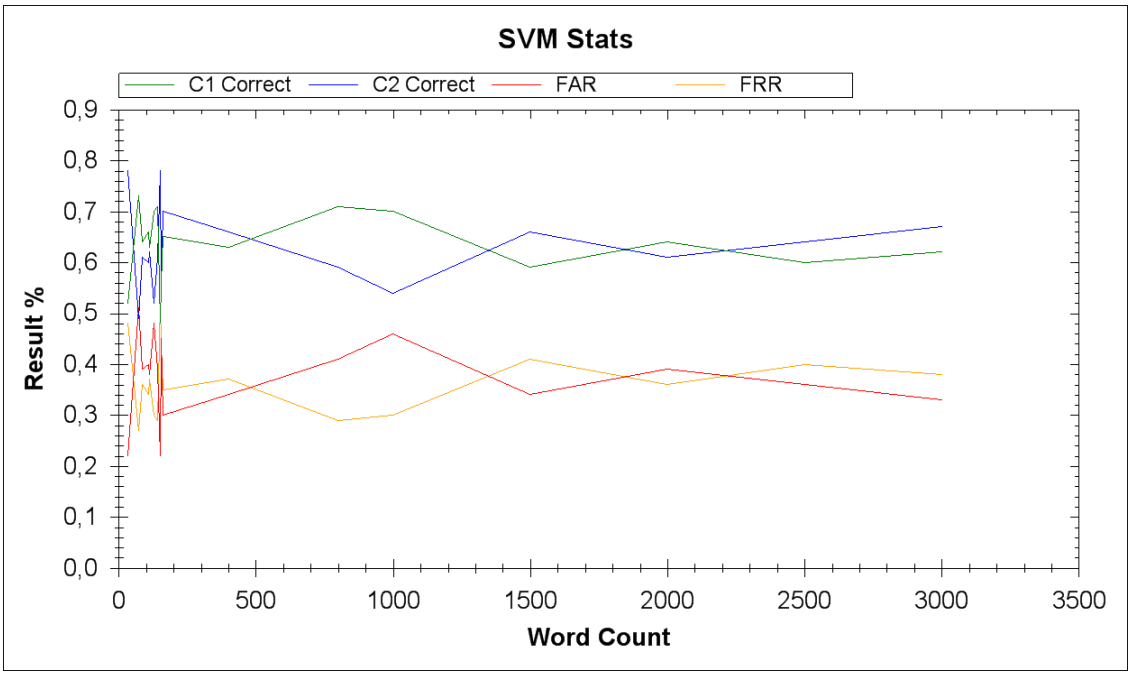


Figura III-5: Resultados de vocabularios con diferentes tamaños para la clase potted_plant

En la Figura III-5, se puede ver el resultado obtenido para la clase *potted_plant*. En el eje horizontal se muestra el número de palabras que componen el diccionario, y en el eje vertical se muestra el resultado en tanto por uno. Las métricas de evaluación a considerar han sido el número de aciertos de la clase concreta (C1) y de la clase "otros" (C2), además de la tasa de falsos rechazos (FRR) y falsas aceptaciones (FAR) (ver apartado III.3).

El resultado óptimo se selecciona utilizando el criterio de que la media de acierto (media de *C1 correct* y *C2 correct*) sea la mayor, teniendo en cuenta que cuanto más tamaño más tiempo tardará la anotación, pero mejor resultado tendrá. En este caso, el tamaño óptimo es cuando el vocabulario tiene 2.500 palabras.

El segundo de los pasos definidos en la metodología es la definición de los parámetros de configuración óptimos para los tres algoritmos de clasificación de referencia sobre el vocabulario óptimo. Los clasificadores elegidos han sido *Naive Bayes*, *SVM gaussiano* y *SVM lineal*. Como es habitual, en el caso de los modelos basados en SVM existen ciertas variables que es necesario optimizar, y son *Complexity*, *Epsilon* y *Tolerance*. Para evaluarlos, se compara el resultado en el set de validación y también sobre el de

entrenamiento. Al igual que antes se decide seleccionar los parámetros que obtienen un error medio de validación mejor.

En la Tabla III-4 se muestran algunos de los resultados de la optimización de los parámetros del modelo SVM Lineal para una de las 20 clases.

Tabla III-4: Resultados de optimización de clasificador SVM lineal sobre la clase "*train*"

Error base datos "entrenamiento"	Complexity	Epsilon	Tolerance	Resultado "validación"		
				C1 (%)	C2 (%)	Media (%)
0	1	0,0001	0,0001	67	77	72
0,035	**0,00097656**	**0,0001**	**0,0001**	**64**	**83**	**73,5**
0,0078	0,00195312	0,0001	0,001	65	80	72,5

Una vez se han elegido los parámetros óptimos del SVM lineal, se ha procedido a aplicar los 3 clasificadores sobre la base de datos de "test" del VOC2007, obteniendo como resultados curvas ROC y precision-recall como las mostradas en la Figura III-6.

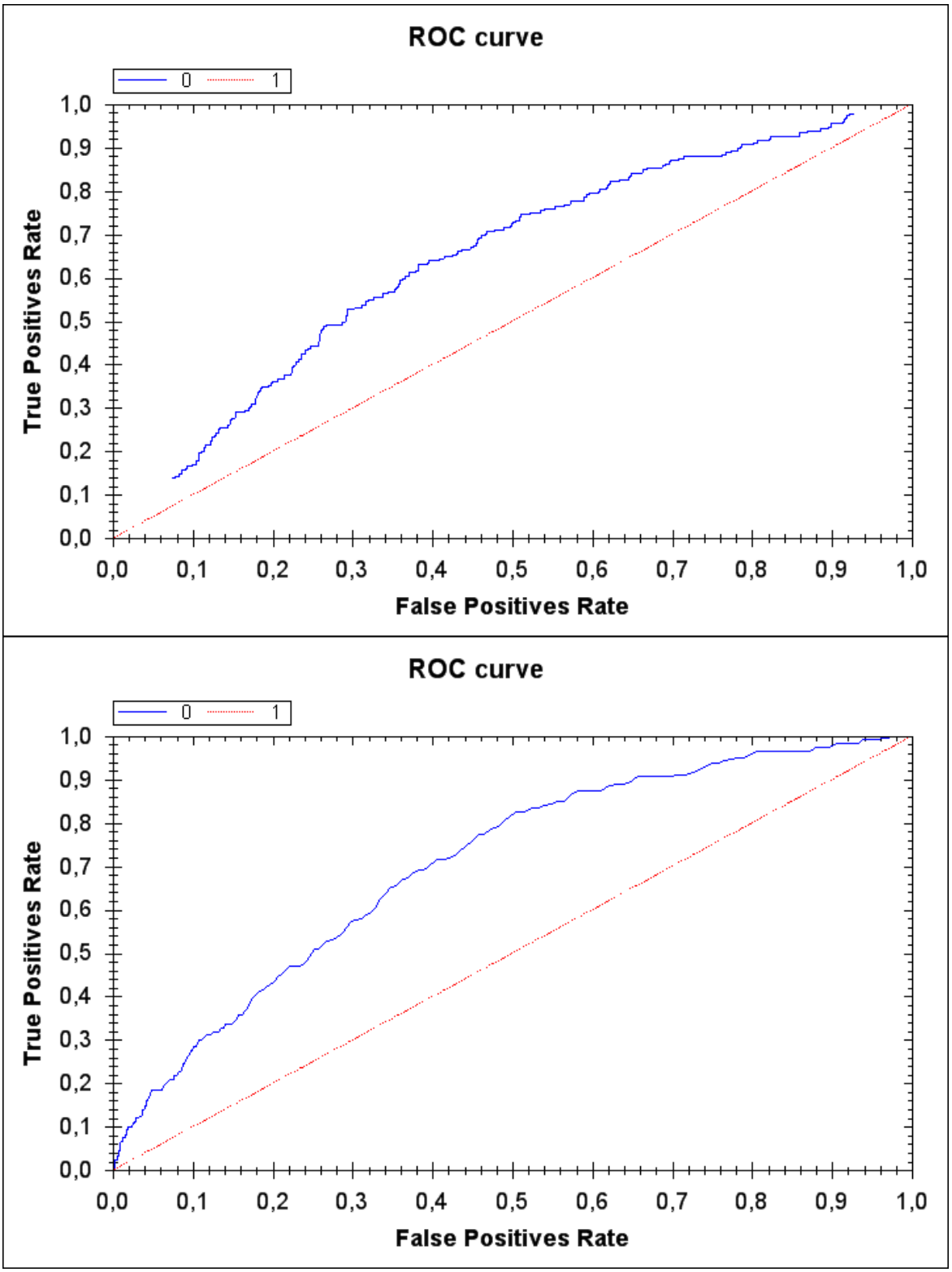


Figura III-6: Curvas ROC de los clasificadores Naive Bayes (arriba) y SVM lineal (abajo) sobre la clase potted_plant

En las curvas ROC presentadas en la Figura III-6, el mejor resultado es el que genera una curva lo más cercana posible a la esquina superior izquierda. Esto se debe a que así el ratio de positivos verdaderos (eje vertical) será siempre lo más cercano a 1.

El segundo de los resultados obtenidos ha sido la gráfica *precision-recall* de cada clasificador, junto con su valor de AP.

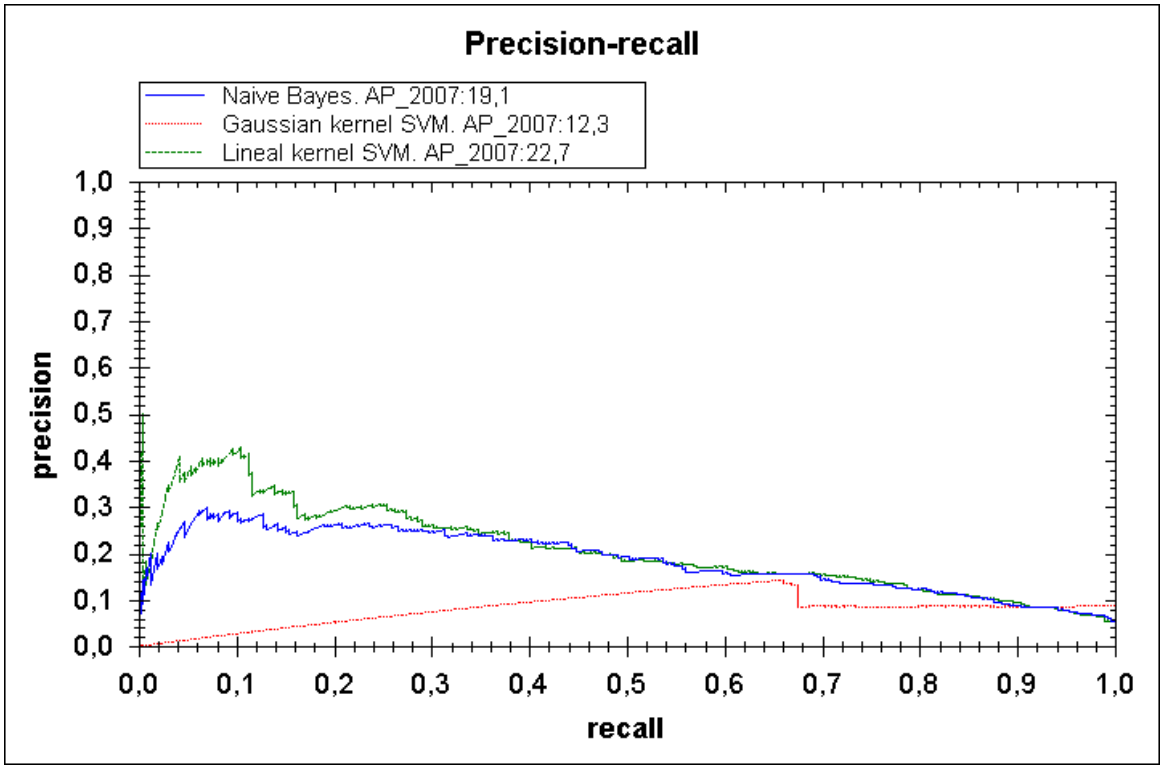


Figura III-7: Gráfica de precision-recall para los tres clasificadores para el objeto "train"

Al igual que en la Figura III-6, lo ideal es tener un área bajo la curva (AP) lo mayor posible, y que esta curva esté cercana a la parte superior derecha. En la Figura III-7 se comprueba que para el caso del objeto "*train*" el mayor área bajo la curva se da para el clasificador *SVM lineal*.

Los resultados se pueden comparar con los del resto de algoritmos de la competición gracias a la publicación de las gráficas de *precision*-recall *para* los algoritmos de la competición VOC del año 2007. Un ejemplo de las curvas de otros participantes en ese año se presentan en la Figura III-8.

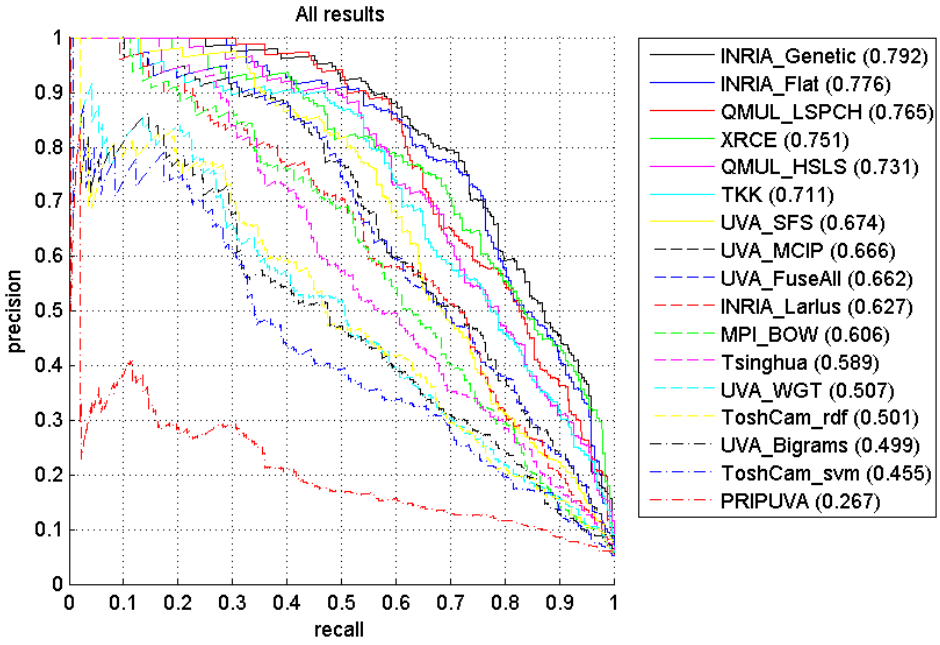


Figura III-8: Gráfica de precision-recall para los detectores del VOC2007 sobre el objeto "train"

En este caso, se ve claramente que el resultado del SVM lineal entrenado (con un AP=0,227) es mejor que el algoritmo peor de la competición (con un AP=0,172).

Esta comparativa numérica es el resultado final de este análisis. Ejecutando este tipo de análisis sobre los 20 objetos de VOC2007, los resultados son los que se muestran en la Tabla III-5.

Tabla III-5: Resultados comparativos de la competición VOC2007 y los tres clasificadores propuestos

Algoritmo	*AP aeroplane*	*AP bicycle*	*AP bird*	*AP boat*	*AP bottle*	*AP bus*	*AP cat*	*AP chair*	*AP cow*	*AP diningtable*
Naive Bayes BSC	0,130	0,231	0,134	0,194	0,096	0,146	0,184	0,247	0,077	0,117
Gaussian SVM BSC	0,104	0,101	0,106	0,077	0,071	0,069	0,118	0,166	0,057	0,078
Linear SVM BSC	0,361	0,203	0,167	0,188	0,104	0,106	0,177	0,251	0,085	0,148
INRIA_Larlus	0.626	0.540	0.328	0.475	0.178	0.464	0.442	0.446	0.260	0.381
INRIA_Flat	0.748	0.625	0.512	0.694	0.292	0.604	0.576	0.531	0.411	0.540
INRIA_Genetic	**0.775**	**0.636**	**0.561**	**0.719**	**0.331**	**0.606**	**0.588**	**0.535**	**0.426**	**0.549**
MPI_BOW	0.589	0.460	0.313	0.590	0.169	0.405	0.402	0.443	0.283	0.319
PRIPUVA	**0.486**	**0.209**	**0.213**	**0.172**	**0.064**	**0.142**	0.314	**0.274**	**0.123**	**0.143**
QMUL_HSLS	0.706	0.548	0.357	0.645	0.278	0.511	0.540	0.466	0.366	0.344
QMUL_LSPCH	0.716	0.550	0.411	0.655	0.272	0.511	0.551	0.474	0.359	0.374

Algoritmo	*AP aeroplane*	*AP bicycle*	*AP bird*	*AP boat*	*AP bottle*	*AP bus*	*AP cat*	*AP chair*	*AP cow*	*AP diningtable*
TKK	0.714	0.517	0.485	0.634	0.273	0.499	0.512	0.517	0.323	0.463
ToshCam_rdf	0.599	0.368	0.299	0.400	0.236	0.333	0.330	0.410	0.178	0.332
ToshCam_svm	0.540	0.271	0.303	0.356	0.170	0.223	0.346	0.380	0.190	0.275
Tsinghua	0.629	0.424	0.339	0.497	0.237	0.407	0.352	0.427	0.210	0.389
UVA_Bigrams	0.612	0.332	0.294	0.450	0.165	0.376	**0.313**	0.399	0.172	0.314
UVA_FuseAll	0.671	0.481	0.433	0.581	0.199	0.463	0.419	0.484	0.278	0.419
UVA_MCIP	0.665	0.479	0.410	0.580	0.168	0.440	0.405	0.485	0.278	0.417
UVA_SFS	0.663	0.497	0.435	0.607	0.188	0.449	0.419	0.468	0.249	0.423
UVA_WGT	0.597	0.337	0.349	0.445	0.222	0.329	0.363	0.368	0.206	0.252
XRCE	0.723	0.575	0.532	0.689	0.285	0.575	0.503	0.522	0.390	0.468

Algoritmo	AP dog	AP horse	AP motorbike	AP pottedplant	AP sheep	AP sofa	AP train	AP tvmonitor
Naive Bayes BSC	0,268	0,180	0,115	0,085	0,062	0,11	0,191	0,183
Gaussian SVM BSC	0,152	0,119	0,103	0,011	0,043	0,094	0,123	0,092
Linear SVM BSC	0,298	0,174	0,146	0,110	0,077	0,125	0,227	0,182
INRIA_Larlus	0.340	0.660	0.551	0.131	0.291	0.367	0.627	0.433
INRIA_Flat	0.428	0.765	0.623	0.353	0.413	0.501	0.776	0.493
INRIA_Genetic	**0.458**	**0.775**	**0.640**	**0.363**	**0.447**	0.506	**0.792**	**0.532**
MPI_BOW	0.344	0.636	0.535	0.223	0.266	0.354	0.606	0.406
PRIPUVA	**0.237**	**0.301**	**0.133**	**0.100**	**0.124**	**0.133**	**0.267**	**0.262**
QMUL_HSLS	0.399	0.715	0.554	0.158	0.358	0.415	0.731	0.455
QMUL_LSPCH	0.415	0.715	0.579	0.156	0.333	0.419	0.765	0.459
TKK	0.415	0.726	0.602	0.317	0.301	0.392	0.711	0.410
ToshCam_rdf	0.337	0.639	0.531	0.290	0.273	0.312	0.501	0.376

Algoritmo	AP dog	AP horse	AP motorbike	AP pottedplant	AP sheep	AP sofa	AP train	AP tvmonitor
ToshCam_svm	0.324	0.480	0.407	0.234	0.218	0.280	0.455	0.318
Tsinghua	0.347	0.650	0.481	0.169	0.308	0.328	0.589	0.331
UVA_Bigrams	0.306	0.616	0.424	0.145	0.209	0.235	0.499	0.300
UVA_FuseAll	0.385	0.698	0.514	0.325	0.319	0.360	0.662	0.403
UVA_MCIP	0.371	0.664	0.501	0.312	0.323	0.319	0.666	0.403
UVA_SFS	0.339	0.715	0.534	0.297	0.312	0.318	0.674	0.435
UVA_WGT	0.347	0.651	0.401	0.264	0.269	0.251	0.507	0.297
XRCE	0.453	0.757	0.585	0.326	0.397	**0.509**	0.751	0.495

En la Tabla III-5 se ven en rojo el peor resultado del objeto y en verde el mejor resultado de toda la columna. Como nota, hay que decir que los objetos "*person"* y "*car*" no se han podido testear por fallos en la base de datos.

III.4.1.2 Análisis de resultados de los modelos discriminativos y conclusiones

En la Tabla III-5 se puede ver un gran volumen de números, pero es necesario resumirlos para dar unas conclusiones claras y útiles. Para ello, hemos comparado el mejor resultado, el peor, contra el del mejor de los clasificadores que hemos implementado en el sub-apartado III.4.1.1 para cada objeto. Este resumen se muestra todo en la Figura III-9.

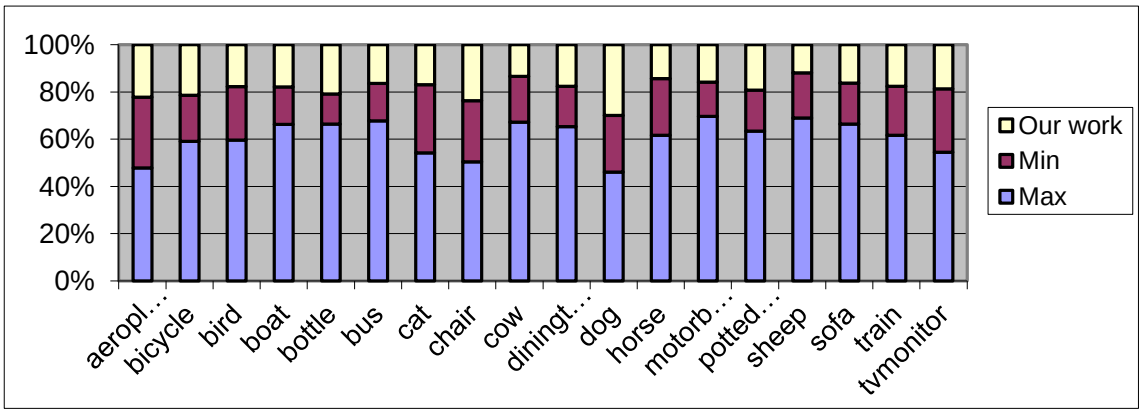


Figura III-9: Resultados comparativos de VOC2007 y la implementación de BoW

Cabe destacar que en la figura mostrada los valores no son absolutos, sino que se han relativizado para poder comparar en un gráfico único todos los resultados.

En este apartado III.4.1 hemos propuesto una modificación del algoritmo publicado en 2004 [CSURK04] que añade mejoras (descriptores SURF y SVM) para aumentar la velocidad de procesado. Este algoritmo es la base de todas las propuestas de VOC 2007, por lo que la teoría dice que el resultado tiene que ser mucho peor, pero nada más lejos de la realidad. A pesar de que muchos algoritmos sí generan mejores resultados, en otros casos nuestra propuesta de 2004 con las mejoras, sigue obteniendo mejores valores que otros algoritmos. Esto muestra que, evidentemente, el estado del arte ha avanzado, pero el algoritmo propuesto en 2004 y utilizado en esta implementación sigue siendo válido.

En cuanto a los tiempos de procesamiento, se ha comprobado que el algoritmo con las mejoras que hemos propuesto tiene unos tiempos bajos. Aproximadamente 1 segundo por imagen en una CPU Intel Core 2 Duo de 2,5 GHz donde otros, como [CSUR04], tardan 5 segundos debido a las características de los propios algoritmos.

A pesar de ello, se ha descubierto un problema que atañe tanto al algoritmo que hemos implementado como a todos los sistemas de detección propuestos dentro de la competición VOC Challenge en cualquiera de sus ediciones. Estos algoritmos pretenden la detección de un único objeto, y se especializan en su detección. Por ello, para anotar una imagen habrá que crear tantos detectores como objetos existan. Se ha visto de forma práctica que generar estos modelos requiere de un proceso bastante manual. Esto hace que, para la anotación a gran escala, como se plantea en esta tesis, con cientos de objetos involucrados será un trabajo excesivamente extenso en tiempo.

Además, nos encontraremos el problema de la precisión. Al ejecutar todos los clasificadores de cientos de objetos, todos ellos tendrán una tasa de error. Estas tasas de error conjuntas podrán generar errores en la anotación final, más frecuentes cuantas más imágenes se tengan. Adicionalmente existe el problema del entrenamiento con grandes cantidades de imágenes.

Por ambos motivos, coste computacional y error agregado, no creemos conveniente seguir con esta línea de investigación, usando métodos discriminativos, para la anotación de imágenes multi-objeto. Esta conclusión se asemeja a la presentada en otros trabajos como [DENG10] o [TSAI11].

III.4.2 Análisis de modelos basados en vecinos más cercanos

La segunda tendencia existente en el campo de la anotación de imágenes es la de los modelos de vecinos más cercanos. Esta técnica posee varios pasos básicos, que son los siguientes:

1) Descripción de imágenes
2) Computación de distancias entre imágenes
3) Búsqueda de imágenes similares
4) Propagación de etiquetas a la imagen de test

Cada uno de estos pasos no ha sido analizado en el estado del arte ni mejorado, sino que trabajos como [FU12] [GUILL11] han buscado generar de nuevo todos los pasos a la vez, para que el algoritmo completo sea mejor que otros del estado del arte. A continuación, se pretende analizar cada uno de estos pasos con tecnologías del estado del arte de cada uno de sus respectivos campos para comprobar si es posible mejorar esos pasos de

forma individual. Para ello, se utilizará la base de datos SAIAPR-TC12 ya que es una de las más utilizadas en este tipo de técnicas [MAKA10] [GUILL09].

III.4.2.1 Análisis de tecnologías de propagación de etiquetas

El primero de los bloques a analizar es el último de los pasos del sistema de anotación: el algoritmo de propagación de etiquetas.

La entrada a la fase de transferencia de etiquetas es un grupo de imágenes junto con sus etiquetas y un valor que dé una idea de la similitud ante la imagen de entrada, mientras que la salida debe ser una lista limitada de etiquetas.

Para calcular esta lista de salida existen muchas posibles aproximaciones directas y en este apartado se van a proponer analizar numerosas variantes inexistentes en el estado del arte, pero que son referencia en el campo del análisis lingüístico. Las implementaciones realizadas y probadas son las siguientes:

- **Greedy Transfer**: La transferencia propuesta en [MAKA10] donde se juega con la información de repetición de las etiquetas y coocurrencias.
- **MaxRep Transfer**: La etiqueta transferida es la más repetida de las presentes.
- **SemiGreedyTransfer**: Sólo se propagan algunas de las etiquetas de la imagen más cercana en función del algoritmo propuesto por [MAKA10].
- **Near Transfer**: Las etiquetas transferidas son las que posee la imagen más similar de la base de datos.
- **Frequency Transfer**: La transferencia sólo propaga las etiquetas con una mayor frecuencia de aparición en la base de datos.
- **Coocurrence Transfer**: Sólo se propagan las etiquetas que aparecen más comúnmente junto con otras en la base de datos.
- **ParallelResistor Transfer**: Se computa un valor para cada etiqueta y se transfieren las que tienen menor valor. Este valor se computa con todas las etiquetas del mismo tipo presentes en el conjunto, y se calcula como valores de resistencias electrónicas en paralelo.
- **ArithmeticMean Transfer**: Para cada etiqueta se computa un valor y se transfieren las que tienen mayor valor. Este valor se calcula en base a la similitud de todas las imágenes que contienen esa etiqueta, y ese valor es la media de todas las similitudes.

- **HarmonicMean Transfer**: Similar al caso anterior pero usando una media armónica.
- **MahalanobisVisual Transfer**: Se computa la distancia de Mahalanobis de cada etiqueta en base a las propiedades visuales de la imagen a la que corresponde.

En cuanto a la metodología de las pruebas, para todos los algoritmos se fijará un número de etiquetas de salida igual a 5, ya que sigue otros trabajos del estado del arte. Además, para hacer las pruebas se ha tenido que fijar un algoritmo común. Como se ha dicho con anterioridad, en este sub-apartado se analiza la fase de propagación de etiquetas, por lo que se debe fijar la primera fase del algoritmo, que es la computación de las imágenes similares. Para ello, se utilizará una propuesta del estado del arte [MAKA10] (apartado II.4.3). En ella se fijará un número de imágenes similares a devolver igual a K=10. Además de las distancias propuestas en dicho artículo, también se ha computado el mismo caso para la distancia *ChiSquare*, ya que suele tener mejores resultados cuando se trata de descriptores visuales basados en histogramas. Los resultados se pueden ver en la

Tabla III-6, que contiene los resultados de *precision*, *recall* y *F-measure* para la configuración de [MAKA10], y la

Tabla III-7, que contiene los resultados para *ChiSquare*. En esta etapa el *recall* es el valor a observar, ya que cuanto mayor sea, indica que en el conjunto de etiquetas de salida hay más etiquetas buenas. En cambio, la *precision* no es relevante, ya que sólo indica si hay muchas o pocas etiquetas en esta fase, característica que se podrá corregir mediante la configuración del número de etiquetas de salida.

Tabla III-6: Resultados de diferentes algoritmos base de propagación de etiquetas

TagTransfer	Precision	Recall	F-measure
GreedyTransfer	0,28	0,21	0,24
MaxRepTransfer	0,22	0,05	0,08
SemiGreedyTransfer	0,16	0,07	0,09
NearTransfer	0,28	**0,29**	**0,28**
FrequencyTransfer	0,23	0,15	0,18
CoocurrenceTransfer	0,09	0,04	0,05
ParallelResistorTransfer	**0,29**	0,17	0,21
HarmonicMeanTransfer	0,20	0,21	0,20
ArithmeticMeanTransfer	0,21	0,22	0,21
MahalanobisVisualTransfer	**0,29**	0,17	0,21

Tabla III-7: Resultados de los mejores algoritmos base de transferencia de etiquetas, usando una distancia ChiSquare para la similitud

TagTransfer	Precision	Recall	F-measure
GreedyTransfer	0,29	0,21	0,24
SemiGreedyTransfer	0,15	0,07	0,09
NearTransfer	0,28	**0,27**	**0,27**
FrequencyTransfer	0,24	0,14	0,178
CoocurrenceTransfer	0,07	0,04	0,05
ParallelResistorTransfer (label=5)	0,27	0,16	0,20
MahalanobisVisualTransfer (label=5,Sigma->minima)	**0,29**	0,17	0,21

Se comprueba cómo la transferencia de las etiquetas de la imagen más cercana, **NearTransfer**, es el algoritmo más equilibrado en términos de *F-measure*. A su vez, el *GreedyTransfer* también se postula como un buen algoritmo con uno de los mejores *recall.*

III.4.2.2 Análisis de descriptores visuales

Como se ha visto en el capítulo II, actualmente existe un gran número de descriptores que son más o menos aptos para cada problemática. En el caso de anotación de imágenes

mediante vecinos más cercanos un punto crítico es almacenar toda la información de las imágenes y sus descriptores, por lo que los descriptores no pueden tener un gran tamaño. En este apartado se han seleccionado aquellos descriptores que ocupan un menor espacio dando buenos resultados según el estado del arte, y son los siguientes:

- MPEG7-Scalable Color Descriptor (II.2.6.1)
- MPEG7-Edge Histogram Descriptor (II.2.6.2)
- GIST (II.2.11)
- CEDD (II.2.7)

A partir de ahí, se han probado los diferentes descriptores visuales utilizando distancias básicas L1 y los dos mejores algoritmos de propagación de etiquetas (*GreedyTransfer* y *NearTransfer*) analizados en el apartado III.4.2.1, obteniendo los resultados de la Tabla III-8.

Tabla III-8: Resultados obtenidos usando diferentes descriptores de imagen

		Descriptor visual				
		EHD [ISO02]	**SCD [ISO02]**	**GIST [OLIV01]**	**CEDD [CHAT08]**	**SCD+EHD**
GreedyTransfer	*Precision*	0,15	0,27	0,19	0,22	**0,31**
	Recall	0,10	0,18	0,12	0,17	**0,22**
	F-measure	0,12	0,22	0,15	0,19	**0,26**
NearTransfer	*Precision*	0,15	0,27	0,18	0,21	**0,29**
	Recall	0,14	0,27	0,17	0,21	**0,29**
	F-measure	0,14	0,27	0,17	0,21	**0,29**

Se comprueba cómo la combinación de descriptores de MPEG7 que proponemos son los que obtienen mejores valores de *precision* y *recall*. Además, ocupan un 98% menos de memoria que la propuesta de [MAKA10], y su velocidad de computación es mucho mayor (II.2.6).

III.4.2.3 Análisis de tecnologías de distancia entre imágenes

Otro aspecto a tener en cuenta para mejorar sobre el algoritmo base de anotación mediante vecinos más cercanos es la distancia utilizada para computar la similitud entre dos imágenes. Esta distancia puede ser una única o una combinación de ellas como se ha visto en el apartado II.3.1. En este apartado se probarán aquellas distancias que son más relevantes para esta técnica según el estado del arte, y son:

- Distancia L1 (II.3.1.2)
- Distancia Coseno (II.3.1.4)
- Distancia L2 (II.3.1.1)
- Distancia Chi Square (II.3.1.5)
- Combinaciones de distancias mediante JEC (II.3.1.9)

Estas distancias se aplicarán sobre los mejores descriptores encontrados en el apartado III.4.2.2, y son los dos descriptores de MPEG7, y para realizar la evaluación final se utilizarán los mismos algoritmos de transferencia de etiquetas que los usados anteriormente.

En la Tabla III-9 se muestran los resultados de estas combinaciones.

Tabla III-9: Resultados obtenidos utilizando diferentes distancias para la computación de los vecinos más cercanos y los mejores algoritmos de propagación de etiquetas de los propuestos. Los resultados están medidos como precision (p), recall (r) y F-measure (F1)

		Distancia						
		L1,L1	**Coseno**	**JEC (L1,L1)**	**JEC (L2,L2)**	**JEC (CS[1],CS)**	**JEC (L1,L2)**	**JEC (L2,L1)**
GreedyT.	*p*	**0,31**	0,28	**0,31**	0,29	0,29	0,27	0,15
	r	**0,22**	0,20	0,21	0,21	0,21	0,20	0,11
	F1	**0,26**	0,23	0,25	0,24	0,24	0,23	0,13
NearT.	*p*	**0,29**	0,28	**0,29**	0,28	0,28	0,27	0,15
	r	**0,29**	0,27	**0,29**	0,28	0,28	0,27	0,14
	F1	**0,29**	0,27	**0,29**	0,28	0,28	0,27	0,14

El mejor resultado se obtiene usando distancias L1. En cuanto a la forma de cálculo, el resultado es igual ya sea utilizando la distancia L1 para el vector combinación de MPEG7-SCD y MPEG7-EH, que usando la distancia JEC. Finalmente, se cree que proseguir con JEC es la mejor opción, ya que la combinación y ponderación de características resulta más interesante desde el punto teórico, y desde el punto práctico será capaz de absorber diferentes variaciones de los descriptores.

Además de estas distancias básicas tal y como se mostró en el apartado II.3.2, existen algoritmos que modifican el espacio de computación y aprenden distancias adaptadas al problema concreto. Para esto se han implementado los algoritmos presentados en el capítulo II, y se han aplicado con el mejor de los descriptores visuales sobre la base de datos de referencia SAIAPR-TC12. Más exactamente, se han utilizado los siguientes algoritmos de aprendizaje de métricas para mejorar la computación de las distancias:

[1] CS=ChiSquare

- ITML (II.3.2.1)
- Metric Learning to Rank (II.3.2.2)
- KISSME (II.3.2.3)
- OASIS binary (II.3.2.4)
- OASIS gradual (II.3.2.4)

Todos estos algoritmos se basan en trabajar con parejas (o ternas) de imágenes. De estas parejas se eligen dos conjuntos, de tal forma que un grupo de parejas sean imágenes similares y otro grupo de parejas sean imágenes no similares. Como se intuye, esta información no está disponible en la base de datos, sino que en ella existen imágenes con etiquetas. Para determinar si dos imágenes son similares, hemos considerado que tengan en común un porcentaje de etiquetas. Para las pruebas, hemos definido este porcentaje como una variable con la que se han realizado diferentes pruebas con valores de 0,2 , 0,5 y 0,8, identificándolo en las figuras como *similarity threshold*.

Otro problema que se puede intuir es el número de parejas existente. Si se toman todas las parejas posibles de una base de datos compuesta por miles de imágenes, el número de parejas con las que trabajar es de muchos millones y no es factible. Por ello, hemos seleccionado de forma aleatoria un conjunto limitado de parejas, y este tamaño del conjunto también lo hemos establecido como un parámetro. Los valores utilizados para este parámetro han sido 100, 1.000 y 10.000, y se ha identificado como *#constrains* en las figuras. Finalmente, en todos los casos se está trabajando con un conjunto de imágenes similares que tiene el mismo tamaño que el de imágenes diferentes, ya que para valores diferentes alguno de los dos conjuntos se limita demasiado.

Para todos estos parámetros hemos medido el *recall*. En este caso, se utiliza esta métrica ya que se busca que las imágenes similares contengan exclusivamente las etiquetas deseadas, debido a que se está entrenando no sólo información visual sino también textual, en forma de parejas.

El *recall* medido para las variables anteriores, y mostrado como eje Z en las gráficas, junto con los dos parámetros anteriores en el eje X e Y, se muestra en la Figura III-10, Figura III-11 y Figura III-12.

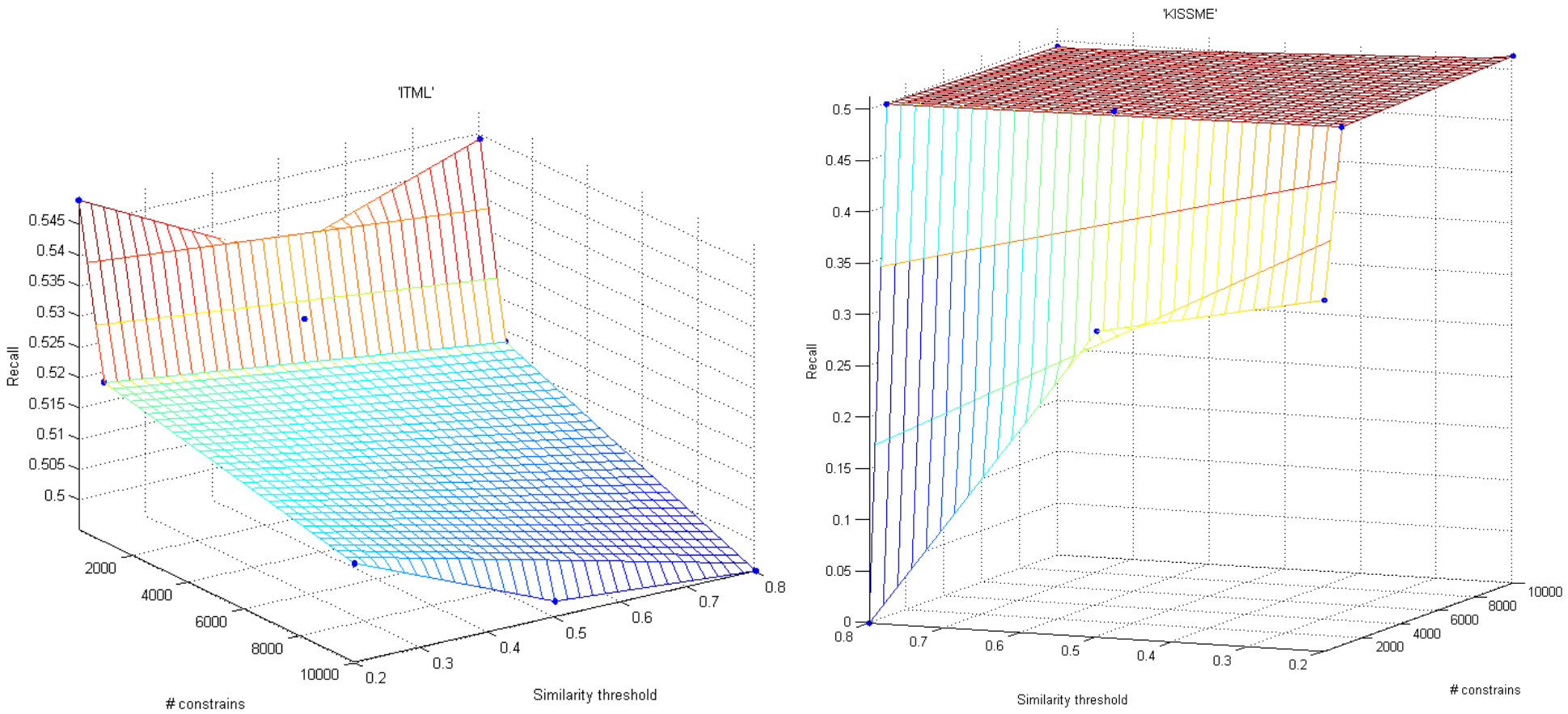


Figura III-10: Recall obtenido con los algoritmos ITML y KISSME para diferentes parámetros

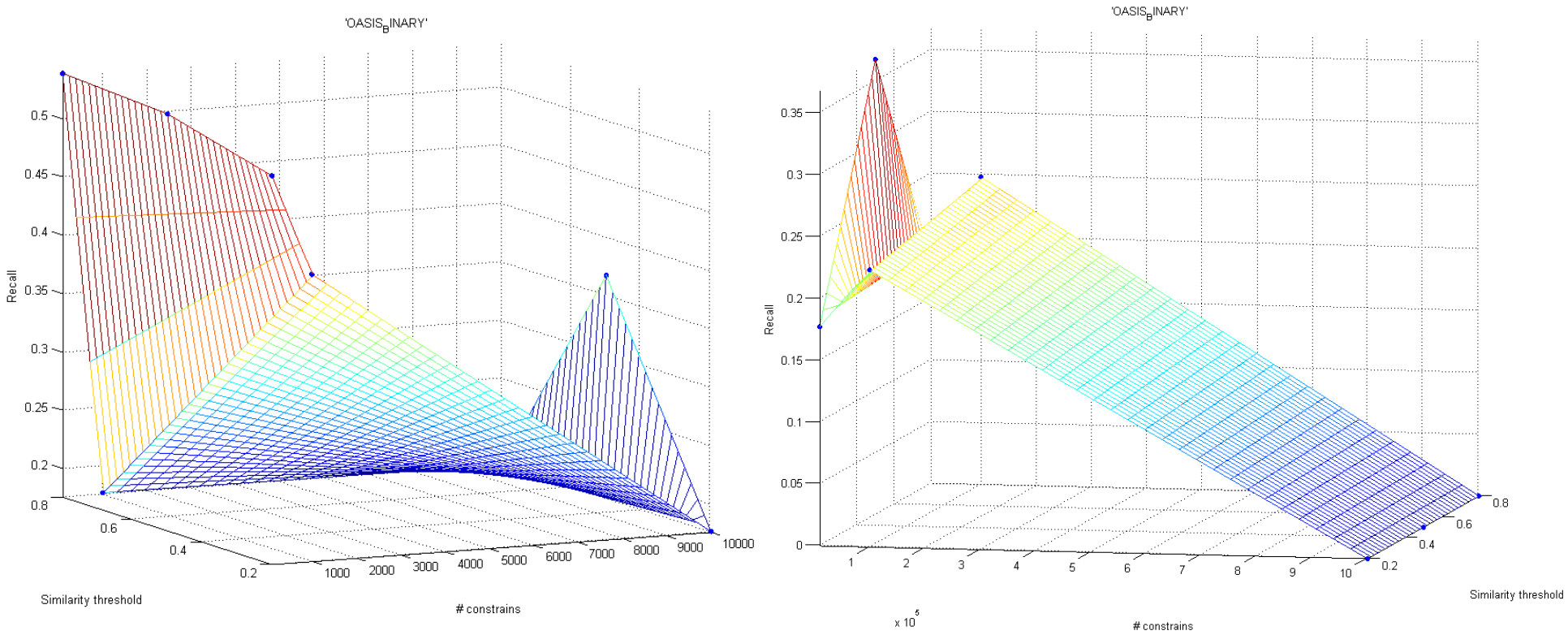


Figura III-11: Recall obtenido con los algoritmos OASIS BINARY para diferentes parámetros

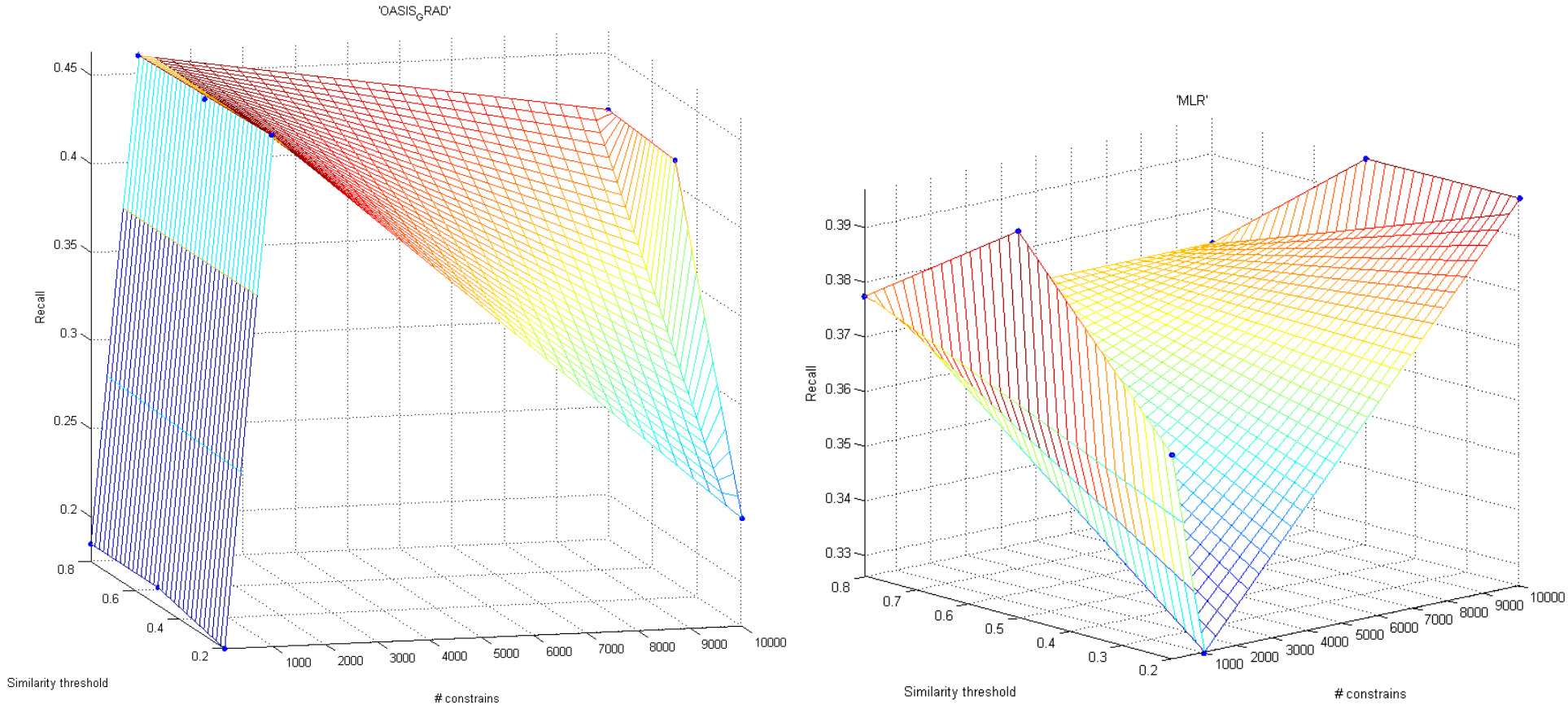


Figura III-12: Recall obtenido con los algoritmos OASIS GRADUAL y MLR para diferentes parámetros

En general, hemos comprobado que los algoritmos funcionan mejor con un menor número de parejas (un número de *constrains* menor que 1000), excepto KISSME donde el mejor resultado se obtiene con un número alto de parejas. En cuanto al uso de un porcentaje de similitud en el número de etiquetas no existe un consenso de uso entre los algoritmos. La idea existente detrás de este porcentaje es que un valor alto refleja que cuando dos imágenes se dice que son iguales es porque semánticamente son muy iguales; puesto que comparten muchas etiquetas. Así que si la similitud semántica es alta, entonces el resultado debe ser mejor, pero no es esto lo que se ve en las figuras. ITML y KISSME no funcionan bajo esta lógica y al aumentar la restricción de etiquetas comunes su funcionamiento es peor. El caso de OASIS_BIN es diferente totalmente y se asemeja a lo esperado. El otro algoritmo que debemos mencionar es el OASIS_GRAD, donde no se utiliza este porcentaje de etiquetas, sino un valor absoluto, y las varianzas en el *recall* sólo se deben al uso de un conjunto diferente de imágenes.

Para comparar numéricamente un escenario de anotación de imágenes, hemos seleccionado dos algoritmos de propagación de etiquetas y los hemos ejecutado tras la búsqueda por similitud aprendida. Los resultados se resumen en la Tabla III-10.

Tabla III-10: Resumen de los valores de Recall para los diferentes algoritmos de aprendizaje de métricas

	Recall		# constrains	Similarity Threshold
	Near Tag Transfer	All Tags Transfer		
ITML	0.2760	0.5488	100	0.2
KISSME	0.2476	0.5124	10000	0.2
OASIS_BIN	0.2762	0.5385	100	0.8
OSASIS_GRAD	0.2176	0.4641	1000	0.2
MLR	0.17	0.40	10000	0.2
NO LEARNING (L1L1)	**0.29**	**0.55**	-	-
NO LEARNING (L1)	**0.29**	**0.55**	-	-
NO LEARNING (L2)	**0.29**	**0.55**	-	-

De forma clara se comprueba que, de todos los algoritmos de aprendizaje de métricas, los dos mejores son OASIS en su versión binaria e ITML. A pesar de ello, ninguno ha conseguido acercarse al recall obtenido con otras métricas sin aprendizaje, por lo que la conclusión es que no se usará aprendizaje de métricas como *baseline* en esta tesis.

III.4.2.4 Análisis de tecnologías de búsqueda rápida de imágenes

A lo largo del capítulo II sobre el estado del arte, así como en este capítulo, se ha visto que la aproximación directa de la recuperación de imágenes similares no es algo factible cuando se trata con bases de datos de un gran volumen, ya que la computación es excesivamente costosa. Para poder hacer este tipo de recuperaciones, en este apartado proponemos la utilización de técnicas del estado del arte de Big Data que permitan un acceso eficiente a toda la información disponible de imágenes.

Para ello, realizamos la propuesta de la Figura III-13, sobre la cual analizaremos si realmente es mejor la utilización de tecnologías de Big Data para el acceso, o por el contrario introducen mucha sobrecarga.

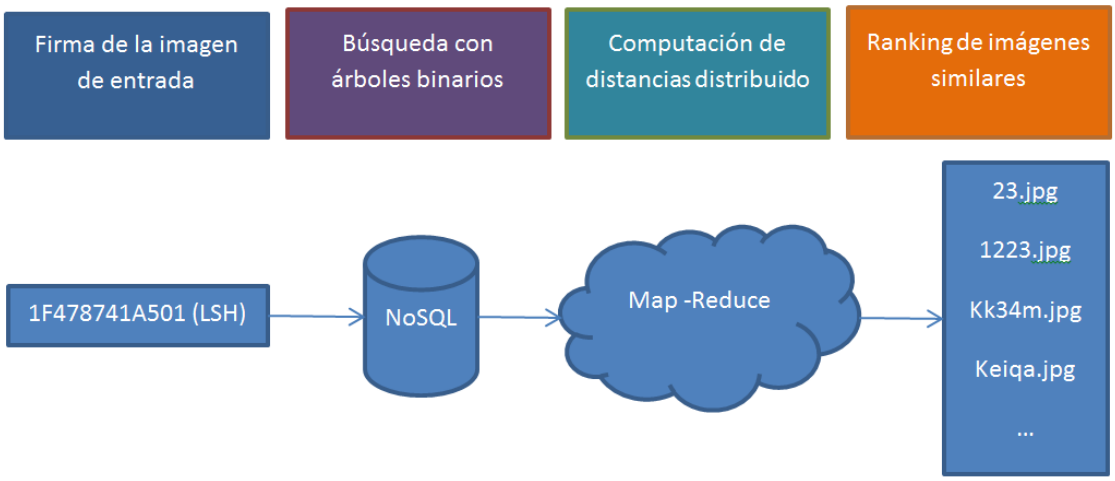


Figura III-13: Fases del algoritmo propuesto para la recuperación de imágenes similares a gran escala

El primero de los problemas al que es necesario enfrentarse es almacenar de forma eficiente las imágenes. Además de la propia imagen, es crucial almacenar los descriptores visuales. Puesto que ésta no es una información estática y es dependiente del análisis o búsqueda a realizar, será necesario disponer de un almacén de datos lo suficientemente flexible. Además, se deberá poder guardar una gran colección de datos, como son el origen de las imágenes (la página web, fecha de acceso, etc.), anotaciones manuales o comentarios. Para ello, se propone el uso de una base de datos distribuida NoSQL orientada a documentos, como es *MongoDB*, ya que permite disponer de un esquema flexible y es capaz de tener réplicas o nodos distribuidos. Esto también permite poder realizar computaciones en paralelo de forma eficiente usando técnicas del estado del arte.

Sobre este almacén de datos NoSQL es necesario construir el sistema de búsqueda de imágenes similares. Como se ha mencionado con anterioridad no es factible computar para cada imagen de la colección la distancia a la imagen de consulta. Por ello, es fundamental utilizar algún sistema de búsqueda aproximada de las imágenes más cercanas.

En este apartado se ha seleccionado el algoritmo Locality-Sensitive Hashing (LSH) [SLAN08] como algoritmo base para esta recuperación. Este algoritmo permite generar una firma numérica (o *hash*) para cada descriptor o conjunto de descriptores de imagen, de tal forma que aquellos vectores que tengan una distancia Euclídea muy baja y, por tanto, sean vectores muy similares, posean la misma firma numérica. Este tipo de algoritmos es muy útil para encontrar entradas similares dentro de grandes colecciones de datos, por ejemplo, buscando páginas web similares [SLAN08], por lo que su aplicación a los descriptores visuales de imágenes es lógica.

Una vez se tienen las firmas para todas las imágenes de la colección, la búsqueda de las imágenes similares es sencilla: dada una imagen de consulta, se generará su *hash* o firma. Con esta firma se buscará en toda la base de datos las imágenes que posean la misma firma y todas ellas serán las imágenes más similares. Para hacerlo más eficiente se propone utilizar la aproximación *multiprobe* [SLAN12], que se centra en generar tres firmas diferentes, lo que logra una fiabilidad mayor al recuperar los elementos más similares.

Para realizar la comparación de firmas en la base de datos las bases de datos NoSQL disponen de técnicas de indexación y búsqueda rápida de un número único. Un ejemplo de éstas es el algoritmo de búsqueda en árbol binario de la base de datos NoSQL *MongoDB*. Por ello, la búsqueda ya no se circunscribe a calcular una distancia entre vectores, sino a usar una arquitectura de índices para encontrar un número concreto.

Tras este paso de comparación de firmas con LSH, ya se dispone de un conjunto de imágenes de la biblioteca que se puede decir que son similares a la de la entrada. En este punto es obligatorio calcular la distancia concreta, entre las imágenes similares, ya que así se pueden ordenar correctamente.

Para este último paso, también se va a aprovechar el almacén de datos NoSQL distribuido propuesto. Ya que las imágenes pueden estar almacenadas en localizaciones espaciales diferentes y cada cálculo de la distancia entre la imagen de consulta y la imagen similar es independiente, es posible usar el paradigma *Map-Reduce*.

El objetivo del modelo *Map-Reduce* es ejecutar un procesamiento determinado sobre un conjunto de datos, también llamado lote de datos, de forma distribuida en diferentes localizaciones [DEAN08]. Para ello, se utilizan las funciones *Map* y *Reduce*. En la primera

se distribuyen los datos por los nodos del clúster en forma de pares clave-valor, y en la segunda se recogen aquellos pares clave-valor que cumplan los criterios del resultado deseado.

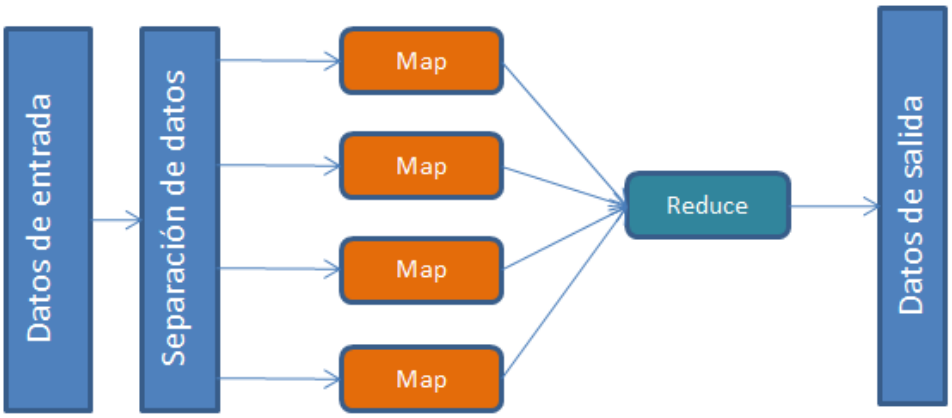


Figura III-14: Diagrama de funcionamiento del modelo Map-Reduce

En relación al cálculo de distancias, este modelo permitirá en la función *Map* el cálculo de la distancia entre cada imagen similar y la de consulta, ejecutándose en cada nodo de la red distribuida de almacenamiento. Por otro lado, el método *Reduce* se encargará de ordenar todas las distancias y podrá generar el ranking final de imágenes similares útiles para la analítica.

Hasta este momento, hemos descrito la propuesta que estamos realizando de algoritmo de búsqueda rápida de imágenes similares sobre grandes colecciones de imágenes. Para ello, hemos usado algoritmos del estado del arte y es necesario ver si nuestra propuesta es más eficaz y rápida que otras.

De esta forma, se ha comparado el tiempo de búsqueda de cuatro algoritmos contra el número de documentos, y se ha visualizado en una gráfica. En el eje horizontal, se muestran diferentes números de imágenes que se han introducido en la bases de datos de validación. En el eje vertical se visualiza el tiempo en segundos que se ha tardado de media en buscar una única imagen similar dentro de la base de datos de validación. Se han probado cuatro algoritmos diferentes que son la búsqueda de las imágenes similares cuando toda la base de datos está en memoria, el uso de la técnica LSH con la base de datos en memoria, así como la búsqueda exacta utilizando *Map-Reduce* y la propuesta

híbrida realizada con *Map-Reduce* y LSH. En la Figura III-15 se muestra la gráfica comparativa de los cuatro métodos.

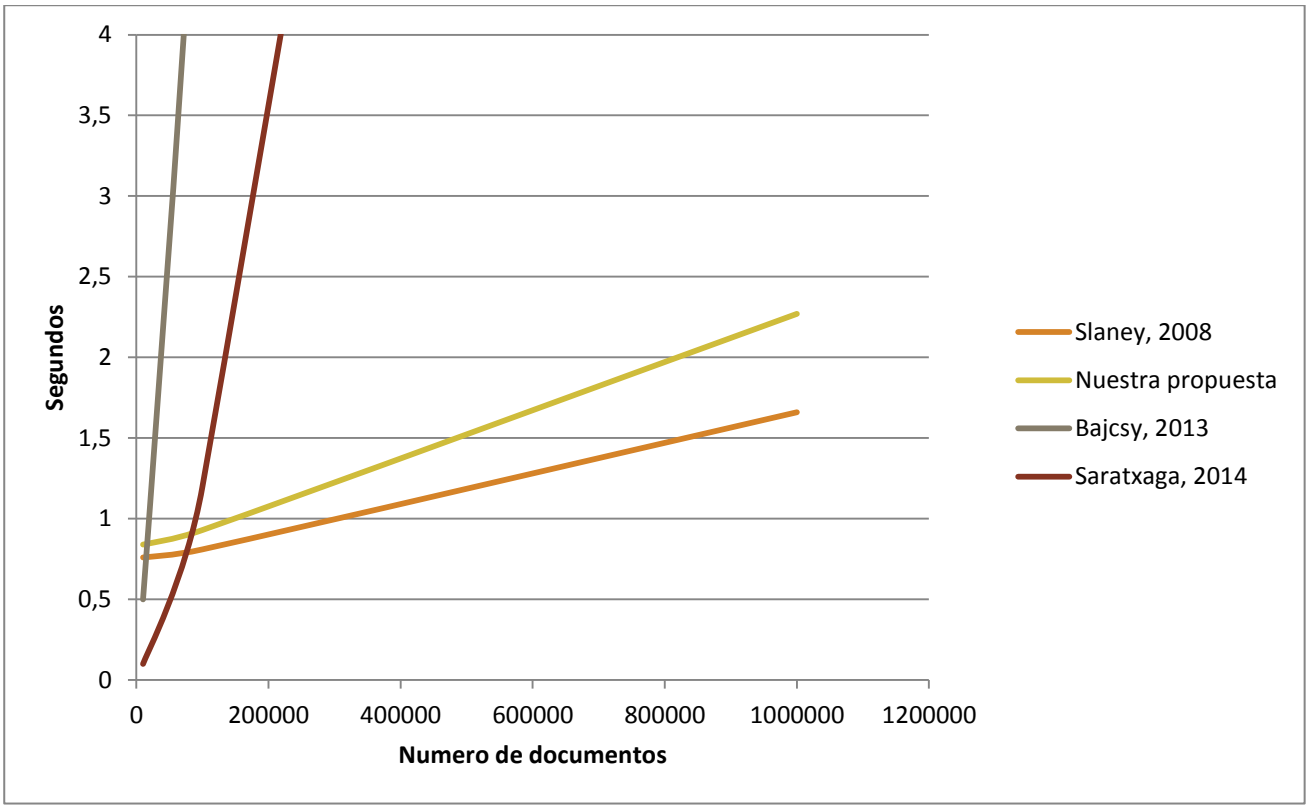


Figura III-15: Comparativa de tiempos medios de búsqueda de una imagen usando cuatro aproximaciones diferentes

El resultado muestra que el tiempo de búsqueda en las técnicas de búsqueda exacta aumenta de forma exponencial con el número de imágenes almacenado, mientras que en las búsquedas aproximadas usando LSH se trata de un aumento lineal. Como es de esperar, el acceso a memoria es mucho más rápido que el uso de árboles de indexación distribuidos. Por ello, la aproximación de LSH en memoria es la que menos tiempo tarda. Aun así, esto no es eficiente grandes bases de datos de imágenes con diferentes localizaciones distribuidas, ya que la multitud de nodos hace que no sea viable mantener todo en memoria.

Como conclusión, se ve que la combinación de tres hash diferentes para cada descriptor, y una búsqueda en estas tres tablas mediante el algoritmo *binary tree*, genera la búsqueda más rápida sin perder nada en el resultado final.

III.5 Conclusiones y definición del *baseline*

En este capítulo III, se han visto los resultados obtenidos por un buen número de algoritmos del estado del arte, y no todos ellos han generado los resultados argumentados por los propios autores. La razón es que cada trabajo se ha centrado en una o varias bases de datos de imágenes concretas o en una problemática muy específica, y no permiten generalizar y aplicar dichos algoritmos a otros grupos de imágenes.

En primer lugar, se ha perfilado el análisis de los algoritmos discriminativos, que son los más utilizados en la actualidad para detectar grupos reducidos de objetos. En este caso hemos demostrado cómo un modelo de *Bag-of-Words* de 2004 modificado ha obtenido resultados aceptables para las etiquetas propuestas en la base de datos VOC 2007. Estos resultados no son nada malos, e incluso se podrían mejorar usando algunas de las últimas técnicas propuestas en la competición VOC 2012. A pesar de ello, se plantea una gran incertidumbre sobre su tasa de no aciertos: fallando alrededor del 20% en cada etiqueta, cuando se aborden 100 o 1.000 etiquetas pueden obtenerse resultados inconsistentes. La anotación de imágenes basada en algoritmos discriminativos se funda en que para cada etiqueta se entrena un modelo y posteriormente se aplican todos los modelos a la imagen. Todos los resultados se ordenan en función de su probabilidad de aparición, y se selecciona un conjunto de etiquetas con mayor valor. Si existen mil etiquetas y cada una de ellas tiene un alto grado de fallo, el resultado final no será real, porque por probabilidad habrá ciertas etiquetas (por ejemplo 10 etiquetas, haciendo el 0,1% del total) que fallen. Al fallar y generar una probabilidad de aparición alta, éstas etiquetas serán las que se utilicen en la anotación, pero estará fallando la anotación. Este problema aumenta si el número de etiquetas aumenta. El principal motivo es que en esta aproximación para anotar una etiqueta no tiene en cuenta el resto, por lo que una posibilidad de mejora será desarrollar una segunda capa de homogeneización y decisión.

En vez de generar este segundo paso, hemos visto más útil utilizar un modelo de vecinos más cercanos, lo que está en línea con lo que proponen otros autores [FU12]. En este caso, el trabajo de Makadia et al. [MAKA10] es el punto de referencia, y el que se ha establecido como trabajo base en el estado del arte. A pesar de ello, hemos comprobado cómo la utilización de diferentes algoritmos básicos de otros campos del aprendizaje máquina ha obtenido un mejor resultado que el propuesto en ese trabajo.

En el apartado III.4.2.2 se han probado diferentes descriptores visuales, y se ha visto cómo el uso de los descriptores MPEG7-SCD y MPEG7-EHD proporciona una mejor tasa de acierto con un 98% menos de consumo de espacio en memoria. En el apartado III.4.2.3 se ha probado con diferentes distancias, logrando un mejor resultado con las distancias L1, mientras que las últimas técnicas del estado del arte de aprendizaje de métricas no han sido capaces de mejorar la anterior. En relación a esta búsqueda, se ha evidenciado que la mejor opción es no usar una búsqueda exhaustiva (III.4.2.4), sino que una combinación de una indexación *de árbol binario* con una técnica de *hashing* proporciona el mismo resultado de anotación que una búsqueda exhaustiva pero con un tiempo de respuesta sub-linear. Finalmente, en el apartado de propagación de etiquetas se han propuesto y probado diferentes algoritmos, y se ha visto que la propagación de las etiquetas de la imagen más cercana únicamente da unos resultados superiores a otros mostrados en el estado del arte.

Utilizando todas estas mejoras hemos definido un nuevo algoritmo base que resume el estado del arte, y algunas anotaciones sobre imágenes de la base de datos SAIAPR-TC12 de este algoritmo base son las mostradas en la Tabla III-11.

Tabla III-11: Resultados de la anotación automática de imágenes de SAIAPR-TC12 utilizando el baseline propuesto en esta tesis

Imagen	Etiquetas predichas
	grandstand, stadium, lawn, uniform, team, people, round
	man, court, tennis, stadium, player, woman, front, net, flag, green
	Sky, tree, people, building, house, mountain, cloud, (palm, street)
	Tree, forest, bush, tourist, woman, front, people

Se ve cómo la mayoría de las anotaciones son correctas, independientemente del tamaño que ocupa en la imagen. De estas anotaciones, algunas de las mejores etiquetas se presentan en la Tabla III-12 y algunas de las peores, en la Tabla III-13.

Tabla III-12: Etiquetas con mejor resultado de anotación

Concepto	Imágenes asignadas correctamente	Imágenes predichas	Imágenes en el Ground Truth	Precision	Recall
Tennis	14	15	17	0.93	0.82
Court	17	18	18	0.94	0.94
Racetrack	6	6	11	1.00	0.55
Sky	510	1334	535	0.38	0.95

Tabla III-13: Etiquetas con peor resultado de anotación

Concepto	Imágenes asignadas correctamente	Imágenes predichas	Imágenes en el Ground Truth	Precision	Recall
Hut	0	0	10	0	0
Bench	0	1	31	0	0
Shoe	0	0	11	0	0
Ravine	0	0	7	0	0

Como se puede ver, mediante el algoritmo propuesto los mejores resultados se obtienen para conceptos genéricos como *tennis*, o *racetrack*. En cambio, los resultados para objetos concretos, como *shoe* o *bench,* son mucho peores. Esto hace ver que la solución propuesta tiene mucho recorrido en la anotación de conceptos de alto nivel, y no tanto en la anotación de aquellos conceptos de más bajo nivel.

III.5.1 Resumen del *baseline* seleccionado

La conclusión obtenida en este capítulo es que el modelo que reproduce la línea base del estado del arte es el modelo de vecinos más cercanos con los siguientes componentes:

- **Descripción de imagen**: descriptores MPEG7-SCD y MPE7-EHD
- **Computación de distancias**: JEC con distancias L1 como base

- **Búsqueda de imágenes similares**: indexación mediante *árbol binario* de tres hashes (*multiprobe*) basado en LSH.
- **Propagación de etiquetas**: sólo se propagan las etiquetas más cercanas.

A partir de este algoritmo, a lo largo de esta tesis se trabajará en mejorar dos de los puntos en los que no se han visto grandes avances en este apartado, y son la descripción de la imagen (en los capítulos IV y V) y la propagación de etiquetas (en el capítulo VI).

IV. Modelo funcional de córtex visual primario

IV.1 Introducción

Tras el análisis de los descriptores visuales realizado en el apartado III.4.2.2 se ha mostrado que el estándar MPEG7 y específicamente sus descriptores de color (Scalable Color Descriptor – SCD) y textura (Edge Histogram Descriptor – EHD) son los que obtienen un resultado más óptimo que otros descriptores del estado del arte usando el *baseline* propuesto. De esta forma, se ha visto cómo estos descriptores ocupan un menor espacio en memoria y consiguen obtener igual o mayor *precision* y *recall* que otros de la literatura.

En el apartado II.2.6 del capítulo del estado del arte se ha mostrado el funcionamiento interno de ambos descriptores pertenecientes al estándar MPEG7. Más en concreto, se ha visto una diferencia fundamental que debe ser recalcada: mientras que el descriptor SCD utiliza el concepto de color perceptual, tal y como lo perciben los humanos, en base al espacio de color HSV, el descriptor EHD utiliza varios filtros específicos definidos manualmente. Dado que ambos descriptores del estándar MPEG7 se han mostrado como los mejores descriptores para la tarea de anotación de imágenes, y dado que la parte de color trabaja con un descriptor bioinspirado, este capítulo se centra en introducir conceptos bioinspirados en la parte de análisis de textura de MPEG7 con el objetivo de mejorar el funcionamiento del descriptor EHD.

Recordando la función del descriptor MPEG7-EHD, éste trata de representar los bordes existentes en una imagen. Dentro del cerebro de los animales, esta representación también se da, y concretamente es el córtex visual primario el que lleva a cabo esta tarea.

En este capítulo se trabajará en el análisis de esa capa del córtex visual y se propondrá un nuevo modelo computacional de dicha capa, de forma que genera una representación de los bordes de la escena al igual que hace el córtex de los macacos. Para ello, en el apartado IV.2 se mostrará el funcionamiento del sistema visual y en el IV.3 se realizará un repaso a los modelos matemáticos funcionales y computacionales del córtex V1 existentes en la literatura. Tras ello, en el apartado IV.4 se realizarán diferentes propuestas de nuevos modelos basados en conceptos de la literatura y, finalmente, en el apartado IV.5 se analizará cómo los modelos propuestos responden ante diferentes efectos, de forma que sea posible seleccionar el modelo que más fielmente represente las respuestas del sistema biológico.

Gracias a ese modelo, en el capítulo V se adaptará el estándar MPEG7-EHD para aceptar la salida del modelo y generar una representación de los bordes de la escena más fiel a lo que ve un macaco.

IV.2 Sistema visual de los macacos

Existe un gran número de estudios que analizan la actividad de las diferentes secciones del sistema visual de un macaco. Este apartado se apoyará en dichos estudios para mostrar el funcionamiento completo del sistema, así como para extraer información útil de este funcionamiento que pueda ser integrada en un modelo computacional de córtex válido para el procesamiento de imágenes.

En primer lugar, se mostrará de forma breve el sistema visual humano, por ser más familiar al lector. A continuación se mostrará el sistema visual de los primates y las partes principales del mismo, centrándose con mayor detalle en el córtex visual.

En la Figura IV-1 se puede ver la representación global del sistema visual humano.

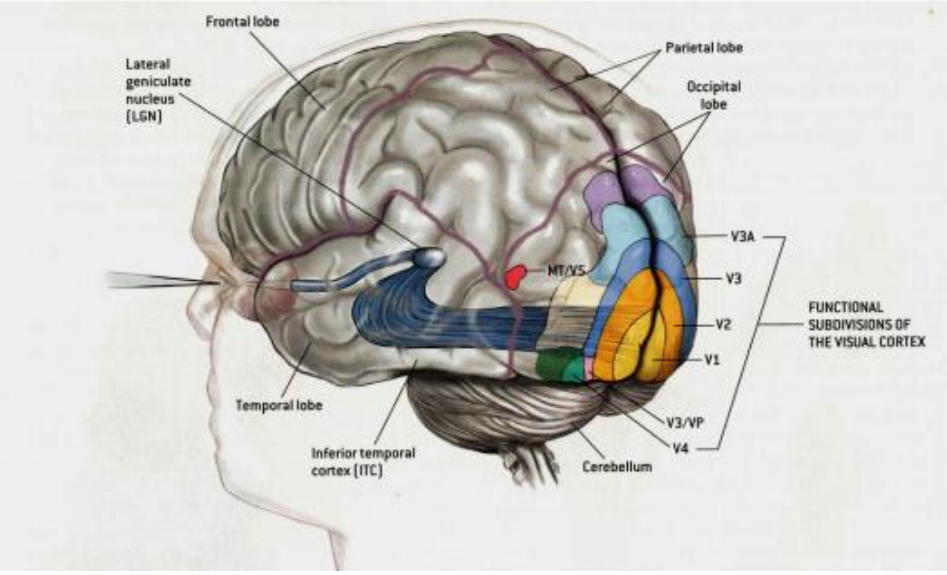


Figura IV-1: Sistema visual humano

En el sistema visual humano, toda imagen es capturada por el ojo. El ojo está compuesto por una lente, denominada cristalino, la cual proyecta las imágenes en el fondo del ojo, denominado retina. En ella, existen fotoreceptores que capturan la información de la imagen y la llevan hasta el *Lateral Geniculate Nucleus* (LGN), donde las señales de ambos ojos interactúan con otras señales provenientes de diferentes zonas del cerebro.

Desde este punto, las señales que representan la imagen se dirigen hacia la parte trasera del cerebro, denominada córtex visual. Este córtex está dividido en diferentes capas que poseen diferentes objetivos, por lo que la información capturada por el ojo entra por la capa inferior (V1) y progresa a través de todo el cerebro para que el humano sea capaz de percibir qué está presente en la escena.

A pesar de ser uno de los más conocidos por el público general, el sistema visual humano, y en general el cerebro humano, es uno de los más desconocidos por los investigadores. La dificultad para hacer experimentos sobre cerebros vivos hace que los investigadores trabajen con otro tipo de animales como son musarañas, gatos o macacos. Este último caso es realmente interesante en la comunidad neurocientífica, ya que el macaco posee uno de los cerebros que más se aproximan al del humano. De esta forma, su funcionamiento es muy similar al expuesto con anterioridad y el detalle de las diferentes capas de esta zona se puede ver en la Figura IV-2.

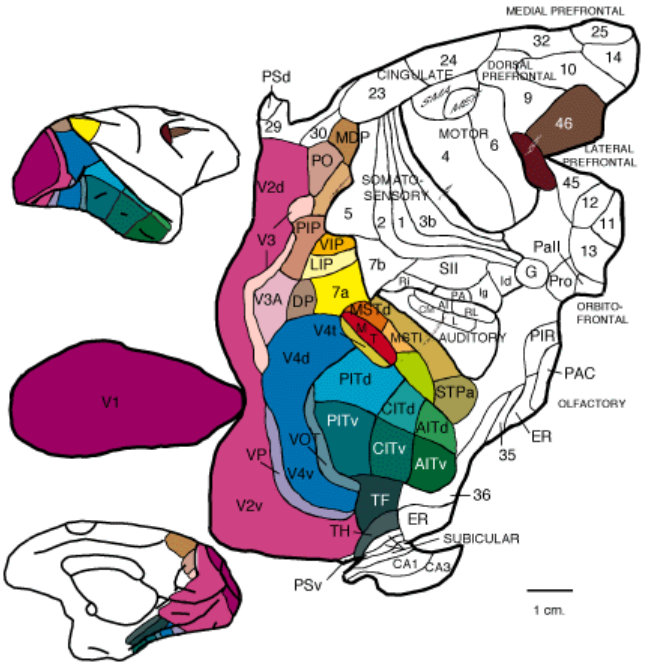


Figura IV-2: Mapa 2D del cerebro de un primate [FELL91]

Como se ha descrito, la señal proveniente de la retina y el LGN se introduce en el cerebro por la capa V1. A partir de ella surgen dos caminos de procesamiento de la información paralelos, aunque interconectados, que son el camino dorsal y el ventral. El camino

dorsal, también llamado el “camino del dónde”, está relacionado con la localización espacial de los objetos en la escena y el guiado de las acciones hacia estos objetos [KREI08]. En él la información fluye desde V1 hasta MT [KREI08].El camino ventral, está relacionado con el reconocimiento de los objetos [KREI08]. En este caso, la información fluye de V1 hacia V2 y V4 [GARR11].

El detalle de de estos caminos se puede ver en la Figura IV-3.

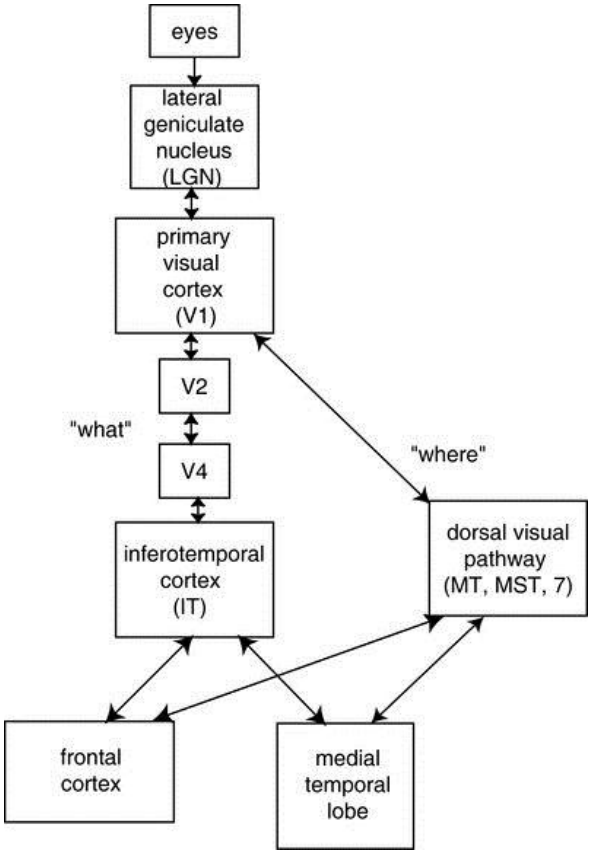


Figura IV-3: Representación esquemática de los caminos dorsal y ventral en el cerebro de los primates [KREI08]

A continuación, se mostrarán más en detalle los elementos principales involucrados en este flujo de información, con especial atención a la capa V1, ya que es el punto de llegada y salida de mucha información visual en el cerebro.

IV.2.1 Retina y Lateral Geniculate Nucleus

Como se ha visto, los fotones incidentes de la escena entran en el ojo a través del cristalino, cuya función es lograr la proyección de la escena en la retina, tal y como se puede ver en la Figura IV-4.

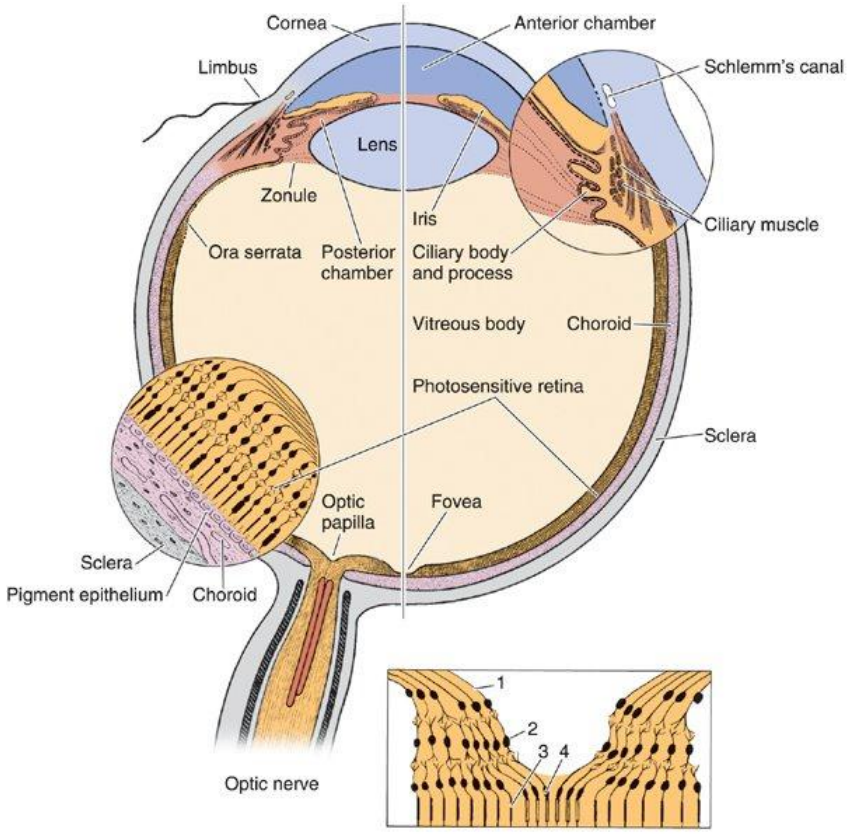


Figura IV-4: Anatomía del ojo [JUNQ05]

Todo el procesamiento de la información visual comienza en la retina. La retina está compuesta por 5 tipos de células, que poseen dos tipos de estructuras básicas en función de la dirección de la transmisión de la información. Estos tipos de células son los siguientes [GARR11]:

- **Fotorreceptores**: estas células son el punto en el que incide la luz exterior, y su respuesta depende de la luz recibida. Existen dos tipos: los conos, sensibles a tres

rangos de longitud de onda relacionados con los colores rojo, verde y azul; y los bastones, sensibles a la luz y dan una respuesta acromática.

- **Células horizontales**: su función principal es la de establecer conexiones horizontales.
- **Células bipolares**: existen 10 tipos diferentes de células bipolares en función de sus conexiones a los fotoreceptores y la anatomía de su axón. En general su campo receptivo está dividido en un centro y una periferia, que tienen efectos opuestos.
- **Células amácrinas**: llevan a cabo una gran cantidad de funciones de procesamiento visual, además de establecer conexiones horizontales con otras células.
- **Células ganglionares**: están situadas en la última capa de la retina, y sus axones se unen para formar el nervio óptico que va hacia otras zonas del cerebro.

Todos estos tipos de células se agrupan en varias capas, representadas en la Figura IV-5.

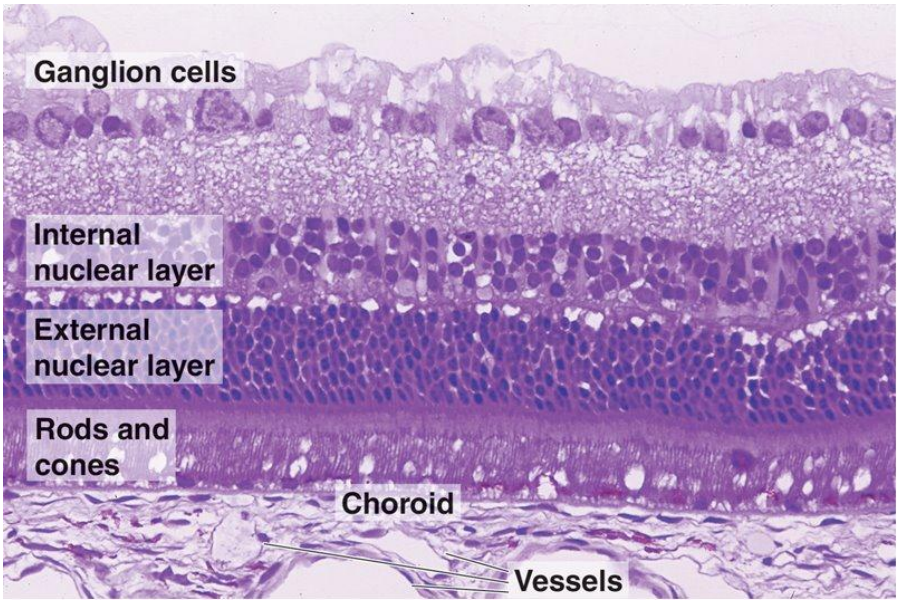


Figura IV-5: Sección histológica de la retina [JUNQ05]

Todas estas células trabajan de forma conjunta para generar una representación compacta de la información procedente del cristalino. En la Figura IV-6 se puede ver la interacción de todas ellas de forma esquemática.

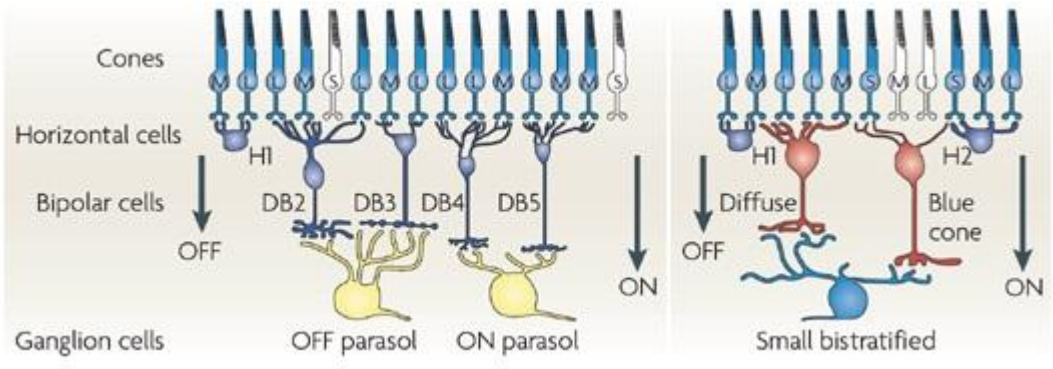


Figura IV-6: Diagrama a escala de las diferentes células de la retina y sus conexiones principales [NASS09]

Tras salir de las células ganglionares, el nervio óptico se dirige al Lateral Geniculate Nucleus (LGN) mediante tres tipos de neuronas diferentes: el koniocelular (K), parvocelular (P) y magnocelular (M) [NASS09]. Las células M son las células más grandes del LGN, capturan la información de los bastones y la utilizan para la percepción del movimiento y profundidad. Por otro lado, las células P perciben la forma de los objetos y parte de su color, principalmente obtienen información de los conos rojos y verdes. Por otro lado, están las células de tipo K, cuya información está relacionada con los conos de longitud de onda corta (azul), pero no está claro su funcionamiento [NUCL14].

Es desde este órgano desde el que se envía la información visual de entrada, ligeramente procesada, hacia el córtex visual primario del cerebro para que comience el trabajo de detección de objetos y movimiento en la escena. En la Figura IV-7 se puede ver la conexión entre la retina, LGN y el córtex primario.

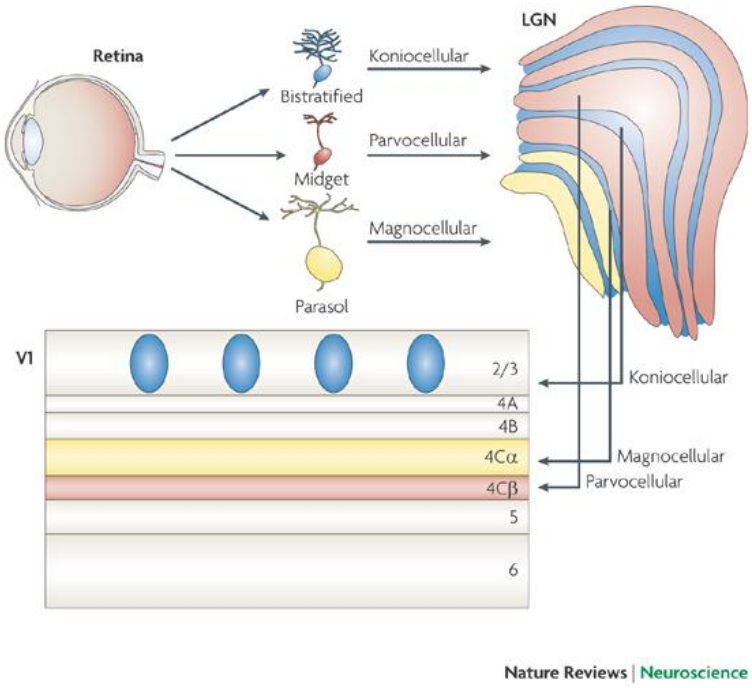


Figura IV-7: Conexiones entre la retina, el NGL y el V1 [NASS09]

IV.2.2 Córtex Visual

En este capítulo IV se trata al córtex, más exactamente al córtex visual, como el órgano del sistema visual que será modelado computacionalmente dentro del descriptor de bordes EHD del estándar MPEG7. El principal motivo es que hasta el córtex visual, ni en la retina ni en el LGN existe información de la textura de las imágenes observadas por el macaco. En la retina se ha visto la existencia de canales acromáticos y efectos de centro-periferia; y en el LGN hay un cierto realce de la forma de los objetos. Pero no es hasta el córtex visual donde la información de textura se comienza a representar y utilizar de una forma jerárquica.

El córtex visual está compuesto por las varias capas adyacentes, tal y como muestra la disposición anatómica de las capas V1 y V2 de la Figura IV-8.

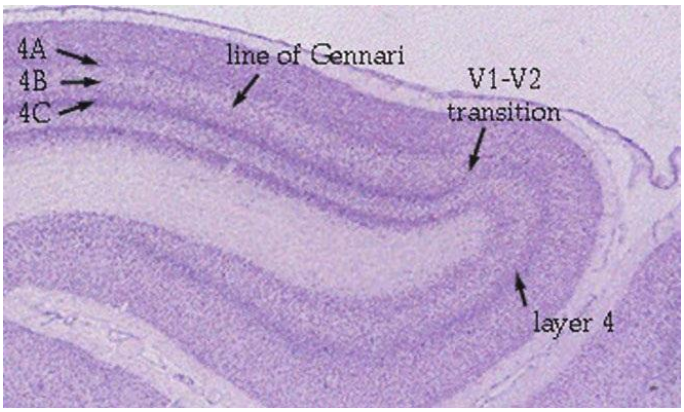


Figura IV-8: Sección histológica del córtex de un macaco, en el que se ve la transición entre las zonas V1 y V2, así como la capa 4 de V1

En los siguientes apartados se mostrará en detalle la capa V1, y de forma más genérica el resto de capas del córtex de los macacos.

IV.2.2.1 Córtex visual primario (V1)

El denominado córtex visual primario (V1) es probablemente una de las capas más estudiadas del córtex visual de los primates [TONG03]. La función básica de las neuronas pertenecientes a V1 es su excitación ante la existencia de un estímulo en una determinada orientación dentro de su campo visual local. La localización espacial está bien definida y es derivada de la evolución de la especie. La imagen de la retina se mapea en el córtex, pero no de forma lineal. Por ejemplo la fóvea de la retina está mapeada en una porción muy amplia del V1, fenómeno conocido como “cortical magnification”. Esto se debe a que la fóvea posee mucha información de detalle. Por ello, existe un mayor número de neuronas de la capa V1 que la representan.

En cuanto a su distribución física en el córtex visual, las neuronas se encuentran distribuidas por toda la región del cerebro perteneciente a V1. En la Figura IV-9, ilustración “A”, se puede observar esta distribución dentro de una sección del córtex.

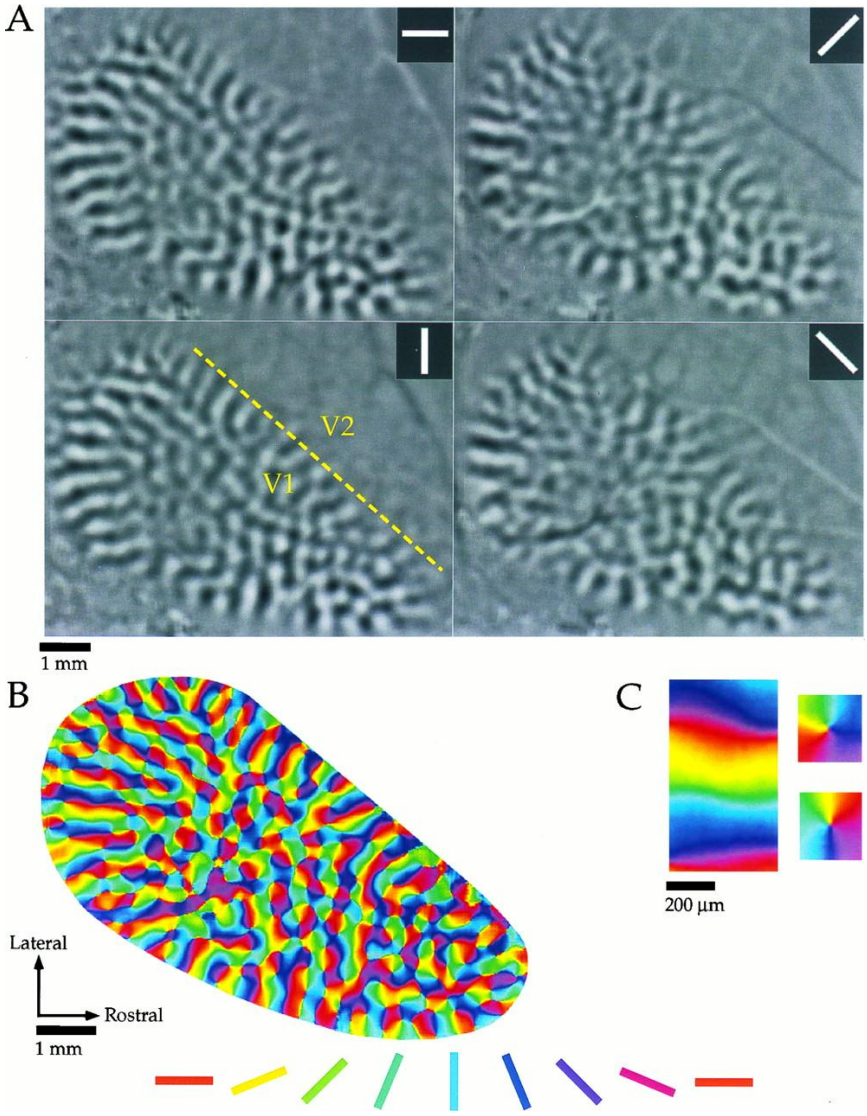


Figura IV-9: Patrones de orientación y conexiones horizontales en el córtex visual primario de la musaraña arbícola [BOSK97]

En la imagen izquierda-inferior de la ilustración “A”, se ve el patrón de actividad de las neuronas resultante tras la excitación mediante un estímulo vertical (una barra blanca a 90º sobre un fondo negro), donde el color negro indica un alto grado de excitación. Se comprueba que existe una clara división entre las zonas V1 y V2, de tal forma que las neuronas de V1 responden ante este tipo de estímulos, pero las V2 no. Por otra parte, en la figura derecha-inferior de la ilustración “A”, se comprueba que ante un estímulo a 135º las neuronas excitadas son diferentes, pero superpuestas con las anteriores. Este efecto se ilustra en la figura ”B”, donde cada patrón de color representa las neuronas excitadas ante una orientación. Este dibujo muestra la existencia de células adaptadas a una

orientación, llamada orientación de preferencia, y también muestra que estas células están rodeadas de otras con una preferencia diferente.

Los primeros estudios sobre las neuronas de la capa V1 los llevaron a cabo Hubel y Wiesel en la década de los 60, ganando en 1981 el Premio Nobel por dichos trabajos. Las neuronas de la capa V1 fueron clasificadas en dos tipos basándose en la estructura de sus campos receptivos: células simples y células complejas [HUBE59] [HUBE62]. En las simples, los campos receptivos poseen zonas separadas, denominadas ON y OFF: las regiones ON responden a barras blancas, mientras que las regiones OFF responden ante barras negras. Por el contrario, en las células complejas ambas regiones están superpuestas, por lo que todas las posiciones del campo receptivo responden a ambos tipos de barras. Estos efectos se pueden ver de forma más clara en la Figura IV-10, que visualiza los campos receptivos de una célula simple (izquierda) y de una célula compleja (derecha). En dicha figura, se indican las regiones OFF en rojo y en verde las ON. Se puede ver cómo en las células complejas ambas regiones están superpuestas, por lo que su respuesta es constante ante variaciones de posición o frecuencia. También se intuye que será invariante a las rotaciones de las excitaciones, ya que no tiene una orientación preferida.

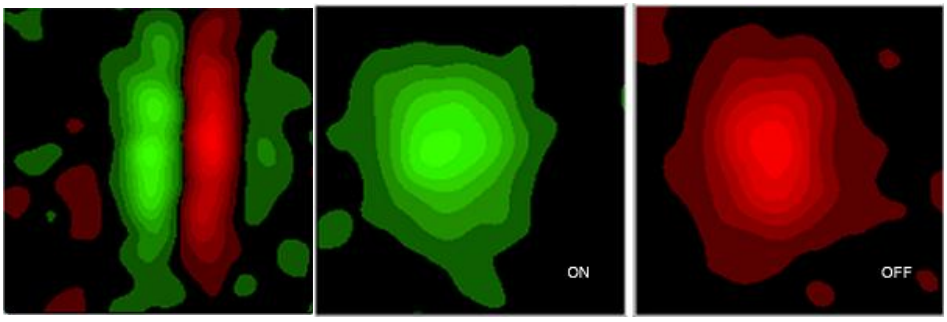


Figura IV-10: Campos receptivos de una célula simple (izquierda) y de una célula compleja (derecha) [DEAN95]

En estos estudios se mostraron tres puntos fundamentales en relación a estas células, y actualmente se mantienen vigentes [HUBE98]:

1. Las neuronas presentes en V1 tienen una localización en el campo visual que obtiene la máxima respuesta. Esta zona del campo visual se denomina campo receptivo ("*receptive field*" en inglés).
2. El campo receptivo cambia de forma suave a lo largo del espacio, formando el denominado *retinotopic map* de la escena.
3. Las neuronas individuales de V1 son las responsables de la representación cerebral de la existencia de una barra en una orientación específica dentro su campo de recepción.

Esta última observación es la más interesante desde el punto de vista de la visión artificial y análisis de imágenes. De hecho, éste es el punto de partida de numerosos trabajos de modelización del córtex visual de los primates [CARA97] [OLSH96]. El modelo de neurona de V1 más básico y más utilizado por la comunidad científica en el ámbito de la visión artificial es un filtro de Gabor orientado. El principio fundamental de un filtro de Gabor se puede definir como una señal portadora sinusoidal modulada por una señal gaussiana. En una dimensión, se puede ver gráficamente representado en la Figura IV-11.

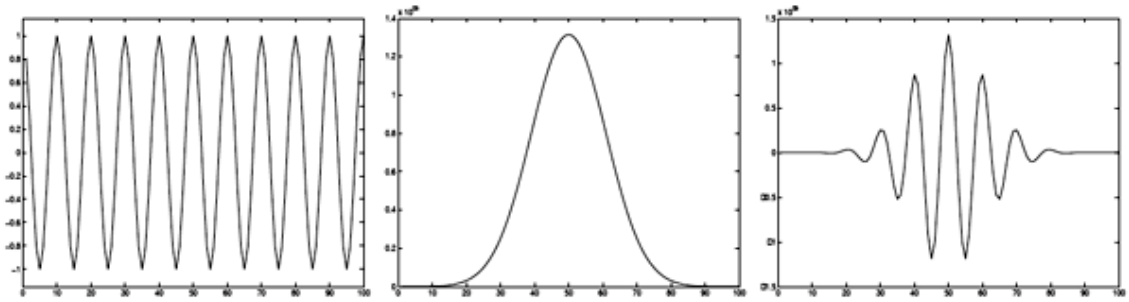


Figura IV-11: Idea de la generación de un filtro de Gabor en una dimensión (derecha) mediante una portadora sinusoidal (izquierda) y una moduladora gaussiana (centro)

En realidad, un filtro de Gabor es una función bidimensional de la siguiente forma:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = m(x', y'; \sigma, \gamma) * s(x'; \lambda, \theta, \psi)$$

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

siendo la parte real:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

y la parte imaginaria:

$$g(x', y'; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x' + \gamma^2\, y'^2}{2\sigma^2}\right) * sen\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

donde:

$$x' = x\, cos\theta + y\, sen\theta$$

$$y' = -x\, sen\theta + y\, cos\theta$$

A pesar de este modelo 2D, las neuronas de V1 son mucho más complejas y son múltiples los estímulos que los excitan. Los principales estímulos o atributos en los que son selectivas estas neuronas son los siguientes [CARA12]:

- **Orientación**: Esta es la principal característica de las células de V1. Esta propiedad define que éstas neuronas son muy selectivas ante la orientación del estímulo, fundamentalmente en las células simples, no como en la retina o en el LGN.
- **Frecuencia espacial**: La segunda gran propiedad de estas células, es que están muy especializadas en la detección de una frecuencia concreta, y para ello tienen un ancho de lóbulo concreto, definiendo la frecuencia como la inversa de la distancia entre lóbulos. Como se ha mostrado, y al igual que en el caso de la orientación, las células simples de V1 poseen múltiples regiones de ON y OFF, y su frecuencia de repetición deberá coincidir con la frecuencia del estímulo para dar una respuesta máxima.
- **Dirección del movimiento**: En un segundo plano se encuentra el dominio del tiempo. Las células en V1 son capaces de trabajar mediante la integración del estímulo en el tiempo. Además, estas células son extremadamente selectivas en la dirección en la que se mueve el estímulo. Para explicar este concepto se puede extender un modelo de campo receptivo basado en filtros de Gabor al dominio temporal. En la Figura IV-12 se puede ver cómo un estímulo vertical moviéndose

en horizontal se puede representar en un cubo 3D, donde dos dimensiones reflejan el plano de movimiento y la tercera dimensión refleja el tiempo. La modelización de la preferencia ante un movimiento determinado de las células de V1 se puede hacer incluyendo un filtro de Gabor en el plano x-y, pero también otro en el plano del movimiento x-t.

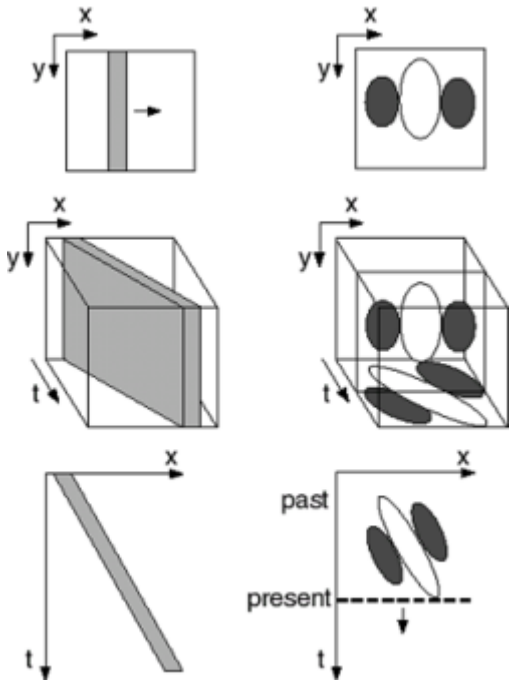


Figura IV-12: Modelo temporal del campo receptivo. A la izquierda el movimiento de una barra en vertical y su representación en un espacio 3D. A la derecha, el modelo de filtros Gabor en los mismos planos [CARA99]

- **Frecuencia temporal**: Siguiendo con el aspecto dinámico, aunque con una selectividad mucho menor a la anterior, se encuentra la preferencia de algunas células del córtex visual primario por una velocidad concreta de movimiento. Aunque muchos trabajos dan por hecho este fenómeno, este punto no está claro, puesto que también existe una dependencia relacionada con el tamaño del estímulo.
- **Color**: Algunas de las células de V1 también poseen una preferencia al color. Es en la retina de los primates, más exactamente en las células ganglionares, donde se transforma la información capturada de color en tres direcciones cardinales: rojo-verde, azul-amarillo y blanco-negro. En principio, se suponía que las células de V1 tenían un amplio rango de preferencia al color, pero ahora se sabe [HORW12]

que las preferencias están focalizadas y relacionadas con las tres direcciones cardinales.

Aun siendo Gabor el modelo más popular, las últimas investigaciones en las neuronas V1 han revelado que éstas no son neuronas con una respuesta estática en el tiempo, sino que la salida ante una excitación concreta varía a lo largo del tiempo. Se pueden distinguir dos fases durante la respuesta de una neurona de V1 ante un estímulo: la respuesta *temprana* y la respuesta *tardía*.

Durante la respuesta temprana, entre 40 ms y 100 ms desde el comienzo del estímulo, las neuronas de V1 están muy focalizadas ante la respuesta en un conjunto limitado de estímulos.

Una aproximación sencilla es suponer que durante esta primera fase de respuesta de las neuronas de V1 se puede modelar la estructura de las mismas mediante un filtro de Gabor configurado en una orientación y frecuencia determinada.

En la segunda fase de la respuesta, a partir de 100 ms desde el comienzo del estímulo, las neuronas de V1 comienzan a incluir información de la organización completa de la escena [LAMM00]. Para ello, existen dos mecanismos principales que actúan sobre las células y son las *conexiones de feedback* desde otras capas del córtex visual, así como las *conexiones horizontales o laterales* entre neuronas de la misma capa del córtex visual. Esta información agregada en las células V1 se cree que puede tener un efecto modulador sobre su señal de salida [ANGE03] [HUPE01], aunque también se ha comprobado que modifican su campo receptivo y su forma si las señales de feedback provienen de las zonas más altas del córtex [SILL06].

Uno de los primeros estudios más detallados en relación a las conexiones horizontales y la selectividad a la orientación fue desarrollado por Bosking et al. en 1997 [BOSK97]. Las conexiones horizontales se forman en las células piramidales, las cuales se pueden encontrar también en otras capas del córtex visual. Estas neuronas son las únicas que proyectan axones fuera de ellas. Cada célula piramidal posee una única dendrita, llamada “dendrita apical”, la cual se introduce y se separa en las partes superiores del córtex. A su vez, posee un axón extremadamente largo que permite cubrir regiones muy amplias del

cerebro, y es éste el que permite el tipo de conexiones horizontales dentro de una misma capa del córtex.

El pilar principal de dicho estudio fue definir que, por un lado, las neuronas de V1 poseen una selectividad específica ante una orientación determinada, tal y como afirmaban Hubel y Wiesel en los años 60. Adicionalmente, este estudio indicaba la existencia de largas conexiones horizontales entre neuronas de V1, que podían llegar hasta varios milímetros en paralelo a la superficie cortical. Estas conexiones no eran arbitrarias sino que las conexiones horizontales que se producían en el córtex visual de la musaraña arbícola era entre neuronas co-orientadas. Además, estas conexiones se generaban principalmente en la misma dirección de la preferencia de orientación y, por tanto, eran conexiones entre células co-alineadas. Así, para una neurona con preferencia al estímulo a 45º, sus conexiones horizontales se originan con otras neuronas de la misma preferencia de 45º. Pero no con todas ellas, sino que preferiblemente se conecta con neuronas situadas a lo largo de la línea situada a 45º de ellas. Este efecto se puede ver en la Figura IV-13.

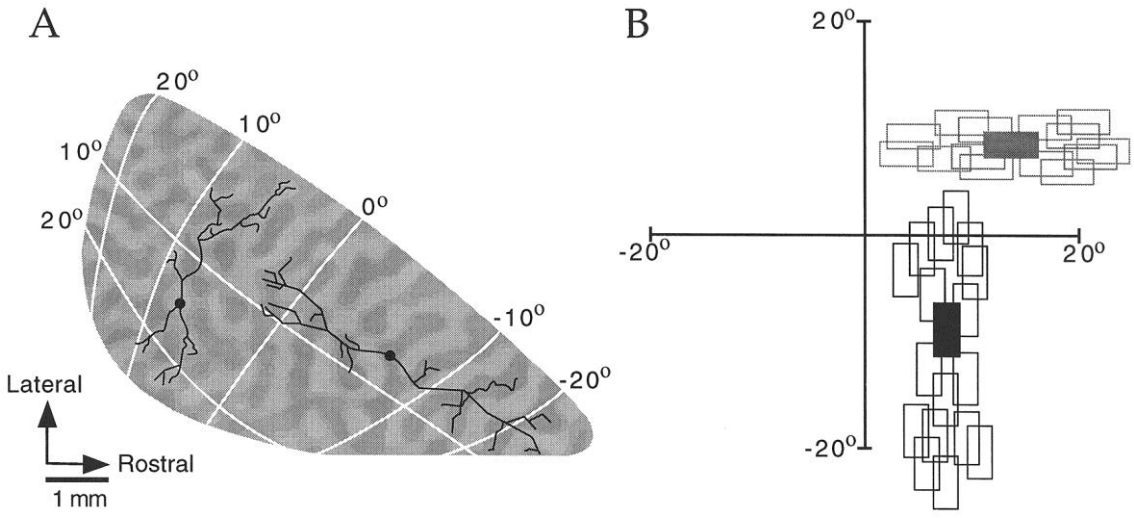


Figura IV-13: Conexiones horizontales entre neuronas co-orientadas y co-alineadas [BOSK97]

En la imagen A de la Figura IV-13 se ven las ramificaciones de las conexiones de dos células en el mapa visual. Mientras una de ellas tiene preferencia por las conexiones a 90º, la otra posee una mayor tendencia a las conexiones a 0º. Lo mismo se ve en la figura B, pero representando la preferencia a la orientación con la relación de aspecto de los

cuadrados que simulan las neuronas. Aquellas neuronas que tienen una preferencia de orientación de 90º, y están representadas por cuadros más altos que anchos, prefieren una conexión con neuronas a 90º.

Este efecto es corroborado en un trabajo más reciente [CHIS03], donde especifican más en detalle la función de estas conexiones. Por una parte, indican que dichas conexiones horizontales se dan en la "capa 2/3" de V1, mientras que no suceden en la capa 4. Además, en ese trabajo se analiza el campo de suma de las células, y se comprueba que existe un fenómeno de *facilitación* relacionado con las conexiones horizontales, la co-orientación y la co-alineación. Este fenómeno permite al conjunto de células conectadas responder ante estímulos específicos en el eje afín a su respuesta, sin perder resolución y rellenando huecos existentes entre excitaciones.

En cuanto a las conexiones de feedback, éstas provienen de otras zonas del córtex visual y del cerebro (Figura IV-14) y, a pesar de que no se conoce de forma exacta su función, sí se sabe que generalmente tienen un efecto modulador sobre la salida de la neurona [CARA12].

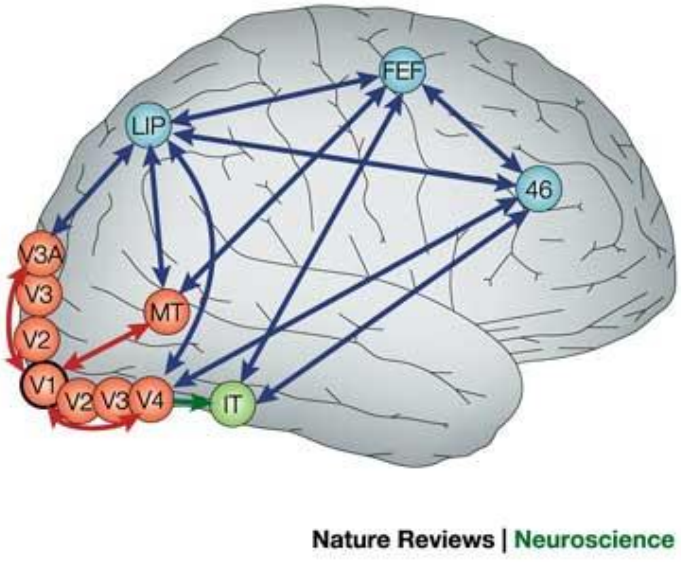


Figura IV-14: Capas del córtex visual y conexiones con otras áreas cerebrales [TONG13]

Son numerosas las interacciones entre diferentes partes del cerebro como puede verse en la Figura IV-14, y muchas de ellas tienen un impacto directo en el comportamiento de las neuronas de V1. Hasta ahora se ha visto cómo diferentes regiones del córtex se mapean directamente como diferentes localizaciones espaciales en la escena visualizada por la retina. Además, cada región tiene una respuesta determinada ante estímulos específicos, como por ejemplo, la orientación. Pero también se ha mostrado que la respuesta de una zona del cerebro no es independiente del resto, y que existen interacciones entre regiones que permiten procesar la información de forma dependiente del contexto [ADES10].

El resultado provocado por las interacciones entre diferentes regiones tiene, a grandes rasgos, dos tipos de efectos: efecto *facilitador* y efecto *supresor*. El primero se ha desgranado al hablar de las conexiones horizontales entre neuronas de V1, pero es el segundo el efecto más fuerte y abundante dentro del córtex visual.

Este efecto trata de suprimir la respuesta de una región de neuronas ante una actividad en una tercera, y todo ello mediado por diferentes tipos de interacciones de largo alcance entre neuronas. Uno puede suponer que ya que las conexiones horizontales están relacionadas con el efecto de facilitación, también lo estarán con el efecto de supresión. Pero esto no es necesariamente cierto. Existen diversos estudios que afirman que ese tipo de conexiones son demasiado lentas y demasiado cortas como para explicar completamente el efecto de supresión producido en una célula [ANGE03]. Por ejemplo, en [SCHW06] se muestra como la inactivación de áreas superiores del córtex se produce al mismo tiempo que disminuye el efecto de supresión, por lo que ambos estarán relacionados y son zonas muy distantes dentro del córtex, haciendo que sean las señales de feedback las que están más relacionadas con el efecto de supresión. Aun así, este fenómeno todavía no ha podido ser validado mediante mediciones neuronales electrofisiológicas, ya que no es sencillo capturar y separar señales de feedback de señales forward retrasadas en el tiempo. Sea como fuere, ambos tipos de conexiones, de feedback y horizontales, juegan un papel importante en la modulación de la respuesta de las neuronas y en especial en el efecto de supresión, de tal forma que las conexiones horizontales puedan influir debido a la estimulación en el campo cercano, mientras que las conexions de feedback influirán debido a la excitación en el campo lejano [SERI03].

Existen diferentes tipos de supresión de la señal de una neurona de V1. Uno de los ejemplos de supresión de respuesta es la “supresión de orientación cruzada”. Ante un estímulo en forma de aspa, y según los estudios que se han descrito en este capítulo, la respuesta lógica de las células de V1 deberá ser la activación de dos grupos neuronales: las que responden a una de las orientaciones del aspa y las que responden a la otra orientación. Por tanto, la respuesta acumulada será la suma de ambas excitaciones. La realidad es diferente y, ante dicho estímulo, la respuesta que se produce es menor que la suma de ambas, debido al efecto supresor que tienen unas contra las otras.

Por otro lado, el efecto de supresión más conocido es el de “supresión del entorno”. Esta supresión se define como el fenómeno que se da cuando un estímulo que excita a una neurona crece más allá de su campo receptivo, y la respuesta de la neurona comienza a disminuir, en vez de mantenerse constante. Así, por ejemplo, utilizando como estímulo una barra en la orientación preferente de la neurona, y con una longitud infinitesimal la cual crece a lo largo del tiempo, se verá que la respuesta de la neurona de V1 crecerá a lo largo del tiempo. Pero no será un crecimiento infinito, sino que existe un punto en el que la respuesta disminuye. No todas las células del córtex visual poseen esta propiedad, y las que la poseen se denominan “end-stopped”, y tienen una mayor activación cuando se producen cambios bruscos de la excitación, por ejemplo, en los bordes de los objetos. Es por ello que se cree que estas neuronas juegan un papel muy importante en la habilidad de los primates de identificar formas [KIPE02].

Hasta ahora se han descrito varias maneras en las que diferentes zonas del córtex visual primario interactúan entre sí para modificar su respuesta, pero evidentemente también existen conexiones e interacciones entre capas de V1 y otras capas del córtex visual, como son V2, V3, V3A, V4 y V5 (también conocida como MT-MTemporal). Un aspecto fundamental en estas conexiones es que no son unidireccionales, sino bidireccionales, generando una proyección hacia atrás de la información en capas superiores del córtex sobre la V1. Esta re-proyección es la fuente de diferentes modificaciones en las neuronas de V1, y está relacionada, entre otros, con la atención [TONG03].

Además de estas conexiones intra-córtex, existen numerosas conexiones con otras secciones del sistema visual (Figura IV-14), como son el área intra-parietal lateral (LIP), los campos frontales de los ojos (FEF) y el área 46; los cuales están involucrados directamente con la atención visual y la planificación motora.

Todas estas interacciones hacen de la capa V1 del córtex visual una de las más activas e importantes del cerebro.

IV.2.2.2 Capas superiores del córtex visual

Comparado con V1, la comunidad científica ha dedicado mucho menos esfuerzo en caracterizar y modelar otras capas del córtex visual, posiblemente por su difícil acceso físico.

Desde la entrada de la imagen al cerebro del primate por la capa V1, esta señal se traslada hacia la parte superior del área visual del cerebro, y en los niveles más altos la información extraída tiene una mayor componente “conceptual” y una menor componente de “señal”. En la Figura IV-15 se muestra el modelo computacional HMAX, que trata de modelar las diferentes capas del córtex visual. Como se ha descrito anteriormente, el objetivo de la estructura jerárquica del cerebro es que a medida que la información de entrada avanza en el córtex, la información generada y representada es mayor: desde bordes orientados en la capa V1 hasta detectores de caras específicos en las capas interferotemporales.

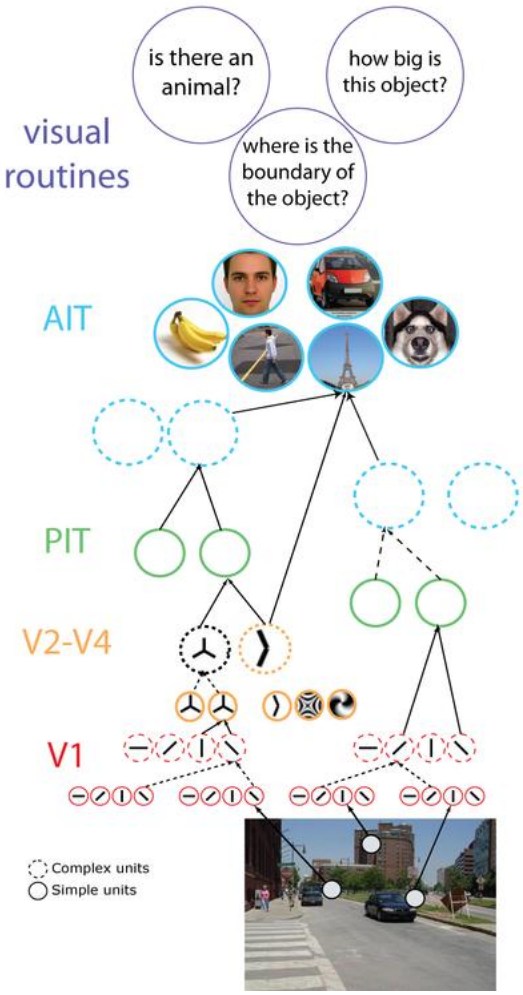


Figura IV-15: Gráfico del modelo computacional HMAX del córtex visual [POGG13]

En general, la mayoría de los investigadores están de acuerdo en que las neuronas de la capa V2 del córtex visual son sensibles a ángulos, o en su forma más simple, dos barras orientadas que se intersecan [ITO04]. Además, también se ha demostrado que la capa V2 es donde existen neuronas que responden a bordes ficticios [HEYD84].

La región V3 es una región con mucha controversia. Si bien en los humanos no está claro dónde comienza y dónde acaba esta región con respecto a V2, en los macacos es una zona minúscula del córtex visual que principalmente se centra en establecer conexiones entre otras regiones.

La capa V4 es mucho más parecida a la capa V1, puesto que tiene preferencias por orientaciones, frecuencias y colores [VISU14]. La mayor diferencia es que las preferencias

son mucho más complejas y, por ejemplo, se representan espirales o patrones concretos de textura [PASU99].

En las partes superiores del córtex visual, las denominadas interferotemporales, se ha descubierto que existen neuronas que detectan objetos complejos, como por ejemplo caras, de forma invariante a múltiples transformaciones, como rotación, escala o movimiento de ojos [KREI08].

Por lo tanto, se ve cómo un modelo completo de córtex sería un buen sistema para detección de objetos en imágenes. En ello se ha trabajado mucho a lo largo de los años, pero no se ha logrado ningún sistema real completo debido a las incertidumbres y desconocimiento que todavía existe del cerebro de los primates.

IV.3 Estado del arte en el modelado del córtex visual primario

Hasta ahora se ha visto una descripción biológica del córtex visual, en especial del macaco, y mucho más en detalle, de su capa V1. Ésta capa es de gran importancia para la detección de bordes, por lo que partiendo del hecho de que se busca una sustitución de la detección de bordes en el estándar MPEG7, se ve cómo la capa V1 es la candidata ideal para ser modelada e insertada en el estándar.

Como se ha mostrado, se dan una gran cantidad de efectos en V1 debido a la gran cantidad de conexiones y comunicaciones entre neuronas. Existen tres grandes formas de modelar todo lo que sucede en el córtex visual [SERI04]:

- **Modelos funcionales**: son aquellos que buscan caracterizar las propiedades de las neuronas y sus respuestas en base a un único algoritmo de procesamiento visual. Por su naturaleza, estos modelos no tienen porqué corresponder al 100% con la estructura real de las neuronas, sino que deben obtener la misma respuesta y presentar las mismas propiedades. Este modelo está guiado por el conocimiento del córtex y por sus respuestas.
- **Modelos estructurales**: son los que buscan caracterizar de forma precisa los mecanismos biológicos y físicos, que son los responsables de los datos fisiológicos de salida del sistema. Este modelo está guiado sólo por el conocimiento del córtex, suponiendo que si se modela de forma similar, la respuesta será similar.

- **Modelos optimizados**: son los que buscan predecir los datos fisiológicos en base a modelos matemáticos genéricos, entrenados mediante la optimización de sus parámetros para dar como salida esos datos fisiológicos. La principal diferencia con los anteriores reside en que en ningún momento se busca una modelización que explique todos los aspectos biológicos, sino que se fuerza a un modelo flexible a que responda como se espera. Este modelo está guiado puramente por las respuestas del córtex.

De cara a esta tesis, es importante no perder de vista el objetivo final: un nuevo algoritmo de anotación de imágenes. Por ello, se descarta analizar los modelos estructurales, ya que se pierden en aspectos biológicos complejos para un sistema computacional. Por otra parte, tampoco se busca que la salida del sistema propuesto sea idéntica a las neuronas del córtex, como proponen los modelos optimizados, ya que puede no ser útil. El objetivo de esta tesis está más alineado con la búsqueda de un modelo que tenga principios biológicos y respuestas plausibles válidas para el análisis de imagen. Por ello, en este apartado, primero se analizarán en detalle los **modelos funcionales** de córtex visual primario.

Estos tres tipos de modelos tratan sobre el modelado teórico de la neurociencia, en el que se comprueba que matemáticamente todo es correcto. A pesar de ello, apenas existen implementaciones prácticas de estos modelos teóricos, ya sean funcionales, estructurales u optimizados, aplicados en el campo de la visión artificial. Sólo existen algunos modelos que poco tienen que ver con los anteriores y, que en esta tesis, se denominarán **modelos computacionales**. Estos modelos son implementaciones reales aplicadas a imágenes 2D genéricas, y se estudiarán en segundo lugar en este apartado.

IV.3.1 Modelos funcionales de la capa V1 del córtex

Para poder generar modelos teóricos del funcionamiento del córtex visual, idealmente sería necesario agrupar todos los efectos que se han visto en el apartado IV.2.2.1 bajo una teoría unificada. Pero debido a la diferente naturaleza de las conexiones existentes, las discrepancias entre estudios, la aparente contrariedad entre ciertos efectos, como por ejemplo, el efecto de facilitación que se convierte en supresión [ICHI07], o

comportamientos múltiples en neuronas, entre otros, no existe un modelo funcional que agrupe el funcionamiento completo del córtex visual.

Por ello, los modelos funcionales comienzan construyéndose sobre una base simple, e introduciendo los diferentes efectos. El comienzo más habitual es partir de la base del denominado campo receptivo clásico (CRF-Classical Receptive Field). Para una neurona de V1, su CRF comprende lo que de forma clásica Hubel y Wiesel han definido como campo receptivo: la zona del mapa visual donde la neurona generará una respuesta ante un estímulo. En su forma más simple, una neurona de V1 es un filtro lineal que se aplica sobre una zona de la imagen, el CRF, y tiene como respuesta una suma de ese CRF ponderada por unos pesos definidos por el perfil de la célula.

Este perfil, como se ha visto anteriormente, se puede definir en base a filtros de Gabor [SCEN01]. La salida de la neurona matemáticamente puede ser positiva o negativa, pero biológicamente sólo será positiva y se lanza a partir de un umbral. Este modelo es correcto para el caso de las células simples, pero en el caso de las células complejas, se debe modelar en base a una combinación de varias respuestas que se agregan para dar la salida final (ver Figura IV-16) [MOVS78].

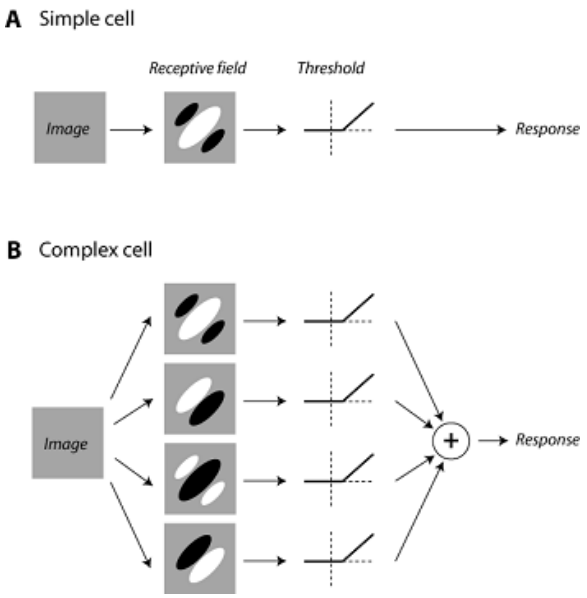


Figura IV-16: Modelos básicos descriptivos de una célula simple y otra compleja de la capa V1 [MOVS78]

Como se ha visto en el apartado IV.2.2.1, existen cuantiosos efectos fuera de este campo receptivo que condicionan la respuesta de una neurona de forma no lineal [SERI04]. En la Figura IV-17 se ven la mayoría de las influencias existentes.

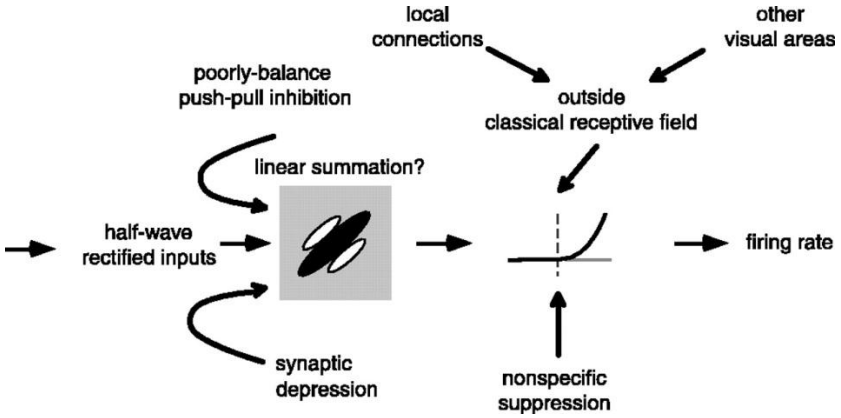


Figura IV-17: Modelo genérico de neurona de V1 [CARA05]

La Figura IV-17 fue publicada en [CARA05] donde además del modelo lineal del campo receptivo clásico se añaden numerosos efectos que influyen en su respuesta. Como se ve, se propone un modelo en dos etapas, donde la primera se presupone un filtrado lineal, mientras que la segunda añadiría los efectos no lineales. Esto se debe a que caracterizando únicamente el CRF no es posible modelar la respuesta conjunta de todas las neuronas de V1, teniendo en cuenta sus interacciones, y es aquí donde entra en juego el concepto de *campo receptivo extendido* (ERF – Extended Receptive Field). ERF se le considera a la unión topológica entre el CRF y su entorno. Centrándose en modelar la supresión del entorno, ya que es el efecto más destacado de estas interacciones entre neuronas del córtex visual primario, diversos autores han intentado describir las propiedades del ERF en un único marco teórico. Existen dos grandes modelos funcionales en los que se trabaja en la actualidad: el denominado *Diferencia de Gausianas* (DoG o modelo resta) y el *Ratio de Gausianas* (RoG o modelo división) [SERI04] (Figura IV-18).

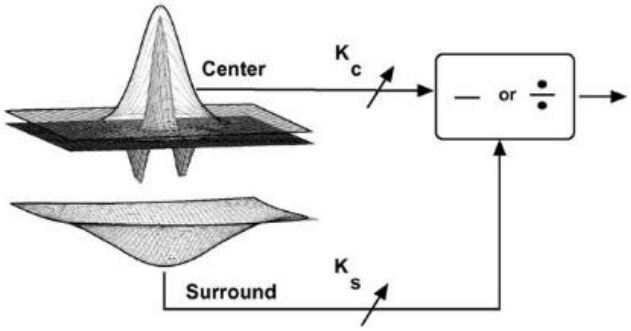


Figura IV-18: Modelos DoG y RoG [SERI04]

Sceniak et al. [SCEN99] [SCEN01] propusieron el modelo DoG donde el ERF podía ser visto como una entidad única descrita como una resta de gaussianas. Así, una primera gaussiana (L_c) se corresponde con la parte de respuesta del CRF, y podría relacionarse con la moduladora del filtro de Gabor que representa el CRF; mientras que una segunda gaussiana (L_s) establecería la contribución supresora del entorno. De esta forma, la respuesta de una neurona a una rejilla circular de radio *x* sería:

$$R(x) = K_c L_c(x) - K_s L_s(x)$$

donde K_c y K_s son las ganancias de los mecanismos de centro y entorno; L_s y L_c son gaussianas de la siguiente forma:

$$L_c = \int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_c}\right)^2} dy$$

$$L_s = \int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_s}\right)^2} dy$$

donde σ_s y σ_c representan el alcance espacial de ambas componentes.

En [CAVA02] se recalca que tanto la parte supresora, el entorno como la excitadora, el centro, son mecanismos independientemente regulados, por lo que es correcto que ambos conceptos aparezcan en la ecuación del modelo DoG con sus propias ganancias y extensiones.

Como se puede entrever, este es un modelo lineal que no puede explicar fenómenos no lineales que se producen, como por ejemplo, la variación no lineal de la curva de

respuesta de una neurona producida por el cambio del contraste. Por ello, Cavanaugh et al. [CAVA02] propusieron una alternativa basada en el cociente entre dos gaussianas, donde la respuesta ante la misma rejilla circular que anteriormente sería:

$$R(x) = \frac{K_c L'_c(x)}{N + K_s L'_s(x)}$$

siendo K_c y K_s las ganancias de los términos centro y entorno, y L_c y L_s:

$$L'_c = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_c}\right)^2} dy \right)^2$$

$$L'_s = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_s}\right)^2} dy \right)^2$$

La particularidad de este modelo es que encaja perfectamente con el modelo de normalización que se presupone existe en otras partes del cerebro [CARAN12b], según el cual, la respuesta de cada neurona se normaliza dividiéndola por las respuestas de un conjunto de neuronas que la rodean. Se ha comprobado cómo este modelo permite explicar diferentes aspectos de varios sistemas sensoriales, como la representación de olores e incluso la modulación de la atención, convirtiéndolo en un candidato a modelo universal de neurona [CARAN12b].

Como se comprueba matemáticamente, tanto el modelo RoG como el DoG buscan una modelización de la supresión por el entorno. En ambos casos si la respuesta del entorno es muy alta, ante la misma respuesta del centro, entonces la respuesta final de la célula será menor. Pero en ningún caso se han introducido otros efectos supresores ni facilitadores, tales como los que se han descrito en el apartado IV.2.2.1. Estos modelos gozan de gran aceptación, ya que son capaces de explicar muchos de los efectos observados en las células. Por ejemplo, se ha observado que el campo receptivo de una neurona aumenta su tamaño cuando el contraste disminuye. Esto se puede tener en cuenta considerando que los términos K_c, K_s, o σ_c son variables y dependientes de dicho contraste. Aun así, siguen existiendo detractores de este tipo de modelos funcionales, que tienen mayor preferencia por modelos estructurales, ya que los primeros no son capaces de predecir el funcionamiento interno de las neuronas, sólo el externo. Por ejemplo, una de las grandes preguntas realizadas sobre el modelo más común (RoG) es: *¿Cómo se implementa una división en una neurona?* [SERI04].

Para terminar, y desde un punto de vista neurocientífico, cabe destacar que estos modelos funcionales son modelos que permiten responder ante estímulos más o menos sencillos. Cuando estos modelos se aplican a imágenes más complejas, y en especial cuando el estímulo varía en el tiempo, existen otros fenómenos que no son capaces de modelar [CARA12], como por ejemplo, las no-linealidades en el dominio temporal. Además, se han propuesto modelos más complejos que intentan dar respuestas más definidas en el espacio y en el tiempo, como por ejemplo [RUST05], donde utilizan un gran número de filtros superpuestos para modelar una única neurona. Todos estos fenómenos quedan fuera del alcance de esta tesis, por ser específicos del modelado de la respuesta exacta de la neurona, y en adelante no se tratarán.

IV.3.2 Modelos computacionales del córtex

El córtex visual, y en especial su capa V1, es una de las más estudiadas de todo el cerebro humano. Su relación directa con la detección de patrones visuales sencillos la hacen susceptible de ser modelada en un algoritmo de visión artificial. Es por esto que durante las últimas décadas son varios los autores que han utilizado estos conceptos y los han implementado en sus algoritmos. Debido al alcance y objetivo de estos algoritmos, la mayor parte de los autores han modelado el córtex como un conjunto de filtros de Gabor [POGG13], tal y como se ha visto en el apartado IV.2.2.1. Estos filtros se han utilizado en numerosas aplicaciones, sobre todo en el campo de la identificación de texturas [CLAU00], pero también en otros campos como el reconocimiento facial [LIU02] o anotación de imágenes [MAKA10].

Aún así se ha visto cómo el córtex visual de los primates es mucho más rico que unos filtros de Gabor y, por ello, otros autores han trabajado en realizar modelos, con mayor o menor precisión, de una o varias partes del córtex visual.

En relación a esta tesis, el mayor exponente de modelo computacional es el propuesto por Serre et al. [SERR05] [SERR07] para la detección de objetos. Para una imagen de entrada, el sistema extrae una serie de características y con ellas alimenta un clasificador típico.

El elemento destacable de este modelo son las características extraídas. Cada elemento del conjunto de características se extrae combinando la respuesta de varios detectores de bordes locales, tal y como llevan a cabo las células complejas del córtex visual.

[SERR05] trata de modelar el córtex visual como cuatro capas de unidades computacionales, que permitirán extraer las características visuales. En ellas, las "unidades simples" (S), combinan sus entradas de una forma Gaussiana, para aumentar la selectividad de objetos. Estas unidades se alternan con las "unidades complejas" (C), que consultan sus entradas y seleccionan los máximos de las entradas, introduciendo una invarianza gradual a la escala y a la traslación. Esta arquitectura se puede ver en la Figura IV-19.

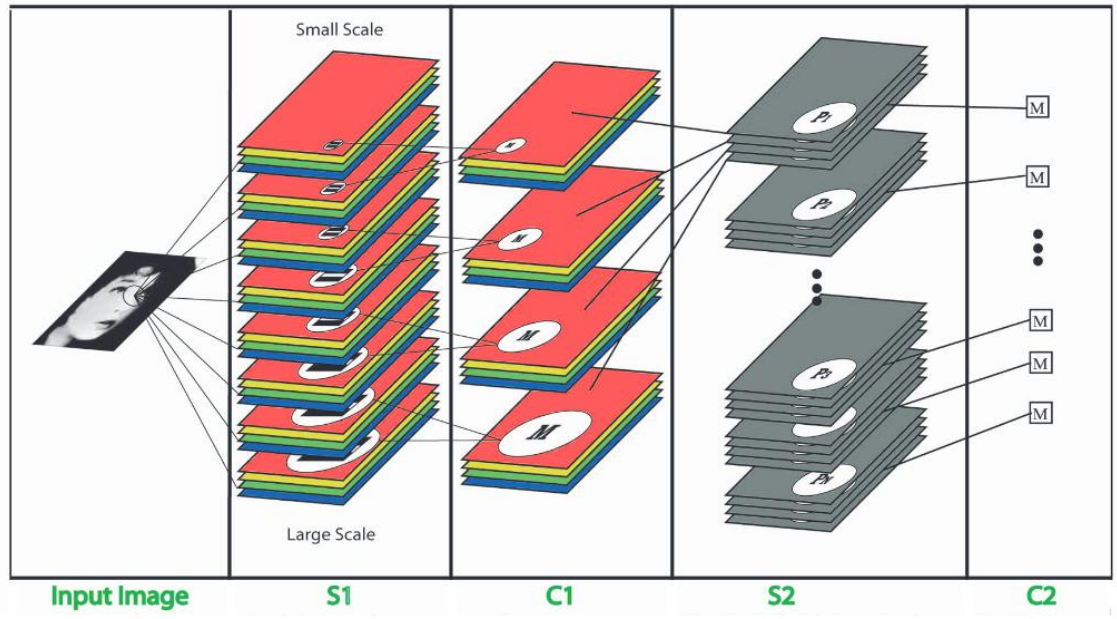


Figura IV-19: Arquitectura propuesta en [SERR07]

En resumen, la arquitectura de capas que extraen las características visuales es la siguiente: El córtex primario (V1) se modela mediante las dos primeras capas (S1 y C1). En la primera (S1) se aplican una serie de filtros de Gabor, mientras que la segunda capa (C1) se encarga de la obtención de una serie de valores del resultado de los filtros de Gabor. Tras ellas, se encuentran las otras dos capas (S2 y C2). Mientras que en S2 opera con los valores obtenidos por C1, en C2 obtiene el máximo de todas las posiciones y escalas, por lo que extrae las características finales invariantes a escala y posición. Estas últimas capas dos modelan la capacidad de aprendizaje y generalización de los sistemas motor y visual de los primates [SERR05].

Además de éste, también hay otros modelos computacionales menos relevantes para esta tesis como son el modelo Neocognitron [FUKU80] (Figura IV-20), que posee

específicamente capas de neuronas simples y complejas; o modelos más modernos, genéricos y no tan adaptados al modelado del cerebro humano como las técnicas de Deep Learning (ver apartado II.4.2.6).

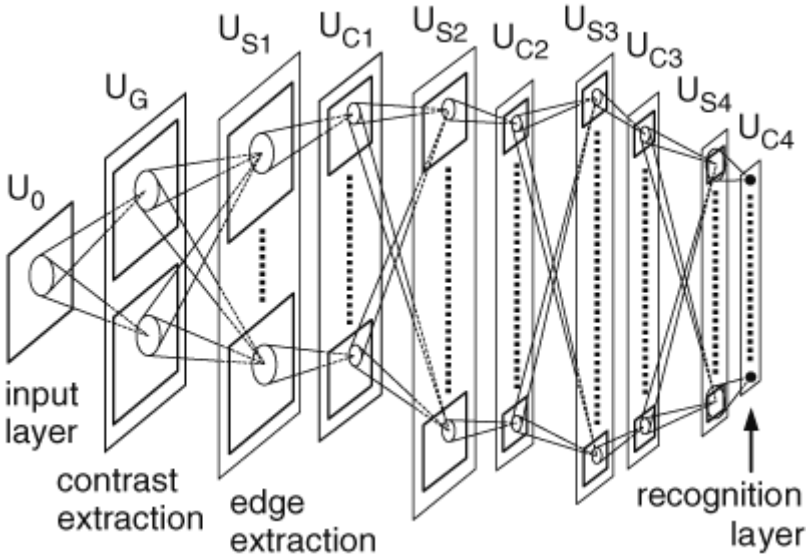


Figura IV-20: Arquitectura típica del Neocognitron [FUKU07]

IV.4 Propuesta de nuevos modelos del córtex

El objetivo de este capítulo de la tesis es el de proponer nuevos prototipos de modelos funcionales y computacionales de la capa V1 del córtex visual, para lo que haremos uso de los conceptos existentes la literatura de neurociencias. Este objetivo se centrará en modelar fenómenos que suceden dentro de dicha capa y de sus neuronas, con el objeto de introducir una mejor detección de los bordes en el estándar MPEG7. Los fenómenos a introducir serán aquellos que presenten un mayor interés para este objetivo y para el campo del análisis de imágenes estáticas, quedando fuera fenómenos referentes a variaciones espacio-temporales o fenómenos puramente biológicos. Una vez propongamos los prototipos de los modelos, seleccionaremos los parámetros de configuración y en el apartado IV.5 mostraremos las respuestas de los modelos ante excitaciones comunes y realizaremos la selección del mejor de ellos.

IV.4.1 Propuesta del modelo de neurona base

El primer paso para hacer una propuesta de modelo de neurona será seleccionar y definir unos conceptos base que marcarán el diseño de los modelos. Tras el análisis realizado de la literatura neurocientífica (apartados IV.2 y IV.3), se tomarán como conceptos base los siguientes:

- **Modelado de las capas inferiores de V1 de un macaco**. Esta tesis se centrará en modelar las células simples de las capas bajas del córtex visual de los macacos. Específicamente se centrará en modelar las neuronas simples de la “capa 2/3” del córtex visual primario (V1).
- **Parte real de un filtro de Gabor como modelo estructural del campo receptivo clásico de la neurona**. Se ha visto cómo diversos autores consideran esta opción como completamente válida, por lo que será la base. En cuanto al campo receptivo extendido (ERF), se realizará mediante conexiones de las células adyacentes, tal y como muestra la Figura IV-21.
- **Efecto de supresión.** Se ha visto que dentro de ERF las neuronas actúan como moduladoras de la señal de respuesta de la neurona central. Principalmente, este efecto es supresor, por lo que es necesario modelarlo.
- **Conexiones horizontales y efecto de facilitación**. En el apartado IV.2 se ha comprobado que un aspecto importante en las neuronas es su comunicación con neuronas de la misma capa mediante conexiones horizontales. Especialmente, de cara al análisis de imagen, se puede ver útil en el caso de estímulos largos en el espacio pero intermitentes o esquinas, donde varias neuronas se activarán y otras teóricamente no. La facilitación es capaz de unir esos estímulos intermitentes, y por ello, creemos que el efecto de facilitación es algo muy importante a modelar en el campo del análisis de imagen, y lo haremos considerándolo perteneciente al ERF.

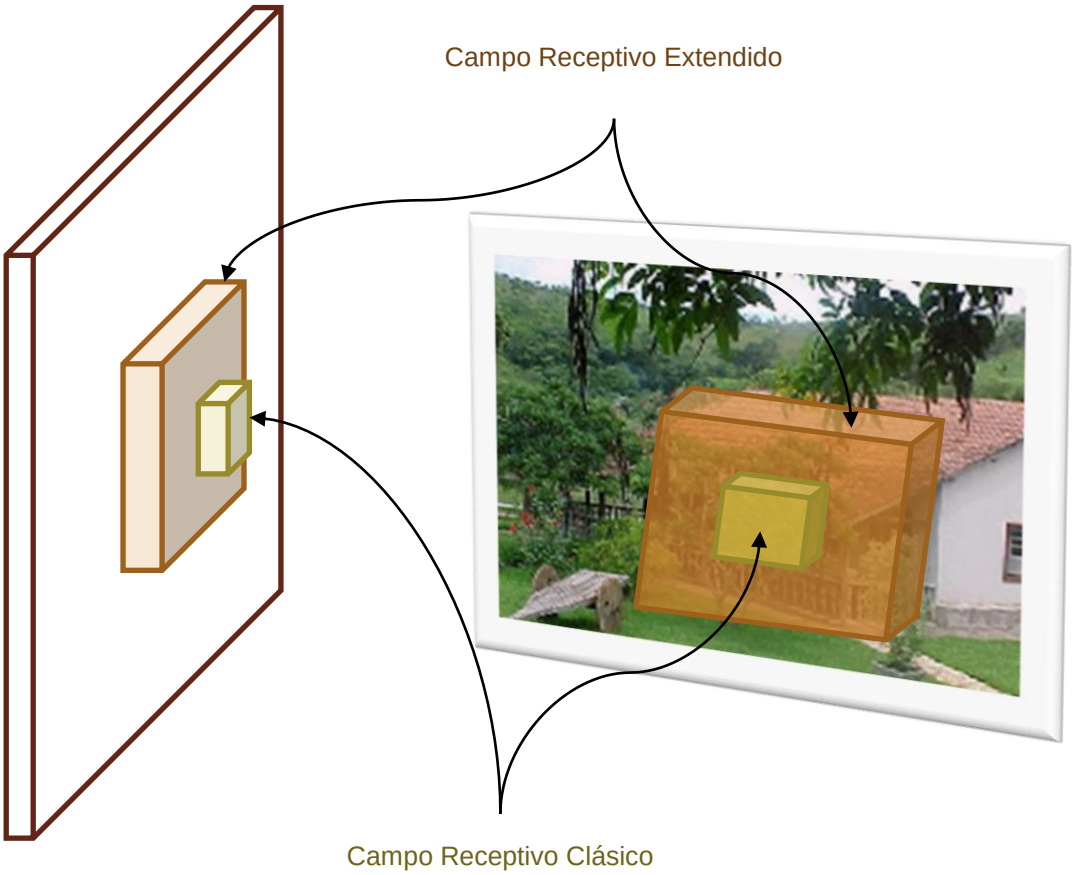


Figura IV-21: Campos receptivos extendidos y clásicos ante una imagen

Con todos estos conceptos y en base a otros trabajos [CARA05], en esta tesis proponemos un modelo de célula en dos fases (Figura IV-22) relacionadas con las dos fases de respuesta (temprana y tardía) que tienen: ante un estímulo de entrada, en la primera fase, la neurona ponderará de forma lineal los píxeles de su CRF en base a un filtro de Gabor, y obtendrá una señal de salida. En una segunda fase, esta señal se verá influenciada por los efectos de facilitación y supresión generados por los mecanismos de centro-periferia y conexiones horizontales dentro de su ERF, para finalmente generar una salida en base a un umbral.

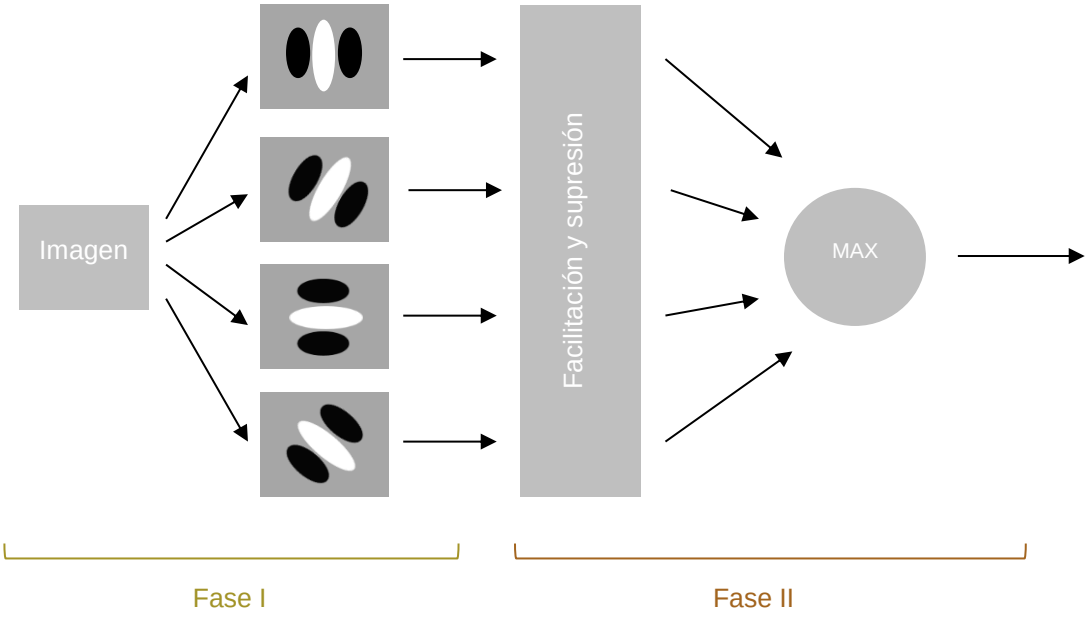


Figura IV-22: Modelo de neurona bifase propuesto en esta tesis

Hasta ahora hemos definido los conceptos base de la literatura y hemos hecho una nueva propuesta de neurona del córtex visual. A continuación trabajaremos en introducir este modelo de neurona dentro de marcos teóricos vistos en el apartado IV.3 para generar varios modelos de córtex visual.

En los siguientes sub-apartados describiremos con detalle los modelos propuestos en esta tesis. Finalmente, los modelos propuestos se compararán en base a la salida de sus neuronas generada ante un estímulo y se seleccionará el modelo que dé una salida más similar a la real, para introducir dicho modelo en el estándar MPEG7.

IV.4.2 Diseño del modelo computacional SERRE de córtex visual

El primer diseño de córtex V1 que vamos a proponer se basará fuertemente en los trabajos de Serre et al. [SERR05] [SERR07], y se denominará como *Modelo SERRE*.

Como se ha visto en el estado del arte, el modelo propuesto por Serre et al. [SERR07] posee diferentes capas que modelan los mecanismos del córtex para el reconocimiento de objetos. En esta tesis sólo se busca modelar la capa 2/3 del córtex, por lo que sólo se tendrá en cuenta las capas S1 y C1 del modelo de Serre et al.

Tabla IV-1: Parámetros de configuración de las neuronas utilizados en [SERR07]

Banda Σ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Tamaño del filtro s	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
σ	28 & 3.6	4.5 & 5.4	6.3 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.2 & 22.8
Tamaño rejilla N^{Σ}	8	10	12	14	16	18	20	22
Orient. ɵ	$0; \frac{\pi}{4}; \frac{\pi}{2}; \frac{3\pi}{4}$							
Tamaño del parche n_i	4x4; 8x8; 12x12; 16x16 (x4 orientaciones)							

Según sus autores, la capa S1 de su modelo se asemeja a las células simples del córtex visual mediante un conjunto de filtros de Gabor de diferentes propiedades. En su modelo existen ocho capas, cada una de ellas posee dos configuraciones de los filtros de Gabor.

Ante una imagen de entrada, en la capa S1 se aplicarán un conjunto de filtros de 4 orientaciones (θ) y 2x8 escalas (σ). Cada escala posee un tamaño fijo en píxeles del filtro (s) y una frecuencia (λ) específica. Esta salida se almacena en base a las 8 bandas y se pasa a la siguiente capa C1, que modela las células complejas. Para ello, realiza una operación de *maxpooling* sobre las respuestas de los filtros de cada banda, en base a su orientación y escala, en una región del espacio.

El modelo que proponen Serre et al. no cumple exactamente con los requisitos y objetivos marcados al inicio del apartado, por lo que es necesario realizar ciertas modificaciones. En concreto, la capa C1 permite una interacción entre células de S1, por lo que se va a presuponer que es un mecanismo de interacción centro-periferia o conexiones horizontales. De esta forma, a efectos de esta tesis la capa C1 del modelo de Serre et al. se considera dentro del modelo de célula simple. A partir de las respuestas generadas, el objetivo de la tesis es obtener la orientación del estímulo. Para ello, añadiremos una última capa a este modelo, de tal forma que para cada píxel del espacio donde existe

salida de las neuronas, se busque la orientación de la neurona como aquella que provee el máximo de las respuestas. Será ésta entonces la orientación del estímulo en ese píxel.

Así, hemos realizado la primera propuesta de modelo de córtex visual, en el que nos hemos basado en Serre et al., modificándolo acorde a los conceptos vistos en el apartado IV.4.1.

Con este modelo modificado se van a realizar unas pruebas básicas que permitirán conocer la respuesta del mismo. Dentro del campo de neurociencia, se usan diferentes rejillas como estímulo para ver las respuestas de las neuronas. En este caso, para comenzar a ver ciertos efectos, se usará una imagen con cuatro rejillas de diferente frecuencia y similar orientación (Figura IV-23).

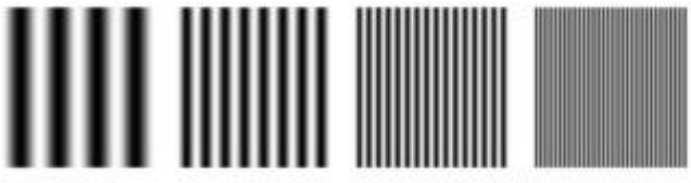


Figura IV-23: Estímulos de pruebas

Si aplicamos este estímulo a un modelo de V1 con sólo un tipo de neurona, es decir, una única estructura de Gabor, se ve la respuesta de la capa S1 del modelo SERRE en la Figura IV-24.

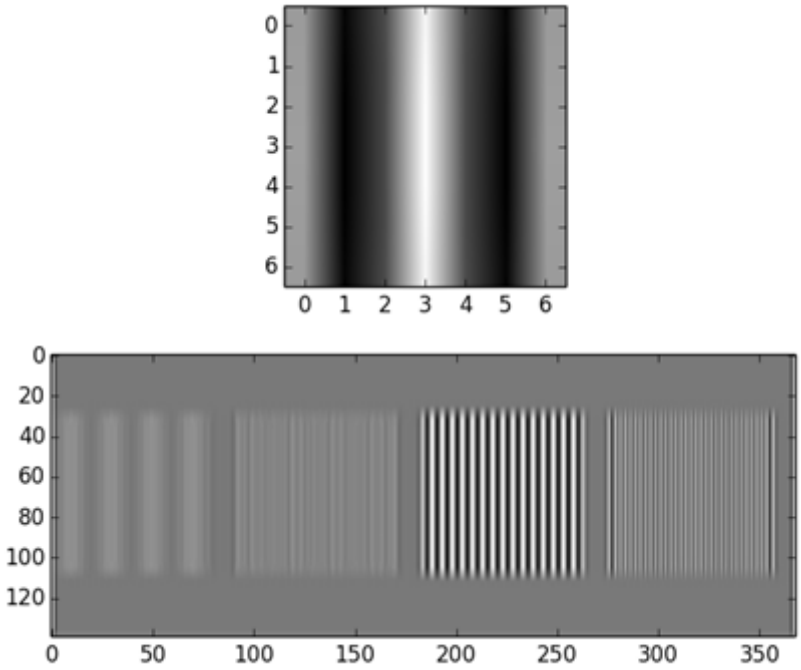


Figura IV-24: Arriba: Estructura de las neuronas del modelo de prueba. Abajo: Respuesta ante un estímulo de rejillas de diferentes frecuencias.

En la parte superior de la Figura IV-24 se ve la estructura de las neuronas, mientras que en la inferior se ve la respuesta de las mismas. El valor más blanco significa una mayor respuesta, por lo que se comprueba cómo la rejilla número tres es aquella que posee una frecuencia similar a la de la estructura neuronal utilizada.

Por otra parte, aplicando el modelo SERRE modificado, y gracias a las conexiones entre neuronas proporcionadas por su capa C1, se genera el resultado de la Figura IV-25.

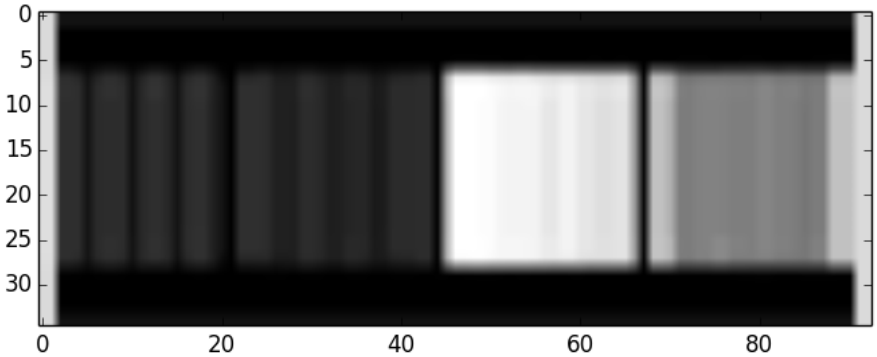


Figura IV-25: Respuesta de la Capa C1 del modelo SERRE

Efectivamente, se comprueba que en toda la zona de la rejilla número tres la respuesta es muy alta, por lo que se intuye que su funcionamiento es correcto. Hay que tener en cuenta que esta prueba se ha realizado sobre una única orientación. Debido a ello, para terminar con esta primera prueba del modelo SERRE, se va a realizar la misma operación con todas las orientaciones y escalas de su modelo, para a continuación calcular el máximo de respuesta en cada punto. La visualización con diferentes colores de este máximo se presenta en la Figura IV-26.

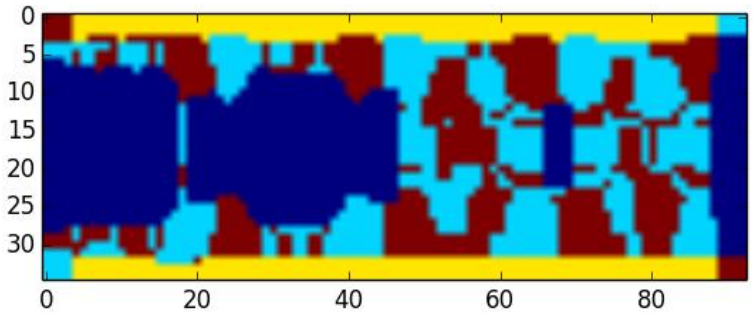


Figura IV-26: Respuesta final del modelo de células simples de Serre et al.

El color azul oscuro indica la presencia de una orientación de 0º en los bordes (barras verticales), el azul más claro indica la presencia de bordes a 45º, el amarillo indica 90º (barras horizontales) y el rojo/granate muestra los bordes detectados a 135º. Hay que tener en cuenta que la convolución introduce los artefactos en los bordes y, por tanto, es necesario descartar los bordes de las imágenes. Como se comprueba, existen numerosas zonas oblicuas, sobre todo en la tercera rejilla. Esto se debe a que en una frecuencia y escala determinadas la respuesta es mucho mayor en vertical y, por ello, el máximo se detecta en los ángulos oblicuos en vez de en los ángulos a cero grados, quizá debido a solapamientos de frecuencia. Esto hace que este modelo no sea todo lo bueno que se desea para este tipo de detecciones de bordes.

IV.4.3 Diseño de los modelos DG1 y DGF

A diferencia del caso anterior, los modelos DG1 y DGF se basarán en el concepto DoG. El modelo funcional DoG fue propuesto por Sceniak et al. [SCEN01] y se basa en la existencia de un campo receptivo clásico (CRF), en el que la neurona central tiene una

respuesta concreta y un campo receptivo extendido (ERF), el cual introduce información del entorno para suprimir la respuesta de la neurona central.

El modelo DoG se define matemáticamente de la siguiente forma:

$$R(x) = K_c L_c(x) - K_s L_s(x)$$

donde *R(x)* es la respuesta de la neurona en la posición *x*; K_c y K_s son las ganancias de los mecanismos de centro y entorno; L_s y L_c son gaussianas que integran la información espacial de la siguiente forma:

$$L_c = \int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_c}\right)^2} dy$$

$$L_s = \int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_s}\right)^2} dy$$

donde σ_s y σ_c representan el alcance espacial de ambas componentes.

En el artículo original, se indica que esta resta de gaussianas se aplica sobre los píxeles de forma directa. Así, L_c actuaría de forma similar al CRF, mientras que L_s sería el ERF, que actuaría suprimiendo la señal central. Si esta resta de gaussianas se aplica de forma directa sobre los píxeles de una imagen, el resultado obtenido no es muy prometedor. Para el caso del estímulo de prueba en forma de rejilla (Figura IV-23), el resultado aparece en la Figura IV-27.

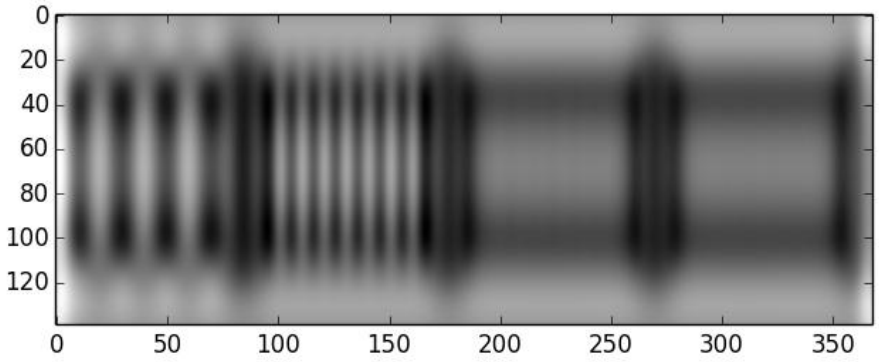


Figura IV-27: Respuesta del modelo DoG puro basado en el artículo seminal [SCEN01].

Se ve que hay mucha difusión de la respuesta, cuando ésta debería ser clara. Por ello, en este apartado vamos a proponer dos nuevos modelos, que funcionan en base a resta de gaussianas, pero que generen una salida más plausible en base a los conceptos definidos en el apartado IV.4.1.

En nuestros modelos el CRF se modelará con la parte real de los filtros de Gabor, mientras que el ERF se modelará como una diferencia de gaussianas aplicada a las respuestas espaciales de los CRF de su entorno. Así, todos los CRF de una región entrarán en juego, bien con un efecto supresor (L_s), bien con un efecto positivo (L_c). Ambos términos no integrarán los valores de los píxeles de la imagen. Por una parte, L_c es directamente la salida del filtro de Gabor (el CRF), mientras que L_s integra las respuestas de los CRF, tal y como se ha descrito anteriormente en el modelo de neurona en dos fases (IV.4.1). El modelo esquemático de esta computación se muestra en la Figura IV-28.

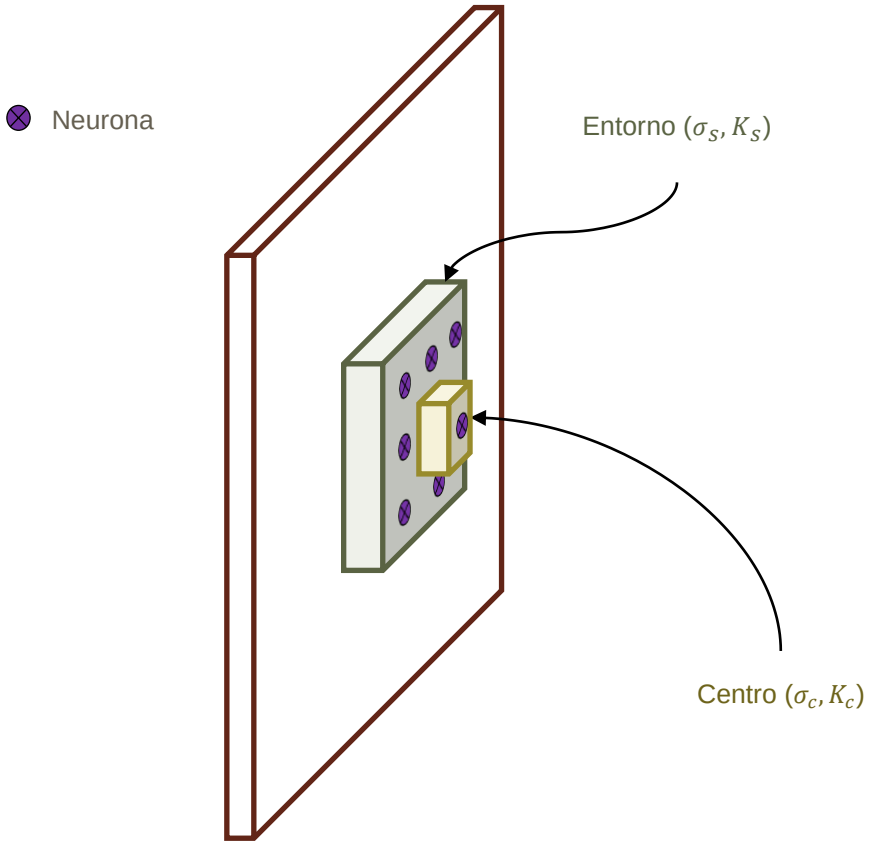


Figura IV-28: Esquema de un modelo de relación entre neuronas basada en el mecanismo de supresión.

En este modelo tenemos varios parámetros a definir: los parámetros del filtro de Gabor y los parámetros de la resta de gausianas (K_c, K_s σ_s y σ_c). En cuanto al filtro de Gabor, se utilizarán los mismos valores expuestos por Serre et al., ya que son parámetros obtenidos para el córtex del macaco. En cuanto a los del modelo DoG, para este primer test se realizará una prueba con varios parámetros plausibles pero con el objetivo de comprobar si el efecto del modelo propuesto es el deseado. Sceniak et al. especifican que L_c es la envolvente gaussiana del filtro de Gabor. Por ello, σ_c tendrá el mismo valor que tiene el primer filtro de Gabor de la tabla Tabla IV-1. Si usamos una σ_s similar, mientras que K_c =1 y K_s = 0.8; y finalmente se rectifican las salidas de los filtros de Gabor [CARA05], el resultado del modelo DG1 que proponemos es el que aparece en la Figura IV-29.

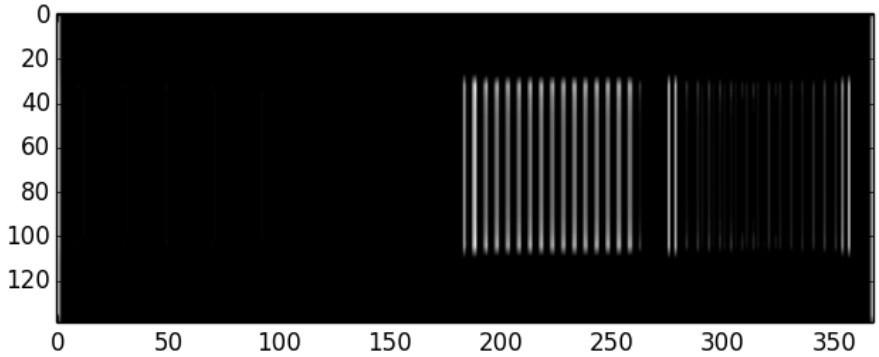


Figura IV-29: Respuesta del modelo DG1 propuesto en esta tesis

El resultado es muy positivo, ya que sucede lo que se predice: para el tercer estímulo, la señal de activación en el centro de la rejilla es de menor intensidad que en el borde, debido a la presencia del efecto de supresión. Es necesario recordar que se ha visto anteriormente que el filtro de Gabor que se está usando como CRF para estas pruebas está sintonizado a una frecuencia similar a esta tercera rejilla. El efecto de supresión de señal se puede observar de forma más acentuada en el cuarto estímulo, donde directamente las barras centrales son casi nulas.

Hasta ahora, se ha visto cómo el efecto supresor de las neuronas del entorno se puede modelar de forma satisfactoria con el modelo que hemos denominado *DG1* (*Diferencia de Gaussianas 1*). Pero en esta tesis también queremos modelar el efecto facilitador generado por las conexiones horizontales, tal y como se ha descrito en el apartado IV.4.1. Para ello, hemos introducido un nuevo término en forma de suma para generar el modelo *DGF* (*Diferencia de Gaussianas con Facilitación*):

$$R(x) = K_c L_c(x) - K_s L_s(x) + K_f L_f(x)$$

En este caso, existen tres componentes que se agregan para dar la respuesta final de una neurona. El primer elemento (L_c) es la respuesta de la neurona aplicando el filtrado de Gabor (CRF). El campo de supresión (L_s) integra todas las respuestas de los CRF en una región adyacente y equidistante, y actúa como parte negativa en la agregación. Finalmente, tal y como se ha visto en el análisis del córtex visual, la facilitación (L_f) se activa en base a conexiones co-orientadas de los diferentes CRF. Esta orientación viene

fijada por la orientación de los filtros de Gabor que se están aplicando, y por el cambio del ratio de aspecto de la gaussiana correspondiente. Así, para un filtro de Gabor a 90º, el campo de facilitación se girará también a 90º, y se considerarán los CRF de esa región, y no los ortogonales (Figura IV-30). En la

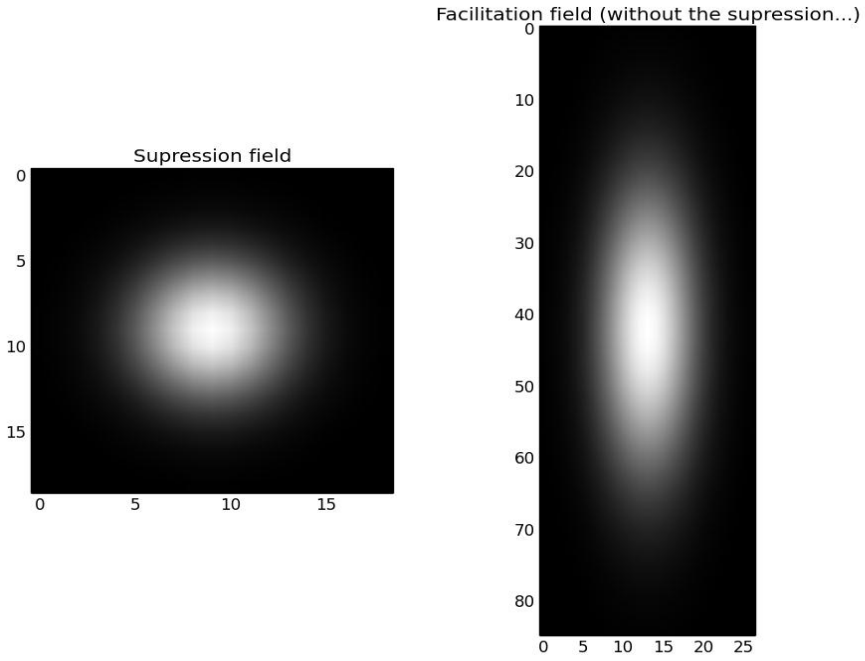


Figura IV-30: Gaussianas que modelan el efecto supresor y el facilitador

En este modelo de neurona, existe un solo mecanismo que afecta a la respuesta obtenida por la integración del centro de la neurona y es el mecanismo de supresión del entorno. La relación de ambas componentes se pueden ver de forma esquemática en la Figura IV-28.

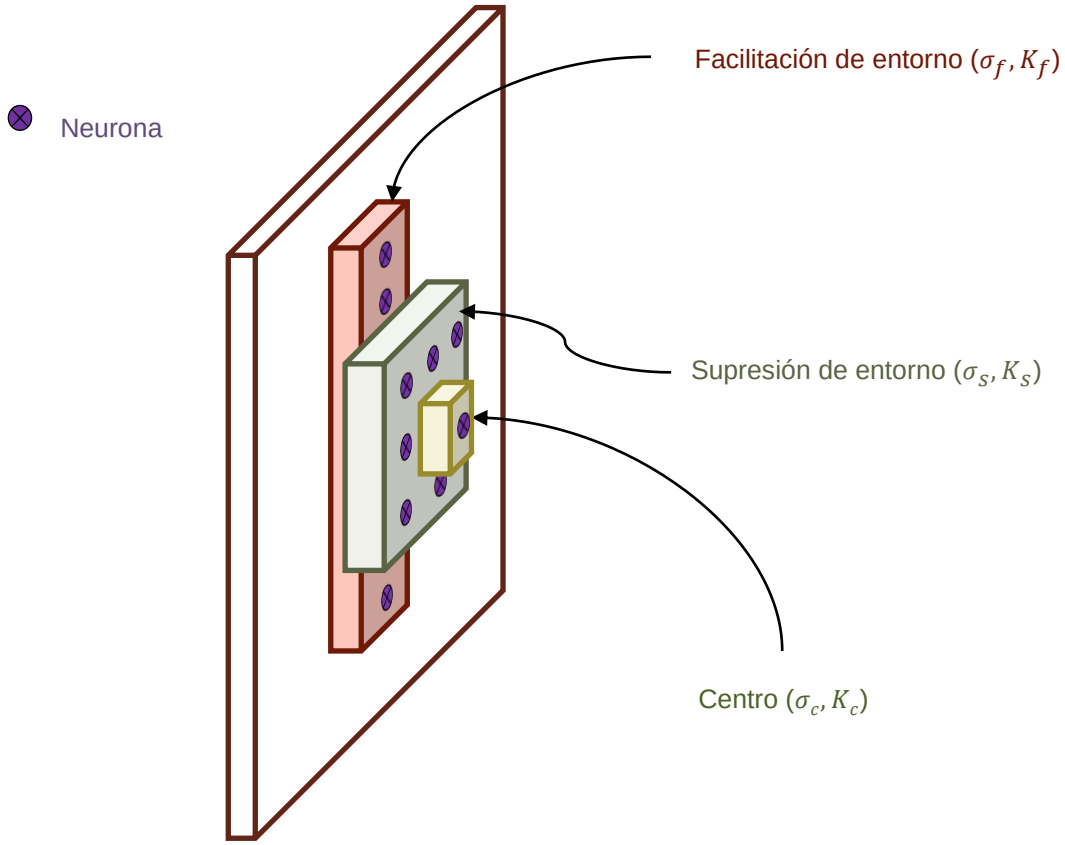


Figura IV-31: Esquema de un modelo de relación entre neuronas basada en el mecanismo de supresión y facilitación.

Con esta configuración, la salida del modelo DGF se presenta en la Figura IV-32.

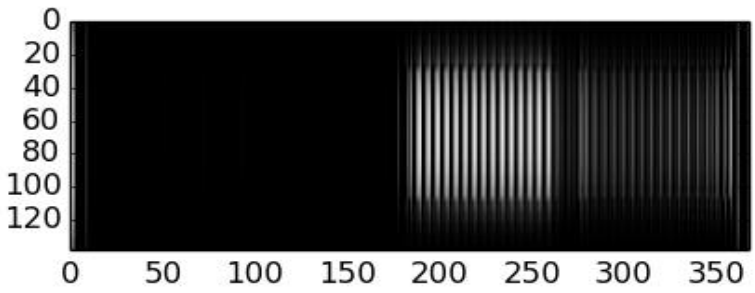


Figura IV-32: Respuesta ante una orientación del modelo DGF

El resultado obtenido sigue en línea con lo esperado. Por una parte, existe la parte supresora en el centro de las rejillas, más intensa en el cuarto estímulo, y por otra parte,

existe una señal de respuesta en los bordes de las rejillas, donde no hay señal, debido a la facilitación.

Las anteriores pruebas se han ejecutado usando un único filtro de Gabor en una orientación específica. Tal y como se ha hecho con el modelo SERRE para decidir las orientaciones de los bordes, se ha computado el valor máximo de todas las salidas para cada píxel. El resultado es el que aparece en la Figura IV-33.

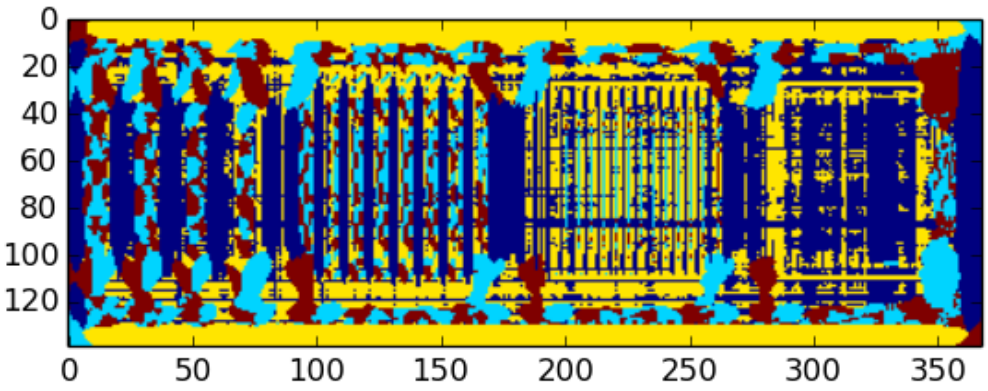


Figura IV-33: Respuesta del modelo DGF

El color azul oscuro indica 0º (barra vertical), el azul claro indica bordes a 45º, el amarillo a 90º y el granate indica la presencia de bordes a 135º. El resultado es mucho mejor que el modelo SERRE, con mayor resolución espacial y falsas detecciones, aunque posee más artefactos. Esto hace que se vea claramente que este tipo de modelos basados en restas y sumas de gaussianas tienen una gran potencialidad en el campo de la visión artificial.

IV.4.4 Diseño de los modelos RG1 y RGF

En este tercer caso, también nos vamos a apoyar en un modelo funcional para crear nuestros propios modelos de córtex, y es el ratio de gaussianas.

El concepto RoG fue propuesto por Cavanaugh et al. [CAVA02] y se basa en el mismo principio que el de diferencia de gaussianas: la existencia de un CRF y de una zona del entorno que le afecta de forma supresora, el ERF. La principal diferencia con respecto al modelo propuesto por Sceniak et al. es su configuración en base a la división de las influencias del centro y del entorno. Esta división parte del hecho de que de forma lineal

no es posible explicar ninguno de los fenómenos no lineales que suceden en el córtex, de ahí que busquen esta no linealidad. Así, el modelo que proponen tiene la siguiente forma:

$$R(x) = \frac{K_c L'_c(x)}{N + K_s L'_s(x)}$$

siendo K_c y K_s las ganancias de los términos centro y entorno, y L_c y L_s:

$$L'_c = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_c}\right)^2} dy \right)^2$$

$$L'_s = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_s}\right)^2} dy \right)^2$$

donde σ_s y σ_c representan el alcance espacial de ambas componentes.

La propuesta de modelo de córtex que hacemos se asemeja a nuestro modelo DG1: computar L_c mediante un filtro de Gabor y agregar la información de las células del entorno con ERF, para computar L_s. Su nombre será *modelo RG1* (*Ratio de Gaussianas 1*).

Se observa que el modelo es muy similar al DG1 en sus parámetros, por lo que para poder compararlo estableceremos los mismos parámetros de configuración en ambos modelos a lo largo de toda la tesis. Estos parámetros comunes se definirán más adelante (IV.4.5) en función de estudios neurológicos de los primates, y se corroborarán estadísticamente con una base de datos de imágenes.

El único parámetro que difiere en estos modelos es N, que es una constante de normalización necesaria para mejorar la no linealidad del modelo. A continuación, se ha preparado un simple test que comprobará dicha no linealidad: A una neurona con la configuración propuesta se le ha presentado un estímulo para dar su respuesta óptima. A este estímulo se le ha modificado el contraste y se ha mostrado el resultado. En la Figura IV-34 se ven diferentes gráficas de respuesta para diferentes constantes de normalización N.

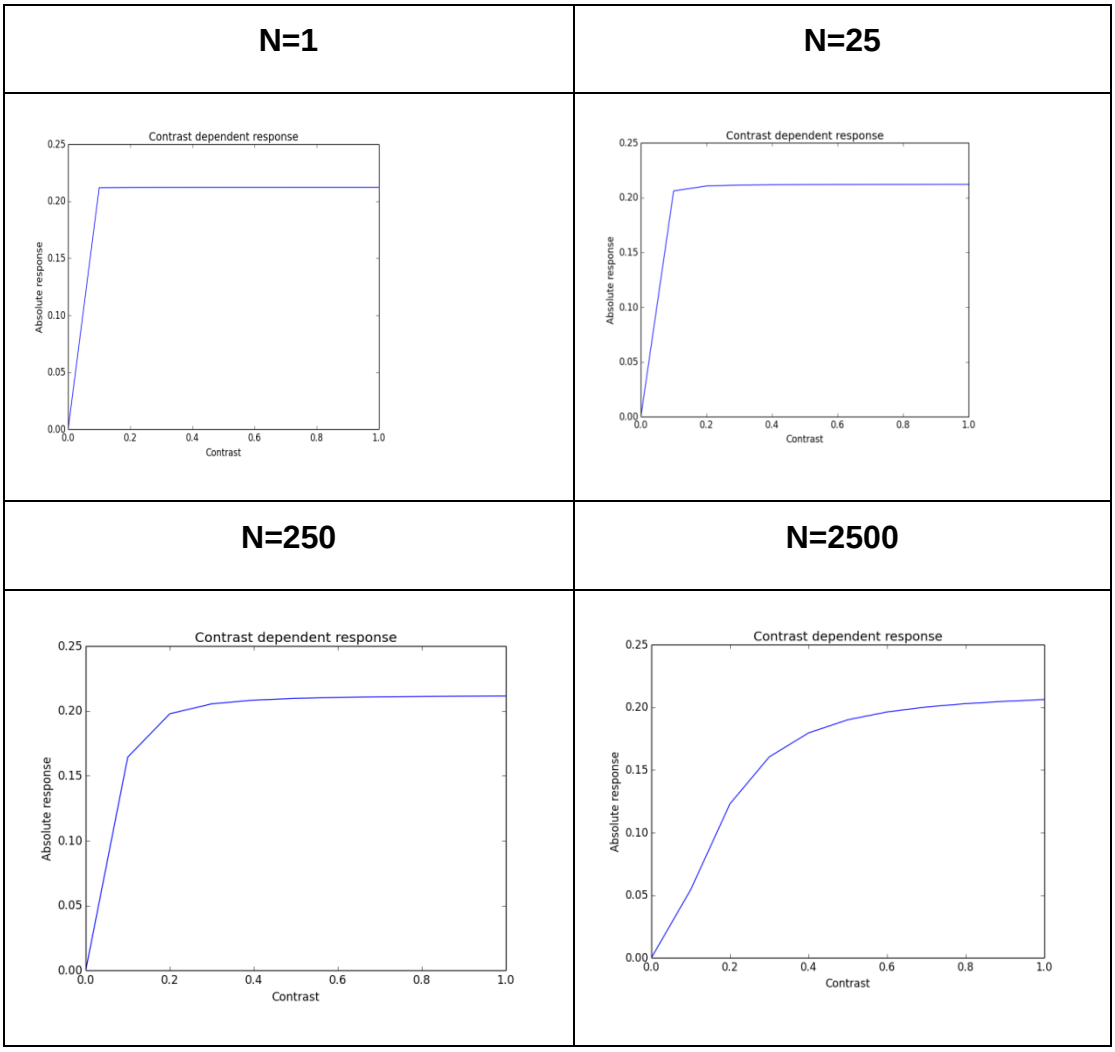


Figura IV-34: Respuestas del modelo no lineal para diferentes constantes de normalización

Queda patente que la mayor no linealidad se produce con valores de N altos, por lo que el valor por defecto escogido será N=2500, y posteriormente se realizará un ajuste fino en base a pruebas exhaustivas con una base de datos de imágenes.

Además del modelo RG1, también se va a realizar la propuesta de su modelo con facilitación, al que hemos denominado *RGF* (*Ratio de Gaussianas con Facilitación*). En el caso del modelo DGF, al ser un modelo lineal, la parte de facilitación era aditiva a la ecuación general. En este caso, al ser un modelo no lineal, la parte de facilitación puede estar tanto en el numerador como en el denominador, afectando no linealmente a la excitación o a la supresión. Se parte de la hipótesis de que este efecto es lineal con

respecto a otros, es decir, actúa como una suma, ya que es un efecto muy pequeño. Si actuase de forma multiplicativa generaría inestabilidades en la propia señal si la facilitación es fuerte. Para poder comprobar qué forma de añadir la facilitación es mejor, se usará un experimento psicofísico concreto. Este experimento parte de una imagen compuesta por estímulos alineados, pero no unidos, que está representada en la Figura IV-35.

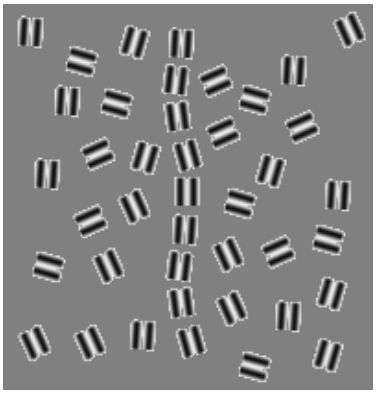


Figura IV-35: Estímulo de prueba para introducción de la facilitación no lineal

Debido al mecanismo de facilitación el cerebro es capaz de conectarlos y generar un contorno unido, y esto es lo que se buscará al introducir el término de facilitación. Antes de continuar, se muestran los resultados obtenidos sólo con el modelo de CRF y con el modelo de supresión en la Figura IV-36.

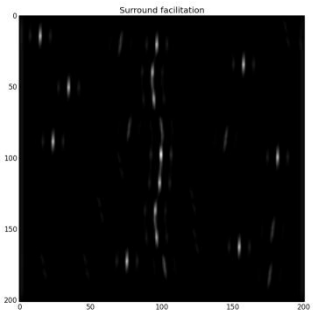
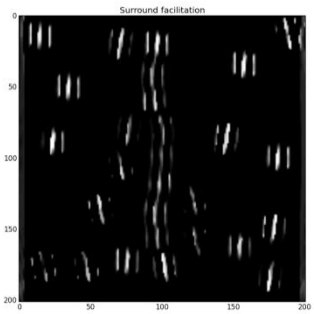
$R = \frac{Kc * {L_c}^2}{N}$	$R = \frac{Kc * {L_c}^2}{N + Ks * {L_s}^2}$
	

Figura IV-36: Respuesta perceptiva de un modelo clásico sin supresión (izquierda) y el modelo RG1 (derecha)

Es necesario mencionar que en la imagen de respuesta del modelo RG1 el contraste entre el blanco y el negro está aumentado para que sea posible discernir las respuestas. Como se ve en ambas imágenes, las muestras no están conectados entre sí.

A continuación, proponemos tres posibles configuraciones de la facilitación en el modelo RGF y sus respuestas. La primera configuración es la siguiente:

$$\frac{Kc * {L_c}^2}{N + Ks * {L_s}^2} + Kf * {L_f}^2$$

En esta configuración, la facilitación se introduce de forma lineal sobre la no linealidad de la supresión. Dependiendo del exponente de la señal de facilitación (no lineal), el efecto es mayor o menor tal y como se ve en la Figura IV-37.

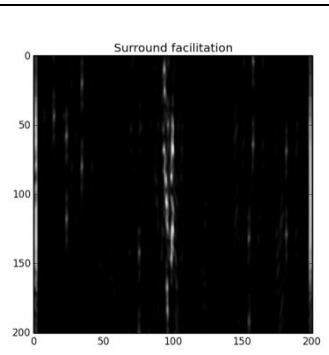
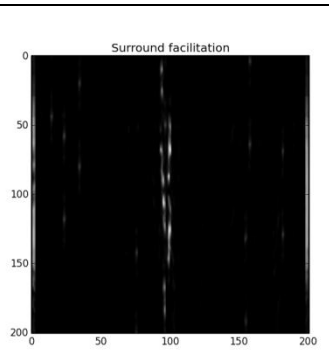
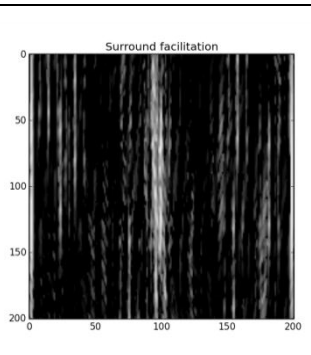
$L_f^{\,2}$	$L_f^{\,3}$	$L_f^{\,0.5}$
		

Figura IV-37: Respuesta del modelo RG1 modificado en función del exponente de la facilitación

Se ve que a menor exponente el efecto de facilitación es más pronunciado. De hecho, el exponente de 0,5 genera un exceso de facilitación que no es conveniente, ya que este efecto es muy sutil. A pesar de que parece un funcionamiento correcto y esperado, esta configuración no es válida por su relación con la supresión. Uno de los efectos que se produce con la supresión real de las células es que puede inhibir completamente la señal

de una neurona. Con esta configuración siempre estaría activa la parte de facilitación, por lo que sería incapaz de modelar este efecto.

Para evitar este problema, en la siguiente configuración se introduce la facilitación en el numerador, junto con la excitación principal, por lo que ahora la supresión también afecta a la facilitación.

$$\frac{Kc * {L_c}^2 + Kf * {L_f}^2}{N + Ks * {L_s}^2}$$

Al modificar el exponente de la facilitación, varía la respuesta, de forma similar a la configuración anterior. Variando los parámetros se comprueba que el valor idóneo es 1,5 , generando la respuesta de la Figura IV-38.

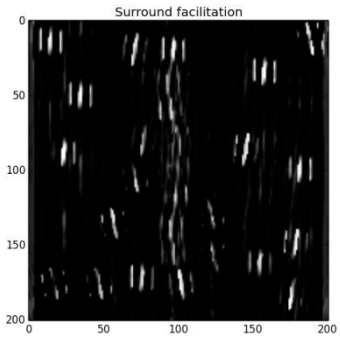


Figura IV-38: Respuesta en la que se ve la excitación y facilitación con $\boldsymbol{{L_f}^{1.5}}$

La última forma de introducir el efecto facilitador es como una resta al efecto supresor.

$$\frac{Kc * {L_c}^2}{N + Ks * {L_s}^2 - Kf * {L_f}^2}$$

Esta configuración no es estable en sí misma, ya que en ausencia de excitación central, no existirá respuesta alguna. Esto va en contra de los estudios que muestran que aunque

no exista una excitación central, puede generarse una señal de salida debida a la facilitación. Por tanto, el modo más adecuado de introducir la facilitación en el modelo RGF es como suma en el numerador.

Por último, este modelo RGF se puede configurar en forma de filtro único. Hasta ahora, los modelos parten del hecho de que se debe aplicar todo en dos fases: computar el CRF con los filtros de Gabor y calcular el efecto de supresión y facilitación del ERF. Para este modelo, por ser el más completo de todos, se ha planteado generar un único filtro que se compute en un único momento. En base a la formulación presentada para el modelo RGF, se puede ver el filtro generado en tres dimensiones en la Figura IV-39.

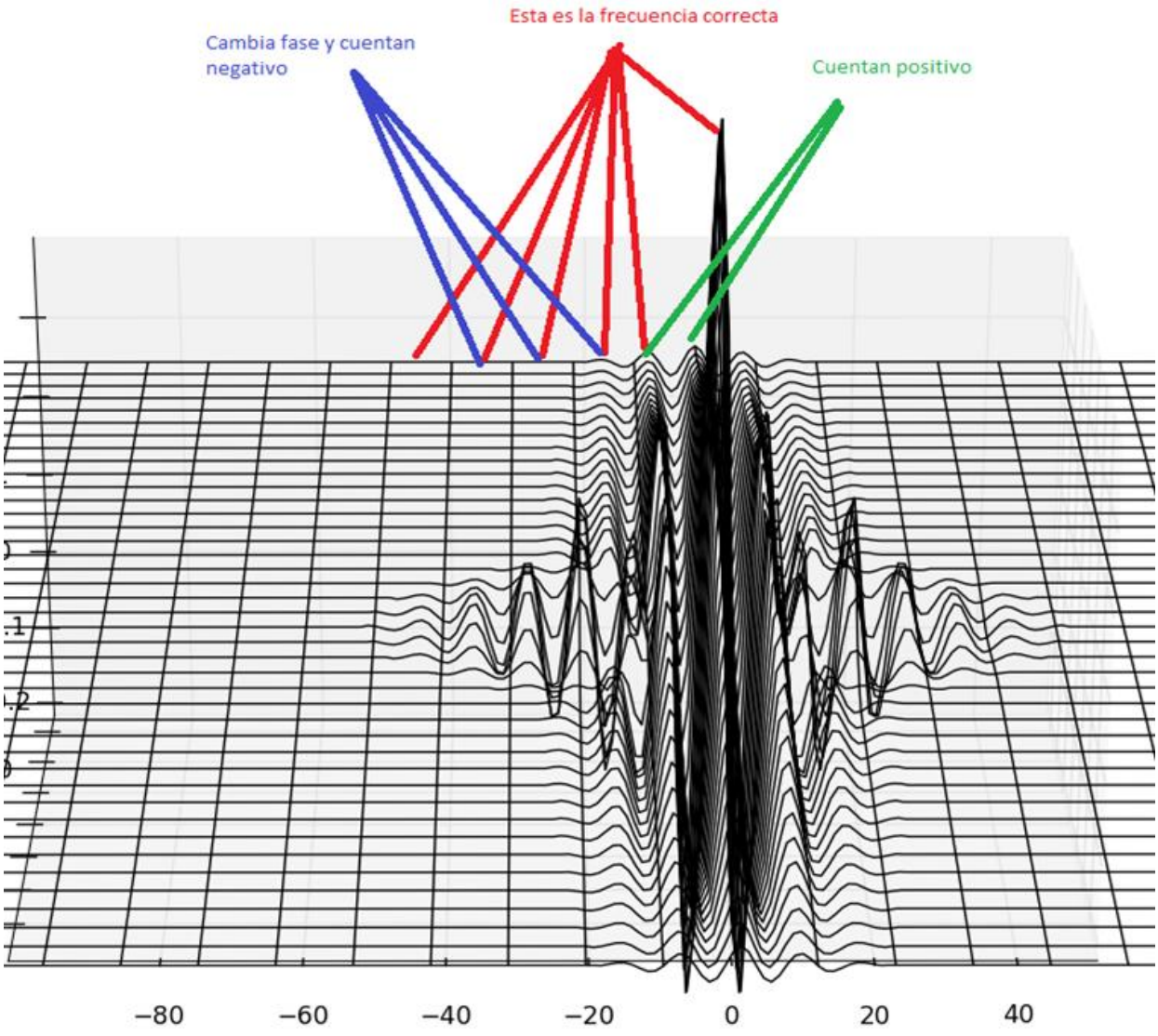


Figura IV-39: Filtro único del modelo RGF

Tal y como debe suceder, existe una mayor elongación en la dirección de orientación del filtro de Gabor, así como una zona circular en el centro como anti fase, ya que es la zona de supresión. El detalle se puede apreciar en el gráfico de 2 dimensiones de la Figura IV-40.

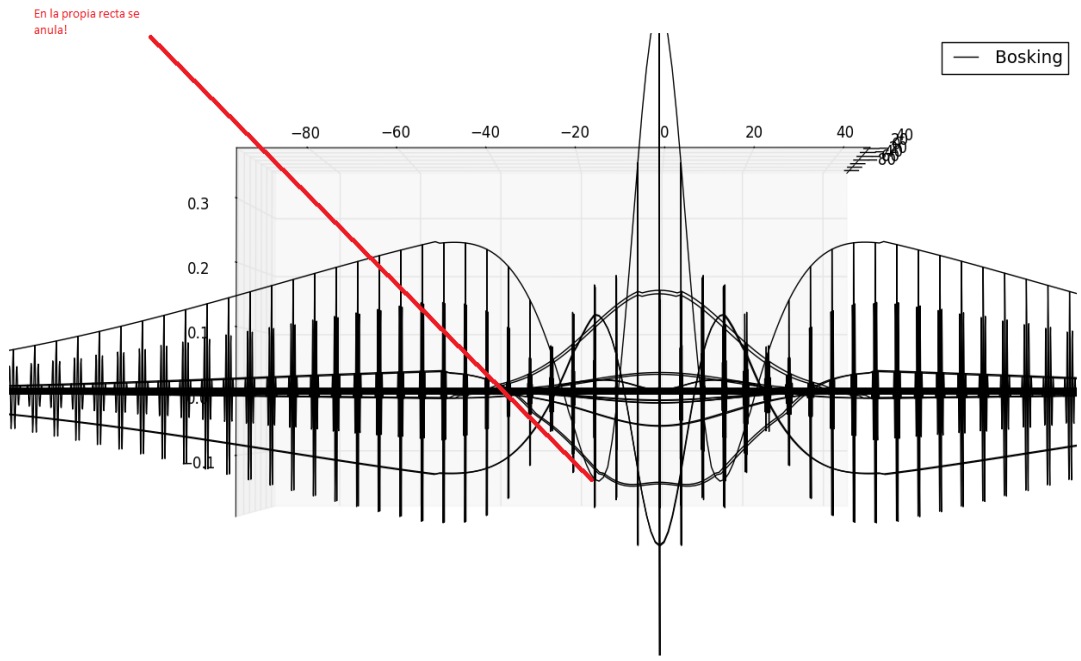


Figura IV-40: Filtro único del modelo RGF

Se ve que existe un cambio de fase en la zona central debido al mecanismo de supresión, pero en los lóbulos laterales está la facilitación y en la central la excitación. Este filtro se puede ver aplicado en la Figura IV-41.

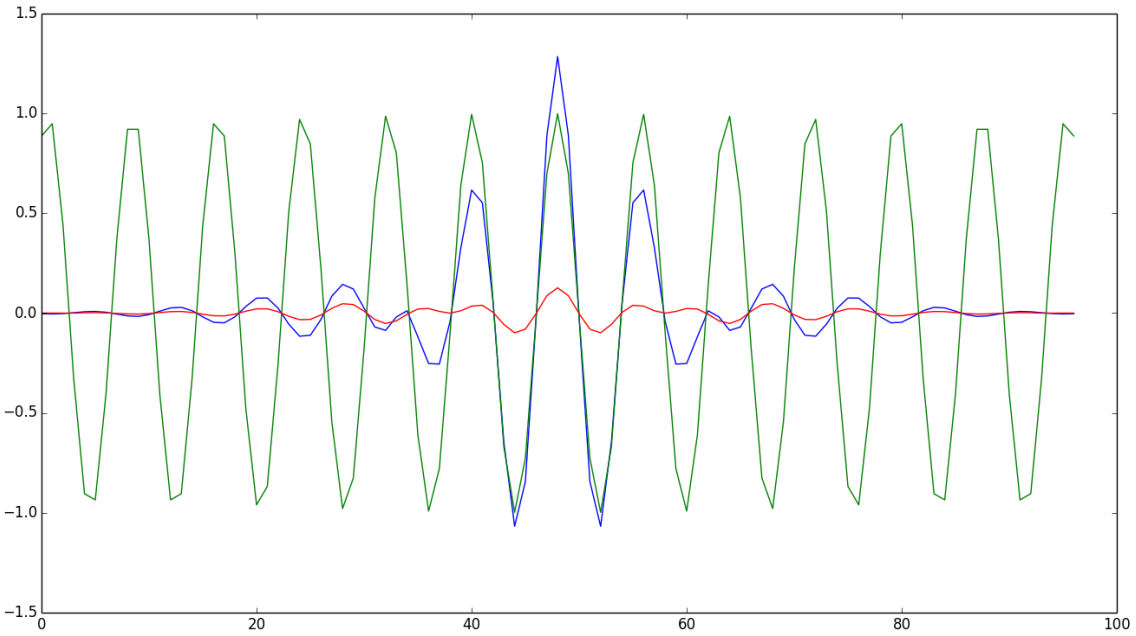


Figura IV-41: Aplicación del filtro propuesto. Verde: rejilla del estímulo, azul y roja: el filtro en la zona central y una zona alejada del centro (respectivamente)

En la Figura IV-41 se ve la rejilla de excitación en 1 dimensión (curva color verde). La frecuencia de la rejilla y su posición coincide con la configuración del filtro, lo que se comprueba en el corte en la coordenada *x=0* del filtro (curva azul). Se ve como los tres lóbulos centrales son los de la excitación primaria y afectan positivamente al resultado. A continuación, hay un cambio de fase en el filtro y pasa a actuar de forma negativa (mecanismo de supresión). Por último, vuelve a cambiar de fase y actúa de forma positiva (mecanismo de facilitación). En una zona más alejada del centro (curva roja) se puede observar este mismo comportamiento.

IV.4.5 Parámetros de configuración biológicos

Hasta ahora, se han descrito modelos de córtex visual propuestos en esta tesis. Se ha comprobado que existen varios parámetros necesarios para el correcto funcionamiento de los modelos. La definición de estos parámetros puede realizarse mediante dos aproximaciones diferenciadas: aprenderlos mediante técnicas de optimización y bases de datos de imágenes, o tomarlos del mundo biológico. Ya que se está definiendo un modelo biológicamente plausible, el paso lógico será utilizar parámetros biológicos, aunque posteriormente se refinarán en base a optimizaciones para que su funcionamiento en aplicaciones reales sea lo más óptimo posible.

Los modelos SERRE, DG1, DGF, RG1 y RGF poseen varios parámetros que se pueden dividir en tres grupos:

- **Campo excitador**: Es el conocido como campo receptivo clásico y se modela en base a filtros de Gabor.
- **Campo de supresión**: Sección del ERF y comprende el entorno que afecta de forma supresora a la respuesta de la neurona.
- **Campo de facilitación**: Segunda sección del ERF, que comprende la zona del entorno que modelan las conexiones horizontales y permite ayudar a la respuesta de la neurona.

Por parte del campo excitador o CRF, existen varios parámetros necesarios en los filtros de Gabor. Los parámetros extraídos de la literatura en referencia a la capa V1 del córtex visual de los macacos se muestran en la Tabla IV-2.

Tabla IV-2 Parámetros de configuración de los filtros de Gabor que modelan el CRF [SERR07]

Banda Σ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Tamaño del filtro s	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
σ	28 & 3.6	4.5 & 5.4	6.3 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.2 & 22.8

En cuanto a los parámetros de configuración del ERF, se ha realizado un ejercicio de análisis y comparación de la literatura de este campo.

En el caso del campo supresor, son dos parámetros principales los necesarios: K_s y σ_s. El primer valor indica el peso de la supresión en el modelo, y el segundo referencia la extensión de la gaussiana que modela esta supresión. En la medición de este último parámetro existen muchas investigaciones que han dado con números semejantes. Lo primero que hay que recalcar es que este parámetro varía con el contraste del estímulo [CAVA02]. Para simplificar el modelo en esta tesis se considerará que su valor no varía. En [CAVA02] se indica que el valor de la extensión del campo excitador es de media 1,4º, en coordenadas del espacio cortical, donde para la musaraña arbórea se pueden transformar como 4,8°/mm en vertical y 5,5°/mm en horizontal [CHIS03], mientras que el ancho del campo supresor es de media 2,7º. Este valor varía en función del punto de análisis en la retina y el córtex, pero el ratio entre el campo supresor y el campo excitador se mantiene constante y tiene un valor medio de 2,5. Esto también corrobora lo visto por otros autores como [SCEN01], donde se afirma que este ratio es de 2,1 +-0,1, valor cercano al anterior. Como conocemos el valor del campo excitador σ_c (ver Tabla IV-2), podemos usar esa relación para obtener el campo surpesor σ_s.

En cuanto al coeficiente K_s, en [CAVAN02] se mide como el índice de supresión. Este índice mide la fracción de señal que se ha reducido con respecto a su óptimo. De media, los autores de este trabajo indican que se produce una disminución del 38% de la señal cuando la excitación pasa al campo supresor. Aunque matemáticamente no es cierto, se considerará la influencia de la supresión K_s como 0,38. Se indica que el valor no es cierto, porque ese 0,38 indica cuánta señal menos existe en total, incluyendo la excitación, pero

aquí se está modelando sólo la influencia de la supresión. Como disminuye un 38% se considera que la influencia de esta señal es de un 38%. Esta simplificación es necesaria ya que este valor se usará en diferentes modelos, y es necesario tener el mismo componente en ellos para poder comparar el modelo y no la optimización de sus parámetros.

En el caso del campo facilitador se pueden diferenciar dos regiones en el córtex, que son las que generan la facilitación lejana y cercana [ICHI07]. La cercana posee una fuerza de facilitación mediana del 51,2% mientras que la facilitación lejana es más débil y sólo posee una fuerza del 28,7% [ICHI07]. Como se lleva describiendo a lo largo de esta tesis, en el modelo que proponemos sólo se tendrán en cuenta las conexiones horizontales lejanas para la facilitación de una respuesta, por lo que sólo se tendrá en cuenta la facilitación lejana. En concreto, el valor anterior se asignará a K_f, ya que del mismo modo que sucedía con K_s, es necesario tener un valor único que pueda ser utilizado por todos los modelos que se han propuesto. Si no, sería necesario recalcular todos los parámetros para todos los modelos en función del propio modelo y de estos valores. Estos cálculos generarían interdependencias entre parámetros, que sí modelarían de forma real el córtex, pero que pueden ser extremadamente complejos para un modelo a usar en el campo de la visión artificial. Así, se considera K_s como 0.287.

Finalmente, sólo falta por obtener el valor de la extensión del campo de facilitación σ_f. En [ICHI07] se indica que el valor medio de los radios de los campos de supresión y facilitación es de 8,8º ± 0,63º and 4,4º ± 0,4°, midiendo desde el pico de la respuesta. Esto genera un ratio medio de 2,0 , entre la extensión de la supresión y la facilitación. Es por ello por lo que se usará este valor para calcular σ_f, ya que se dispone del valor de σ_s.

Tanto en la supresión como en la facilitación se han obtenido ratios entre diferentes extensiones de los campos. Esto posee una ventaja clara, y es que el campo excitador central está definido por diferentes filtros de Gabor con diferentes escalas y amplitudes. Gracias a los ratios anteriores no es necesario identificar cada amplitud de supresión y facilitación, sino que al tener la misma relación para cualquier escala, éstos valores de σ_s y σ_f se pueden computar para cada una de ellas.

El resumen de los parámetros se presenta en la Tabla IV-3.

Tabla IV-3: Resumen de los parámetros de configuración biológicos

Center	K_c	1
	σ_c	*Tabla IV-2*
Surround – Supression	K_s	0.38
	σ_s	2.5 * σ_c
Surround – Facilitation	K_f	0.287
	σ_f	2.0 * σ_s

IV.5 Efectos del córtex visual primario y respuestas neuronales simples

En los apartados anteriores se han propuesto una serie de modelos neuronales que pueden componer el córtex visual primario. Se han expuesto varios prototipos de modelos, y en ellos se ha comprobado la existencia de numerosos parámetros de configuración que son necesarios. Estos parámetros se han obtenido en base a los trabajos de investigación en el campo de la neurociencia, y con ellos ya es posible generar un modelo de V1 biológicamente plausible.

En este apartado se va a comprobar si su funcionamiento es similar a las neuronas reales, de tal forma que se pueda seleccionar un único modelo de todos los propuestos para aplicar en el campo de la visión artificial.

Para realizar esta selección se van a elegir una serie de efectos producidos en las neuronas reales que son interesantes desde el punto de vista del procesamiento de imagen. Estos efectos se han obtenido de la literatura y son los que se describen y estudian en la mayoría de los artículos científicos. Cada efecto lleva asociado a él un comportamiento de la respuesta neuronal en función de un estímulo dado. En el apartado siguiente se mostrarán los efectos seleccionados, así como una breve descripción de los mismos.

IV.5.1 Definición de efectos a modelar

Preferencia ante una orientación. Ante un estímulo orientado en la misma dirección que la neurona, la respuesta de dicha neurona será máxima. En cambio, si se modifica la

orientación del estímulo, la respuesta de la neurona disminuirá. El mínimo viene marcado por una tasa de “disparo” espontáneo de las neuronas cerebrales.

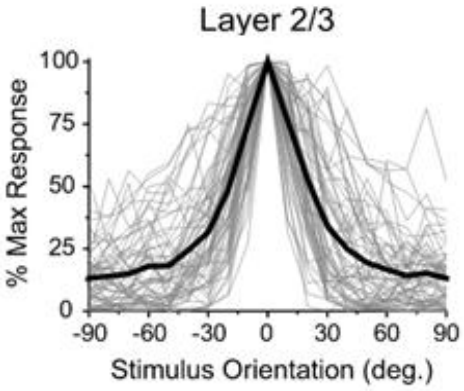


Figura IV-42: Efecto en neuronas reales [CHIS03]

Supresión del entorno. La existencia de un centro y un entorno se evidencia en este efecto. Partiendo desde el centro del campo receptivo, a medida que aumenta el diámetro del estímulo, la respuesta de la neurona aumenta. Llegado a un cierto límite, la respuesta disminuye al aumentar el tamaño, como se ve en la Figura IV-43.

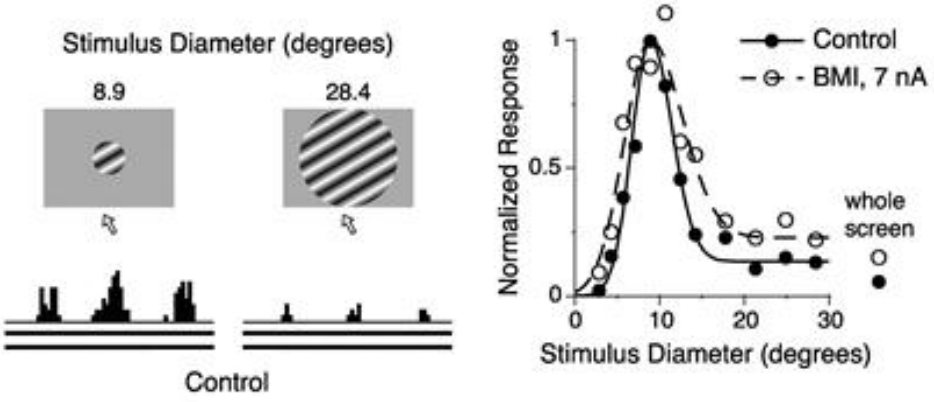


Figura IV-43: Efecto en neuronas reales [ICHI07]

Facilitación ortogonal por supresión. Este efecto está derivado de los dos anteriores: una neurona se excita ante una orientación concreta, y a la vez, si la excitación es muy amplia su respuesta se suprime. Si esa excitación modifica su orientación, entonces la señal deja de suprimirse, y se puede ver el fenómeno como una facilitación por orientación cruzada.

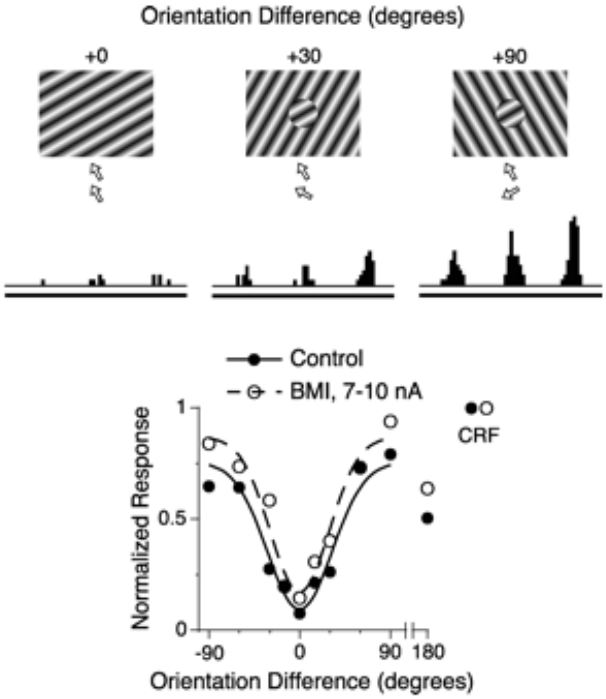


Figura IV-44: Efecto en neuronas reales [OZEK04]

Facilitación de conexiones horizontales. Este efecto es debido a la existencia de conexiones horizontales largas. Partiendo de un estímulo del CRF, así como de una corona de tamaño pequeño, si se aumenta el radio interior de esta corona se comprueba que la respuesta aumenta. Este aumento finaliza en un punto en el que la respuesta disminuye debido al efecto supresor. Como nota, cabe destacar que se ha comprobado que este efecto es dependiente del contraste (tres gráficas diferentes en la Figura IV-45, en la que cada una se relaciona con un contraste). Tal y como se ha mencionado en otras ocasiones con el objetivo de simplificar el modelo, este efecto no se tendrá en cuenta en los modelos propuestos (sólo se considerará la gráfica roja o gris del siguiente dibujo).

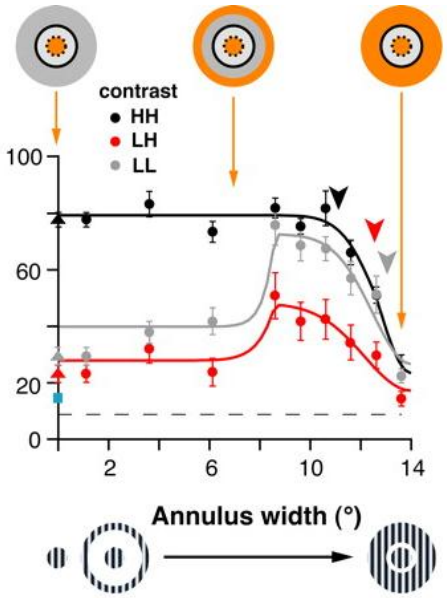


Figura IV-45: Efecto en neuronas reales [ICHI07]

Conexiones co-orientadas. Este efecto está relacionado con las conexiones entre neuronas y la facilitación. Como se ve en la Figura IV-46, cuando se presentan estímulos discontinuos en la misma orientación preferida por la neurona, la respuesta aumenta con el tiempo. Por el contrario, si los estímulos no poseen la misma orientación, la respuesta no varía. Este efecto se modela directamente al programar la neurona y sus conexiones, por lo que no se evaluará, pero sí queda constancia de su existencia.

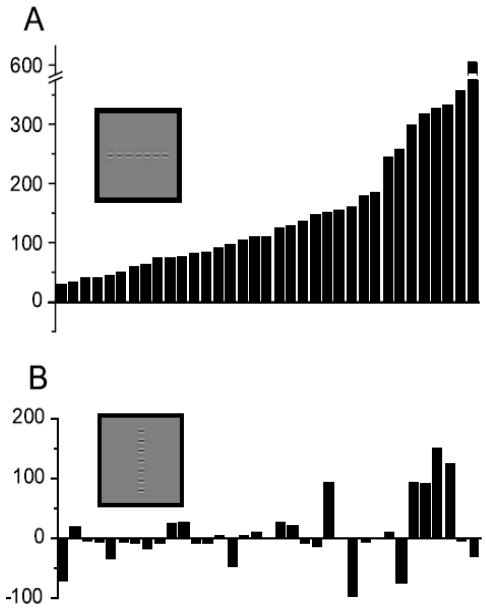


Figura IV-46: Efecto en neuronas reales [CHIS03]

Variación de la respuesta con el contraste. Un efecto que se ha visto en diferentes estudios es la variación de otras respuestas con el contraste. Por ello, se tendrá en cuenta el efecto aislado de que la respuesta de una neurona varía de forma no lineal con el contraste de la excitación, como se ilustra en la Figura IV-47.

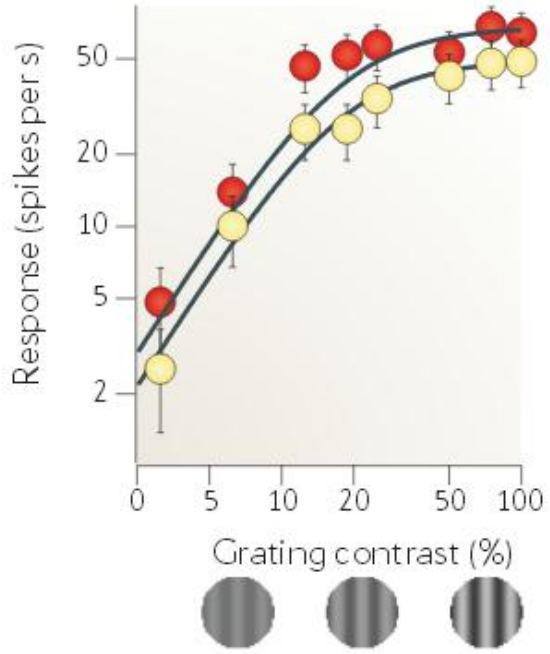


Figura IV-47: Efecto en neuronas reales [CARA12b]

Detección de contorno. Este efecto es derivado de experimentos psicofísicos y no tanto neuronales. El efecto se da cuando se presenta un conjunto de estímulos coalineados pero no conectados. En este momento se produce una respuesta más intensa en ellos, de tal forma que se cierran contornos abiertos y alineados entre sí.

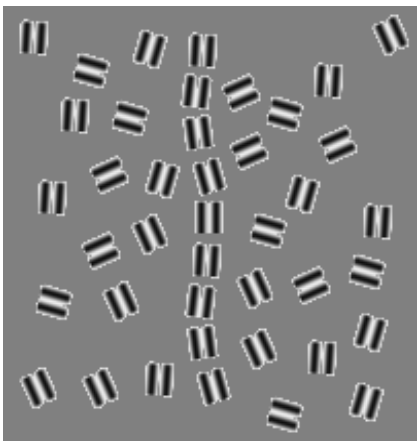


Figura IV-48: Patrón del experimento [SERI04]

IV.5.2 Modelos de córtex y respuesta neuronal simple

Una vez que se han seleccionado los efectos más interesantes desde el punto de vista de análisis de imagen, se han mostrado las respuestas reales de las neuronas y se han definido los modelos a utilizar, es el momento de estimular a las neuronas artificiales basadas en dichos modelos con los estímulos que se acaban de proponer en el apartado IV.5.1.

Los modelos que se van a probar son los definidos que hemos desarrollado en el punto IV.4 de esta tesis, y se resumen de la siguiente forma:

- **Campo receptivo clásico (CRF)**: En este modelo, la neurona está modelada como un filtro de Gabor de una orientación y frecuencia determinada. Su salida será la convolución de este filtro de Gabor con la excitación en su punto central.
- **Modelo SERRE:** En este modelo se posee un CRF como el descrito anteriormente y una segunda capa de max-pooling (o winner-take-all) para seleccionar el valor de salida final simulando parcialmente la supresión y la facilitación del ERF.
- **Modelo DG1**: En este modelo, además del CRF, se modela el efecto supresor del ERF como una diferencia de gaussianas. Para obtener la salida, se consideran varias neuronas, se calculan sus CRF y, posteriormente, se ejecuta la diferencia de gaussianas sobre la neurona central del estímulo.
- **Modelo DGF**: Similar al anterior, también se tiene en cuenta la facilitación en forma de suma.
- **Modelo RG1**: En este caso, el efecto de la supresión se modela de forma no lineal como una división.
- **Modelo RGF**: Similar al modelo RG1 donde la facilitación se introduce como una suma en el numerador, permitiendo mantener la no linealidad del modelo.
- **Filtro basado en el modelo RGF**: Finalmente también se ha generado un filtro único que evite tener que ejecutar en dos fases el cálculo de la respuesta de las neuronas.

Una vez recordados los modelos propuestos en esta tesis, vamos a obtener sus respuestas ante diferentes estímulos con el objetivo de seleccionar el modelo que más se ajuste a la realidad. Para ello, hemos ejecutado varias pruebas sobre nuestros modelos

neuronales utilizando los efectos vistos anteriormente. Tras obtener las respuestas, se va a proceder a cuantificar numéricamente los resultados obtenidos.

En primer lugar, para el efecto de **preferencia ante una orientación**, se va a mostrar un estímulo de la misma frecuencia que el filtro de Gabor que modela el CRF de todos los modelos y se va a rotar 180º. Las respuestas de los diferentes modelos se muestran en la Figura IV-49.

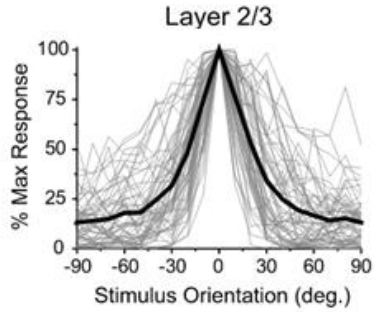
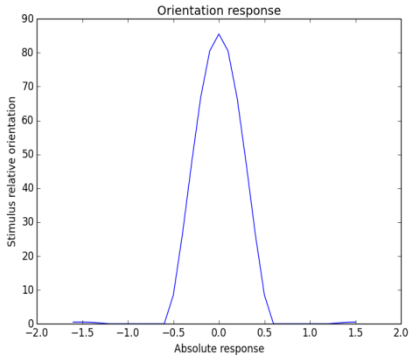
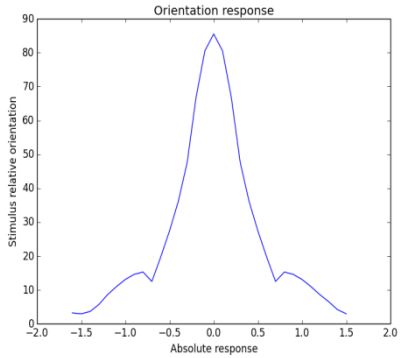
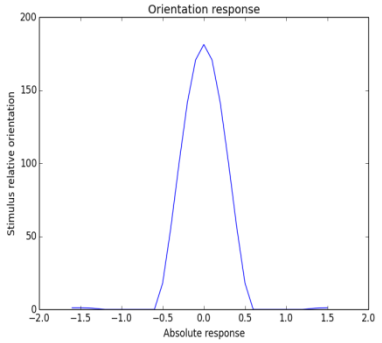
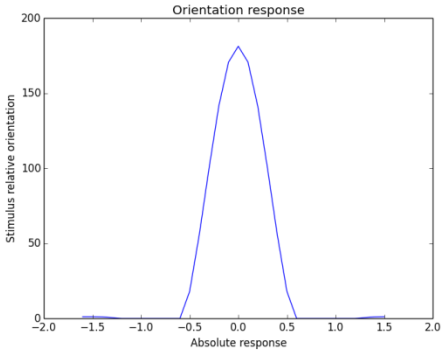
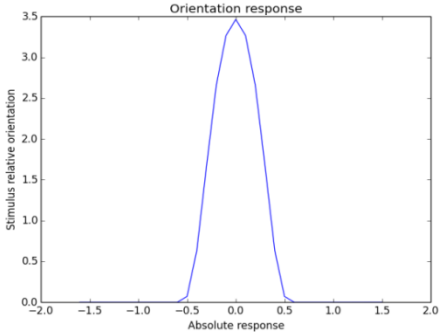
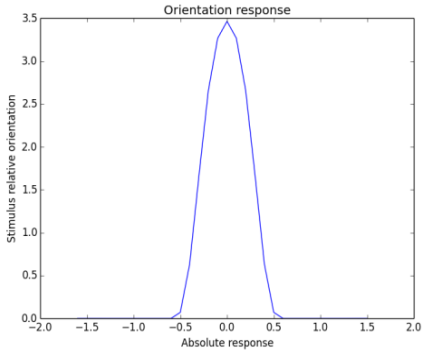
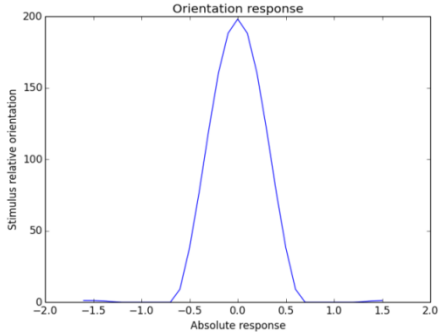
Respuesta real	CRF	Serre	Sceniak
			
Bosking lineal	**Cavanaugh**	**Bosking normalizado**	**Filtro Bosking**
			

Figura IV-49: Respuesta de los modelos de córtex propuestos para corroborar la preferencia ante una orientación

Ante este estímulo, se ve que todos los modelos propuestos generan una señal que es bastante similar a la real. Para cuantificar esta similitud, se realizará un análisis detallado al final de este apartado. Aun así, la mayor diferencia es la ausencia de respuesta más allá de los 30º [VALO82]. Esto es debido a que no se ha modelado la tasa de activación espontánea que sí tienen las neuronas reales, pero que no es necesaria para un modelo de visión artificial. Por otro lado, la especificidad se cumple de igual forma, es decir, a partir de 30º (o 0.5 radianes) la respuesta es despreciable.

El segundo de los efectos que se analiza es la **supresión del entorno**. En este caso, el resultado obtenido para los modelos propuestos se ve en la Figura IV-50.

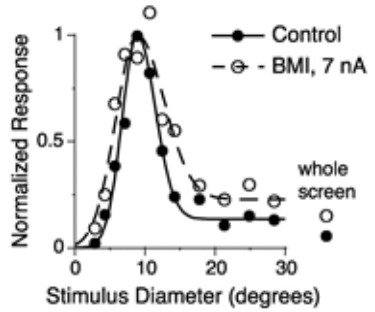
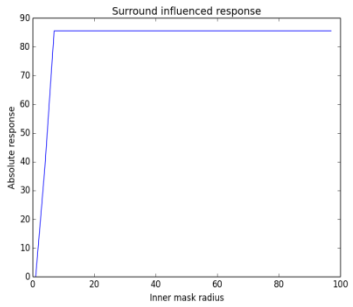
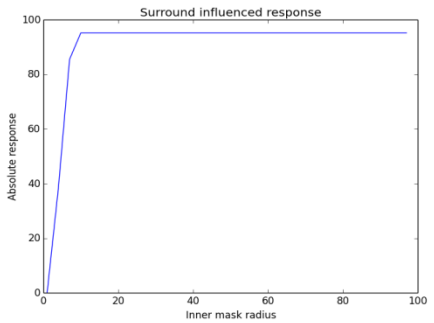
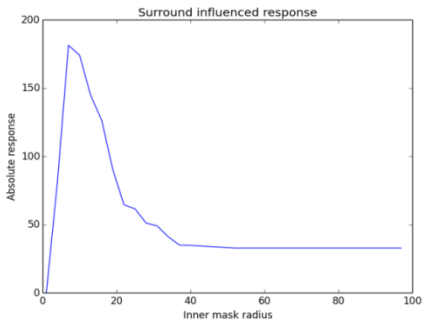
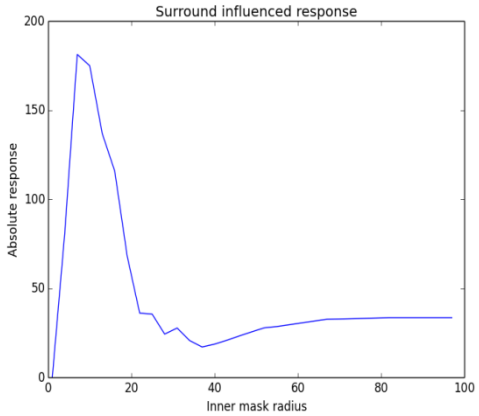
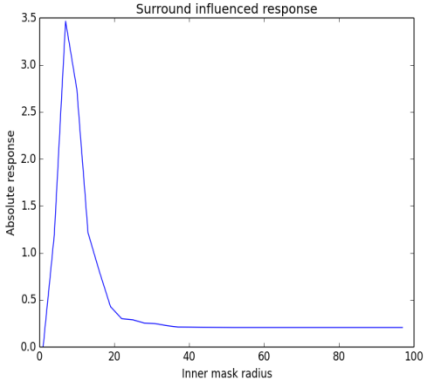
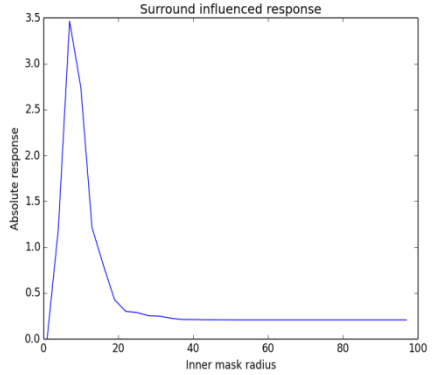
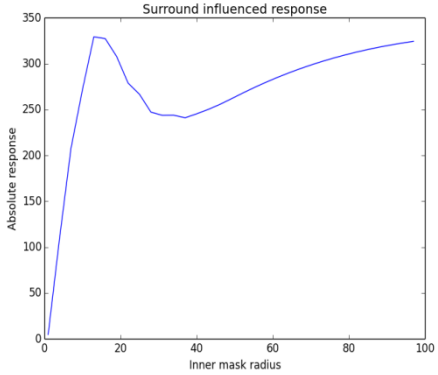
Respuesta real	CRF	Serre	Sceniak
			
Bosking lineal	**Cavanaugh**	**Bosking normalizado**	**Filtro Bosking**
			

Figura IV-50: Respuesta de los modelos propuestos para corroborar la supresión del entorno

Con este efecto de supresión se comienzan a ver las diferencias entre los modelos. Por un lado, se ve cómo al aumentar el tamaño del estímulo, todos los modelos aumentan su respuesta. Si bien este aumento se detiene en un punto y comienza a disminuir debido al efecto del entorno, en el modelo clásico de campo receptivo (*CRF*) no sucede eso. Esto es lógico, ya que este modelo no tiene ningún tipo de conexión con su entorno. Por otro lado, el modelo SERRE, tampoco genera esta supresión y en este caso sí tiene conexiones con las neuronas del entorno.

Además de estos efectos de supresión, se ve cómo aparecen efectos de facilitación en aquellos modelos que lo incluyen (DGF y RGF). A diferencia de ellos dicho efecto no aparece en la gráfica real. Esto puede ser posible porque en muchos estudios indican que son las mismas neuronas las que cambian de un efecto supresor a uno facilitador [ICHI07], y en el trabajo que se toma como referencia sólo han excitado el córtex para ver el efecto de supresión.

Cabe destacar el efecto facilitador del *filtro basado en RGF*. Se ve cómo en la cola de la respuesta, se compensa completamente el efecto de supresión, por lo que es necesario comprobar si realmente es así o si el filtro no posee los parámetros adecuados.

El siguiente efecto a revisar es la unión de los dos anteriores en la denominada **facilitación ortogonal por supresión.** En este caso, cuando se varía la orientación del estímulo en el ERF se debe disminuir la supresión. El resultado obtenido por los diferentes modelos propuestos se ve en la Figura IV-51.

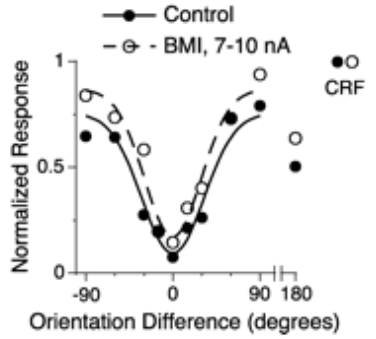
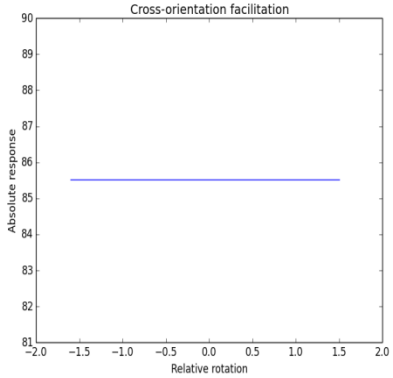
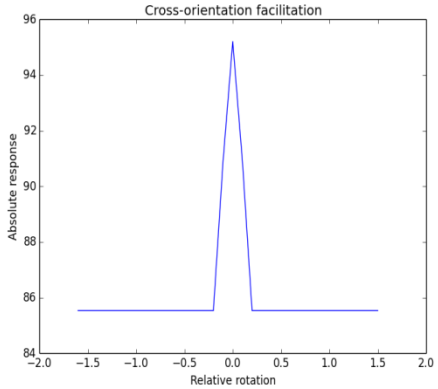
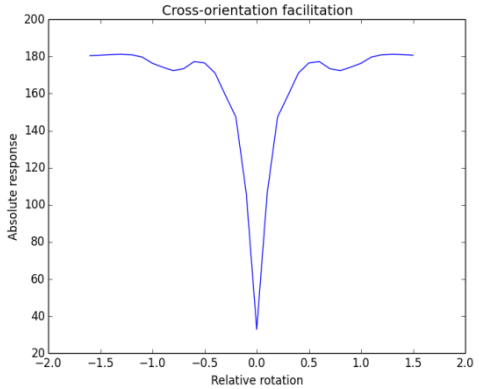
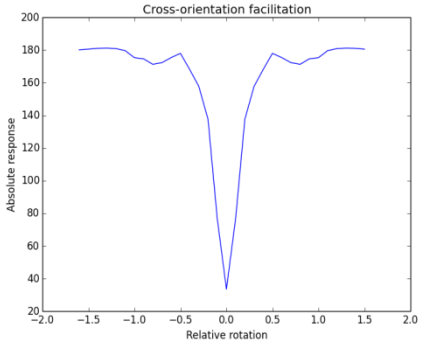
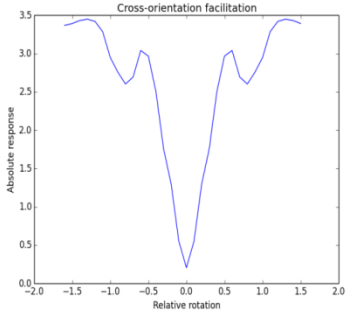
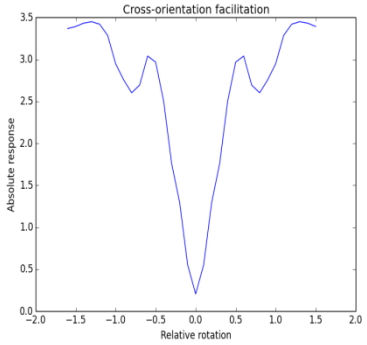
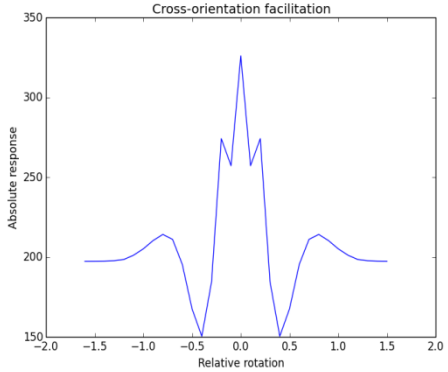
Respuesta real	CRF	Serre	Sceniak
			
Bosking lineal	**Cavanaugh**	**Bosking normalizado**	**Filtro Bosking**
			

Figura IV-51: Respuesta de los modelos propuestos para corroborar la facilitación ortogonal por supresión

En este caso, también se pueden ver grandes diferencias entre los dos primeros modelos y el resto. Tanto el modelo *CRF* como el modelo *SERRE*, no presentan ningún tipo de facilitación ni supresión. De hecho, el modelo *SERRE* presenta un pico, pero es mínimo con sólo un 10% de su amplitud. Por otra parte, el *filtro basado en RGF* tampoco funciona como debería. Se ve que está más excitado en el centro, por lo que es síntoma de que no está suprimiendo la señal correctamente. Analizando en detalle el filtro, esto se debe a que la supresión actúa como una suma desfasada, no como una resta. Esto genera que realmente no se reste y que cuando se mueva ligeramente la frecuencia (por cambios en la orientación) la supresión actúe de forma no esperada. Esto muestra que el *filtro* no parece un modelo válido.

El siguiente efecto a mostrar es la **facilitación por conexiones horizontales**. Para ello, se ha utilizado una excitación central y un anillo que aumenta su radio interior hacia el centro de la imagen. Los resultados obtenidos por los diferentes modelos se muestran en la Figura IV-52.

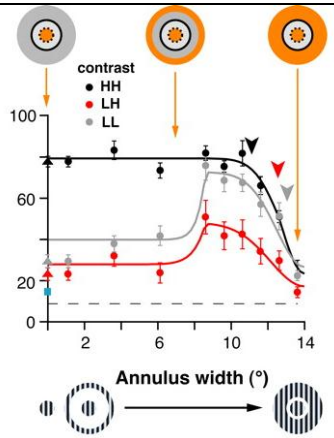
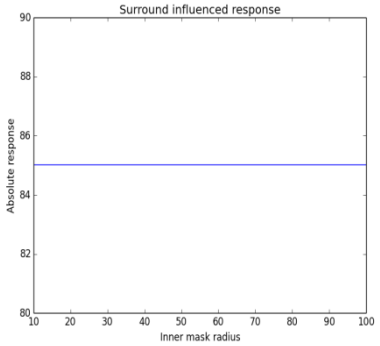
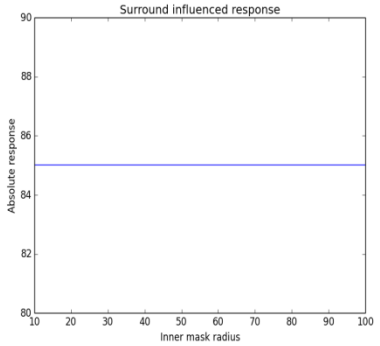
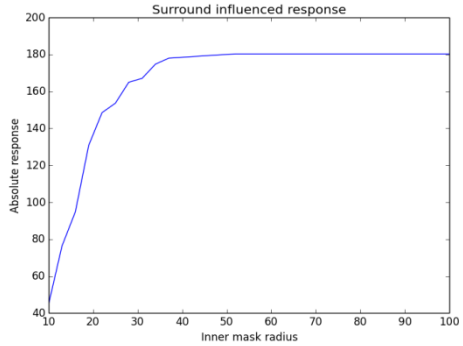
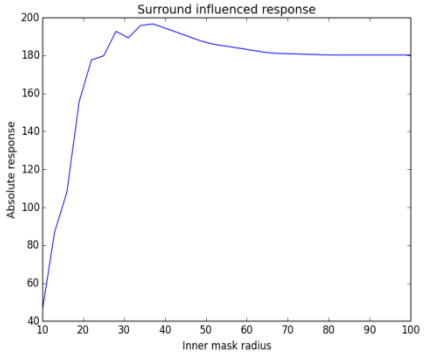
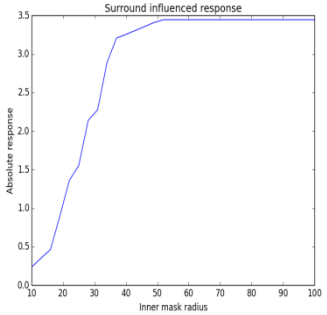
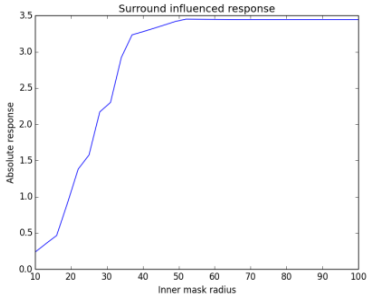
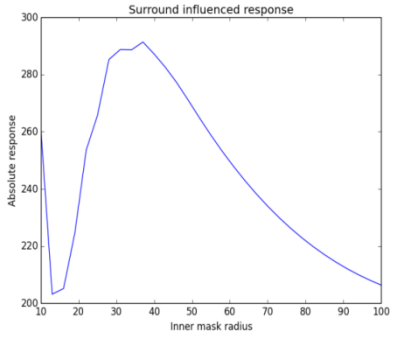
Respuesta real	CRF	Serre	Sceniak
			
Bosking lineal	**Cavanaugh**	**Bosking normalizado**	**Filtro Bosking**
			

Figura IV-52: Respuesta de los modelos propuestos para corroborar la facilitación por conexiones horizontales

Como era de esperar, sólo está presente la facilitación en aquellos modelos en los que explícitamente se ha incluido un término para ello (los modelos *DGF y RGF*). Es necesario resaltar que el resultado esperado es similar a la gráfica roja, pero en modo espejo. El modelo que genera una gráfica más alineada con la señal esperada es el *modelo DGF*. A su vez, el *filtro basado en RGF* posee demasiada facilitación, generando situaciones de recuperación total de la señal suprimida. Por el contrario, el *modelo RGF* posee una ligera facilitación, mostrada en un tamaño de anillo de 50 píxeles y un aumento ligero de la señal.

El siguiente fenómeno a tratar es la **variación no lineal de la respuesta con el contraste** del estímulo. Para ello, se ha presentado una rejilla orientada en el ángulo de preferencia y en su frecuencia dentro del CRF. Para generar la gráfica se ha modificado el contraste entre la zona más clara y la más oscura, generando las respuestas mostradas en la Figura IV-53.

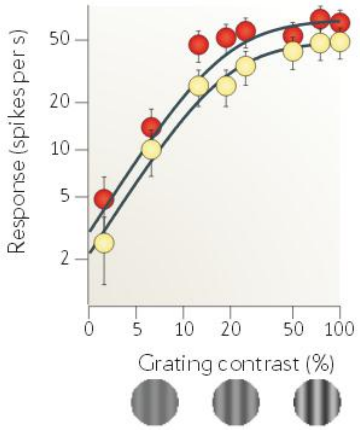
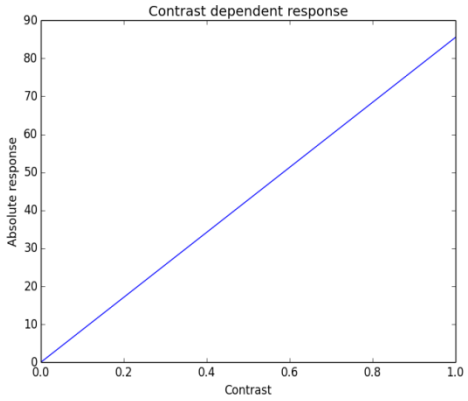
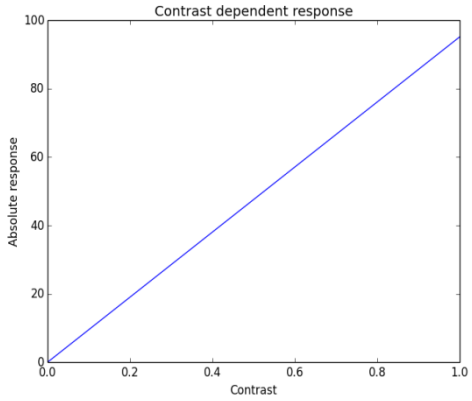
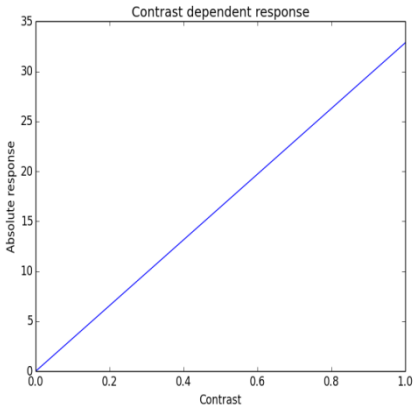
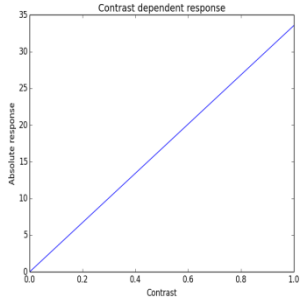
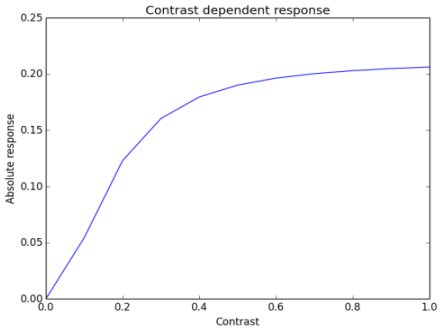
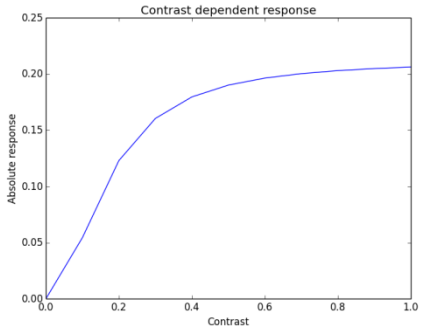
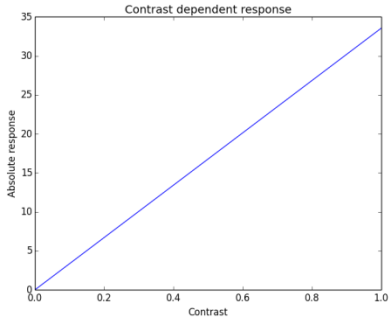
Respuesta real	CRF	Serre	Sceniak
			
Bosking lineal	**Cavanaugh**	**Bosking normalizado**	**Filtro Bosking**
			

Figura IV-53: Respuesta de los modelos propuestos para corroborar la variación no lineal de la respuesta con el contraste

Se comprueba cómo la mayoría de los modelos generan una respuesta totalmente lineal con la variación del contraste. Los únicos que no son lineales son los modelos basados en RoG. En cambio el *filtro basado en RGF*, basado en RoG, es lineal, por lo que se vuelve a comprobar su no funcionamiento.

Por último, se verá el resultado del experimento sobre el **cierre de contornos**. Para este caso no se probará con el *filtro basado en RGF*, y tampoco se mostrará señal real ya que no se dispone de ella. Las respuestas obtenidas se muestran en la Figura IV-54.

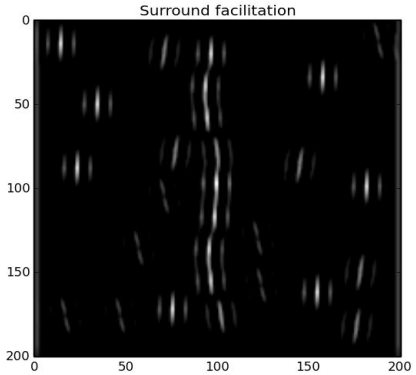
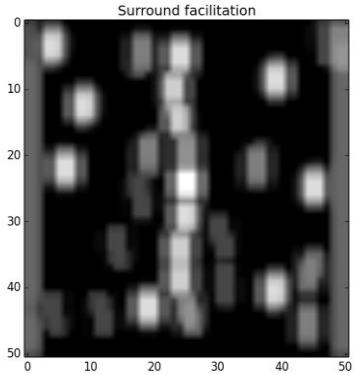
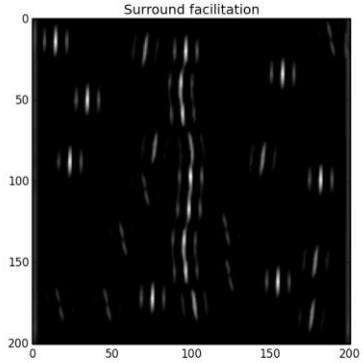
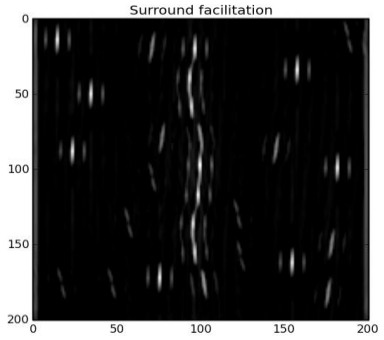
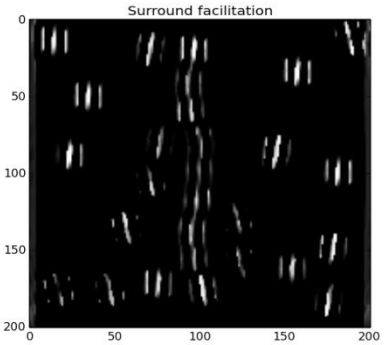
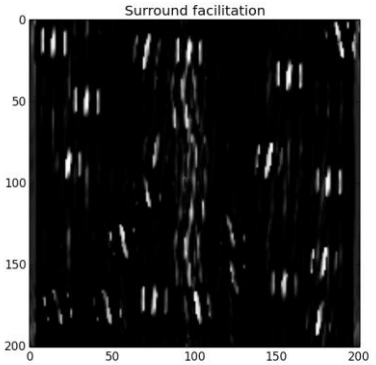
CRF	Serre	Sceniak
		
Bosking lineal	**Cavanaugh**	**Bosking normalizado**
		

Figura IV-54: Respuesta de los modelos propuestos para corroborar el cierre de contornos

El objetivo de esta prueba es comprobar que, con la facilitación de las neuronas, las líneas centrales brillen más que los otros elementos. Visualmente, el resultado del *modelo SERRE* es muy alentador, mientras que otros no hacen sino imitar el resultado básico del *modelo CRF* (aunque eliminando ciertas conexiones debido al mecanismo de supresión). El mejor resultado de obtiene con el *modelo RGF*, ya que genera conexiones donde otros no lo hacen, por ejemplo, entre la tercera y la cuarta rejilla superior.

Con estas pruebas se ha comprobado de forma cualitativa las salidas de los diferentes modelos. Para validar de forma cuantitativa estos resultados se va a computar el *análisis procrustes* [KEND84] entre las gráficas generadas y la gráfica real. Este análisis trata de encontrar la traslación, rotación y escala existente entre una gráfica y la gráfica objetivo mediante la minimización del error cuadrático medio entre ellas. Como en este apartado, los modelos de córtex y los valores reales del córtex no poseen las mismas unidades es interesante utilizar una medida que evite la escala. También la traslación es importante, ya que se han visto efectos que posee un grado de “ruido de fondo” generado biológicamente que, evidentemente, no está en el modelo. Por ello también es interesante eliminar toda la componente continua de las gráficas y analizar sólo la forma de las mismas.

En la

Tabla IV-4[1] se muestran los errores cuadráticos de la correlación de la gráfica generada con respecto a la original, así como su suma total, por lo que a menor valor, mejor coincidencia entre la salida del modelo y la realidad.

[1] En la Tabla IV-4 se presenta el error cuadrático cometido para cada efecto y por cada modelo. Este error se calcula correlando las gráficas ideales y generadas evitando traslaciones, rotaciones, etc. Este tipo de comparativa es especialmente sensible en ciertos casos, como por ejemplo el *Modelo SERRE* para la Facilitación ortogonal por supresión. A pesar de que visualmente se comprueba que no es válido, ya que genera salida para señales paralelas, el error cuadrático es bajo puesto que se invierte la señal para evitar el "efecto espejo". Por ello, es necesario tener especial cuidado al analizar esta tabla.

Tabla IV-4: Error cuadrático de la respuesta de los modelos con respecto a los efectos reales

Efectos	***CRF***	***SERRE***	***DG1***	***DGF***	***RG1***	***RGF***	***Filtro RGF***
Preferencia ante una orientación	**0.009**	0.024	0.146	0.146	0.378	0.378	0.172
Supresión del entorno	0.672	0.770	0.451	0.513	0.135	**0.133**	0.921
Facilitación ortogonal por supresión	0.120	0.093	0.238	0.238	**0.047**	**0.047**	0.335
Facilitación por conexiones horizontales	0.704	0.704	0.321	**0.255**	0.522	0.514	0.561
Variación no lineal de la respuesta con el contraste	0.018	0.019	0.021	0.021	**0.015**	**0.015**	0.021
Error cuadrático total	1.523	1.517	1.177	1.173	1.097	**1.087**	2.010

Analizando todos los resultados obtenidos, se pueden extraer una serie de conclusiones que permitirán continuar con la implementación de un modelo de neurona de V1 válido para las aplicaciones de visión artificial. La conclusión principal es que los modelos RG1 y RGF son los que simulan de forma más completa las salidas de las neuronas reales ya que poseen el menor error cuadrático agregado para todos los efectos y el menor error en la mayoría de los efectos individuales. De forma numérica, el peor efecto modelado es la facilitación por conexiones horizontales, que no es tan acentuado como en los modelos basados en resta de Gaussianas. A pesar de ello, el análisis de las curvas permite ver que la curva generada por el modelo *RGF* es mucho más similar a la ideal que el *RG1*, ya que es un efecto sutil.

Como conclusiones secundarias, se comprueba que el *Filtro basado en RGF* no es válido ya que genera señales no deseadas y el error cuadrático generado es el mayor. Por otra parte, del *Modelo RG1* y *Modelo RGF*, el mejor es el último, ya que modela de forma clara la facilitación en el experimento psicofísico, además de generar una curva ligeramente

más similar que el *Modelo RG1* para el efecto de supresión del entorno, debido a la ligera facilitación existente.

IV.5.3 Modelos de córtex y respuesta neuronal conjunta

En el anterior apartado se han evaluado diferentes modelos de neuronas simples de la capa 2/3 del córtex V1 de los macacos. El objetivo de esta propuesta de neuronas de la capa V1 no es el propio diseño, sino que es su aplicación al campo del análisis de las imágenes. Por ello, en este apartado planteamos un segundo conjunto de pruebas. En estas pruebas, se generará un modelo completo de V1 y se analizará la salida de todas las neuronas trabajando conjuntamente. El modelo completo de V1 generado es regular y tiene la forma mostrada en la Figura IV-55.

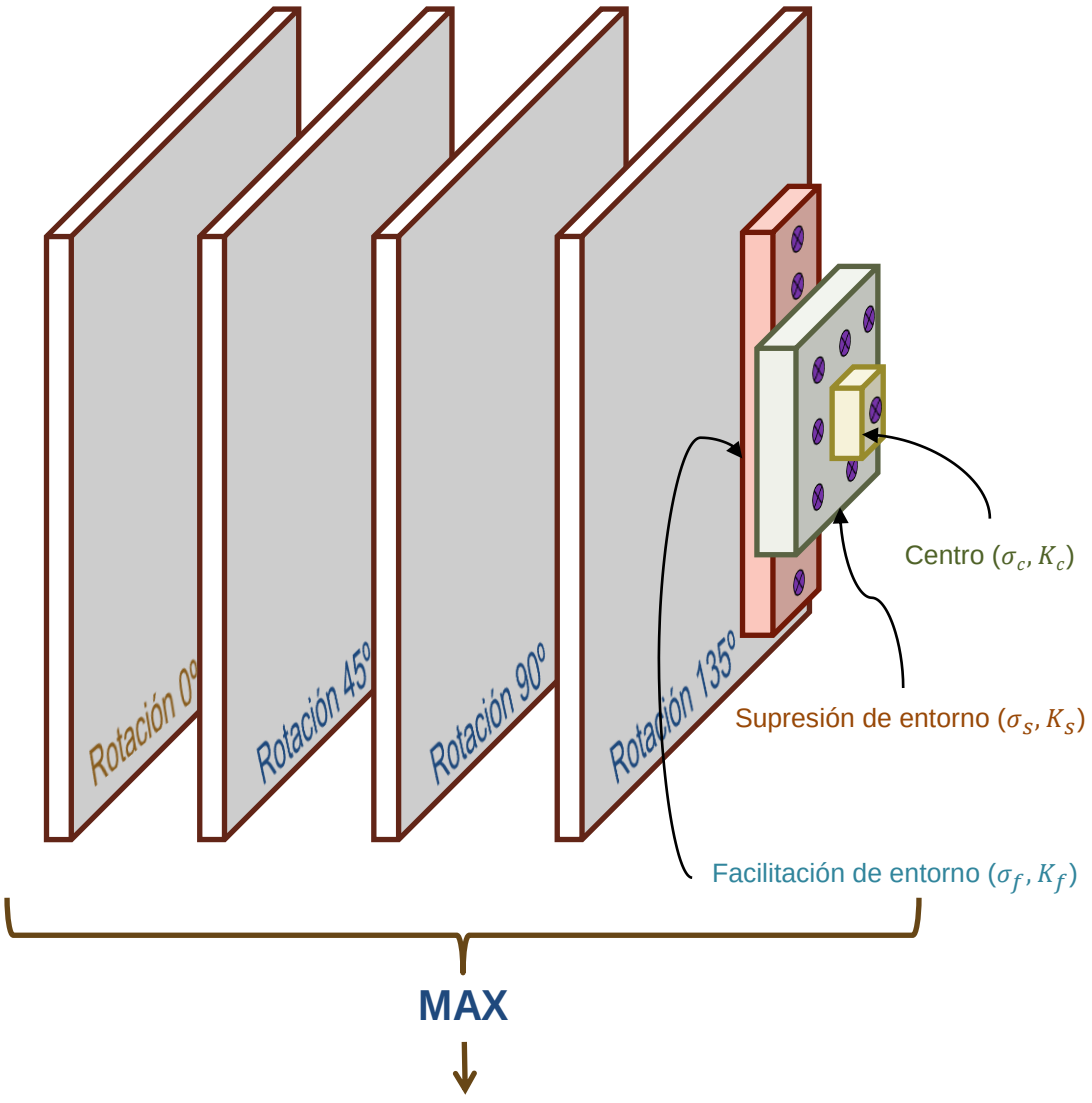


Figura IV-55: Modelo completo de V1 propuesto en esta tesis

En el modelo presentado, cada píxel de la imagen se corresponde con el centro del CRF de una neurona modelada. Para el cómputo del ERF se utilizan las neuronas del entorno (en función del modelo), que están dispuestas con los parámetros definidos en el apartado anterior. Además, en cada píxel, no solo existe una única neurona, sino que existen 4 tipos de neuronas, cada una con su orientación y sus conexiones entre la misma orientación. Este mismo patrón está repetido para cada píxel de la imagen.

Para este segundo conjunto de pruebas se han seleccionado ocho imágenes que pretenden representar diferentes tipos de efectos y texturas que se pueden encontrar en imágenes reales. El primer conjunto de imágenes a presentar se muestra en la Figura IV-56.

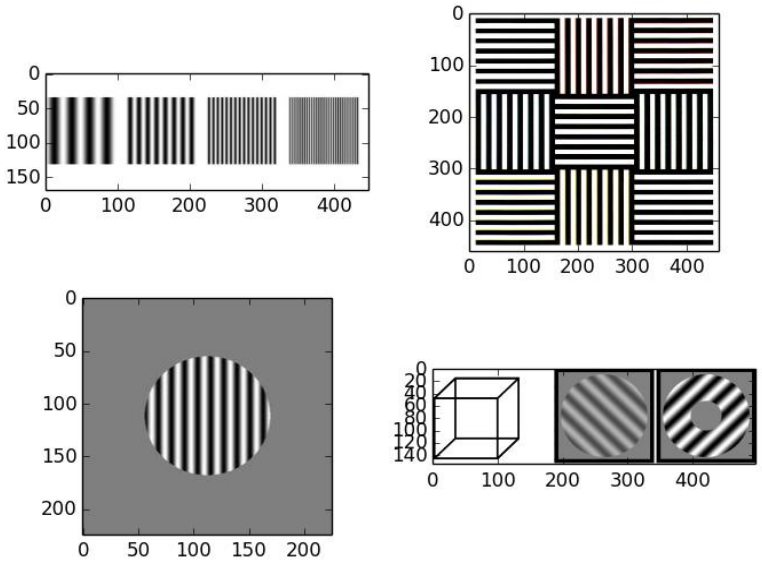


Figura IV-56: Diferentes rejillas para simular un estímulo orientado

En esta imagen aparecen diferentes rejillas con diferentes formas, orientaciones y contrastes. La salida esperada por un buen modelo de córtex es que las neuronas de la orientación correcta se activen y tengan una respuesta mayor que las que tienen una preferencia por diferente orientación.

Para hacer la prueba, se ha creado el modelo completo de córtex basándose en los modelos descritos y probados anteriormente, y para cada píxel se ha obtenido qué neurona (y por tanto, qué orientación) es la que tenía una mayor respuesta ante su excitación. Para visualizarlo se ha usado un código de colores, donde el amarillo representa la capa de 0º (o barras con orientación horizontal), el azul oscuro la capa de 90º, azul claro 45º y marrón 135º. Las respuestas obtenidas por los diferentes modelos se muestran en las siguientes.

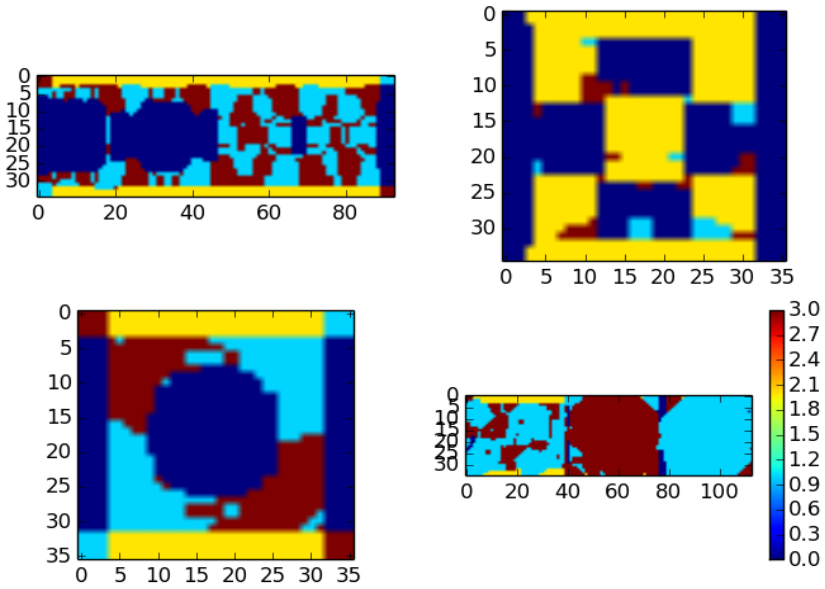


Figura IV-57: Respuesta completa del Modelo SERRE

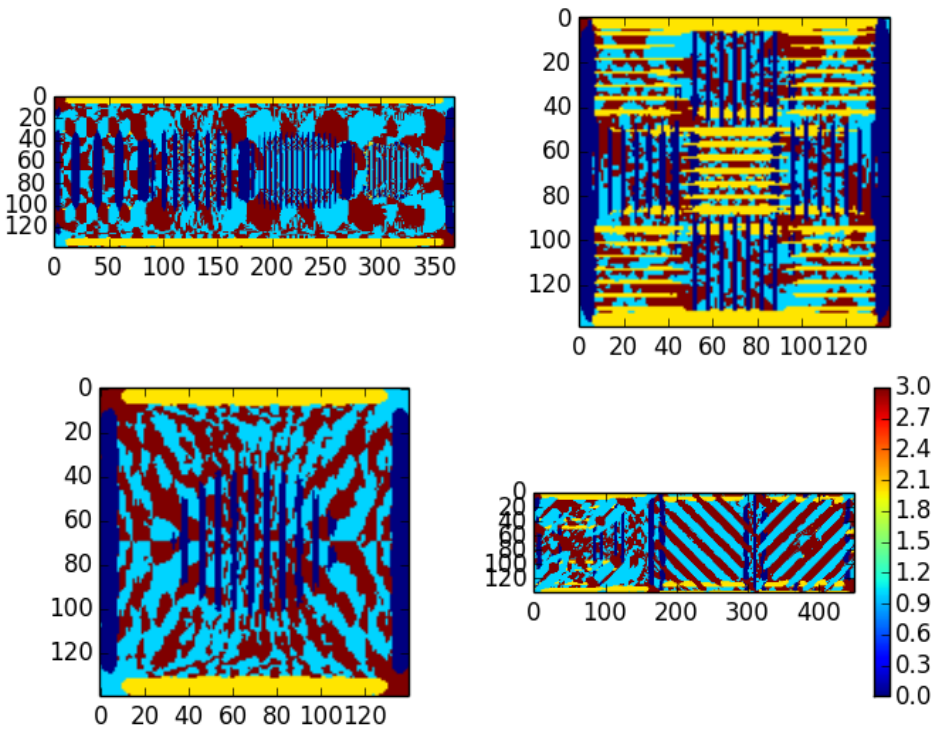


Figura IV-58: Respuesta completa del Modelo DG1

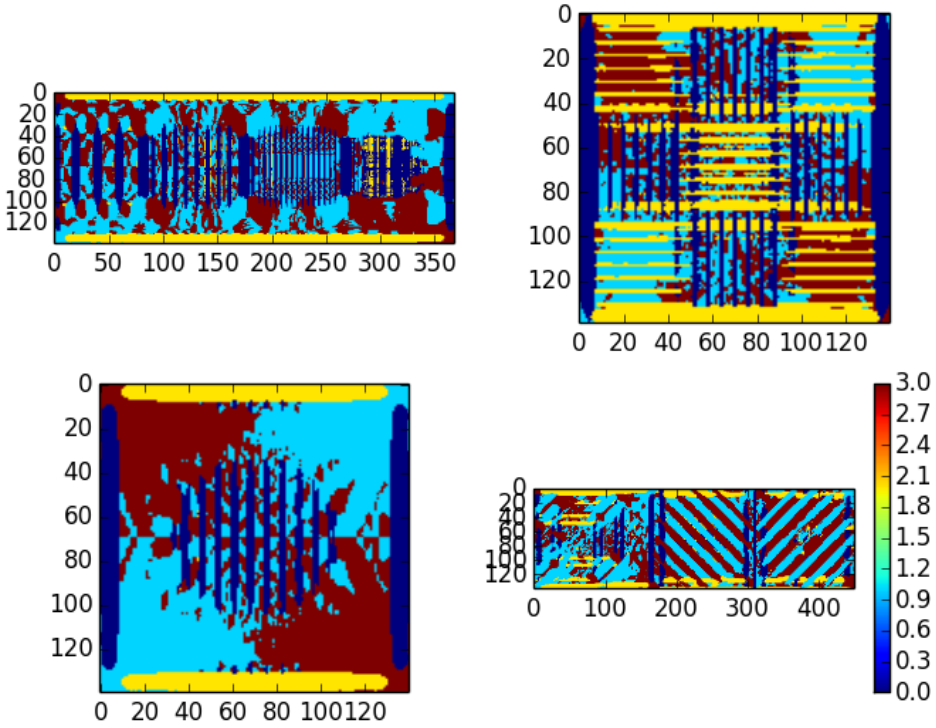


Figura IV-59: Respuesta completa del Modelo DGF

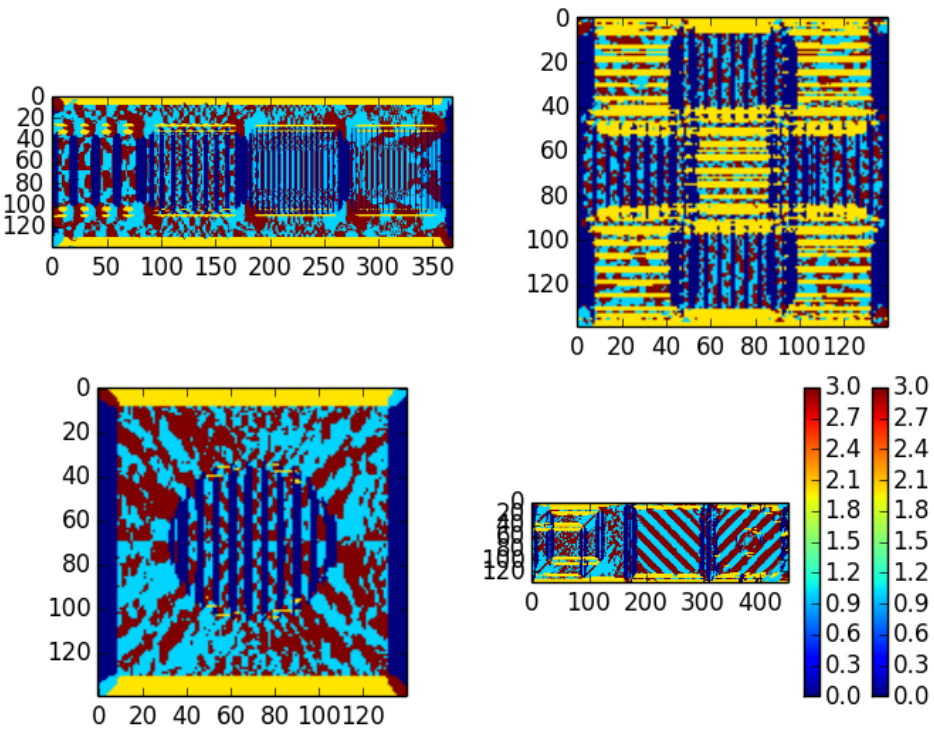


Figura IV-60: Respuesta completa del Modelo RG1

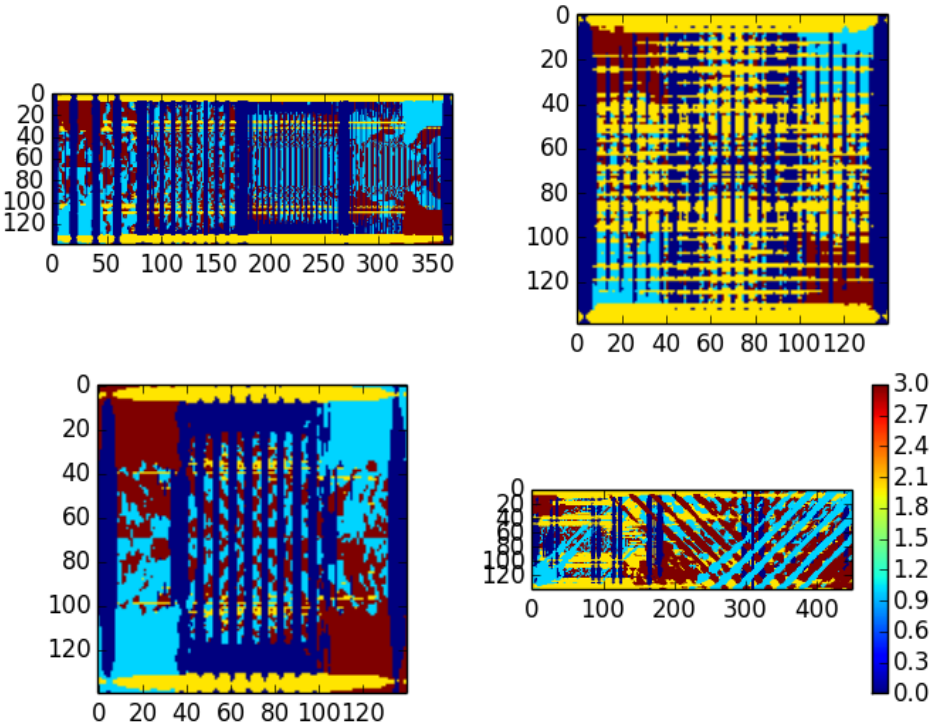


Figura IV-61: Respuesta completa del RGF

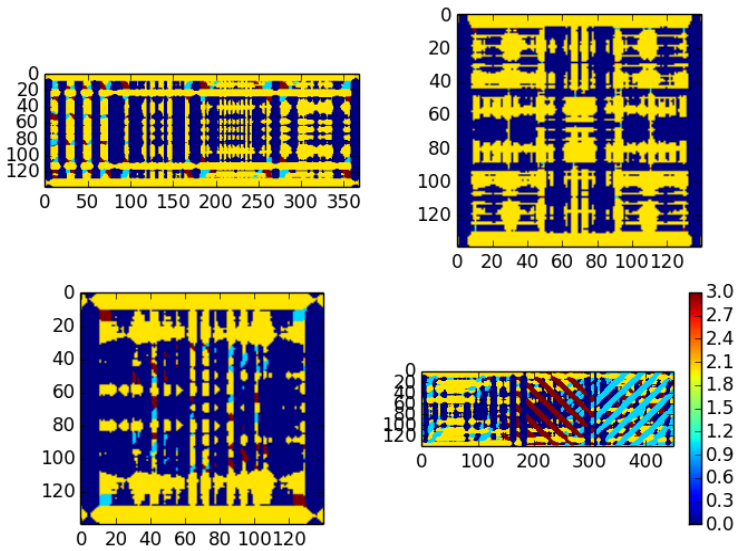


Figura IV-62: Respuesta completa del Filtro basado en RGF

En las imágenes anteriores se comprueba lo mismo que se ha visto en los modelos de neuronas simples. Por un lado, el *Modelo SERRE* sirve para ver de forma general la respuesta, pero no es nada preciso debido a su capa de selección del máximo. En cuanto a los modelos lineales (*Modelo DG1* y *DGF*), son mucho más precisos, pero en las orientaciones oblicuas no funcionan correctamente (ver respuesta inferior derecha). Por otra parte, los modelos no lineales son más precisos. Así, en el *Modelo RG1*, se comprueba cómo incluso en el cubo de la respuesta inferior derecha se detectan esas pequeñas líneas oblicuas de forma correcta, pero las rejillas oblicuas generan unos resultados no satisfactorios. A pesar de ello, el *Modelo RGF* sí que genera la respuesta esperada, aunque con más ruido debido al exceso de facilitación. A pesar de ello, este exceso de ruido podrá ser eliminado configurando de forma más precisa sus parámetros. Por último, se ve de nuevo cómo el *Filtro basado en RGF* no obtiene ningún resultado relevante.

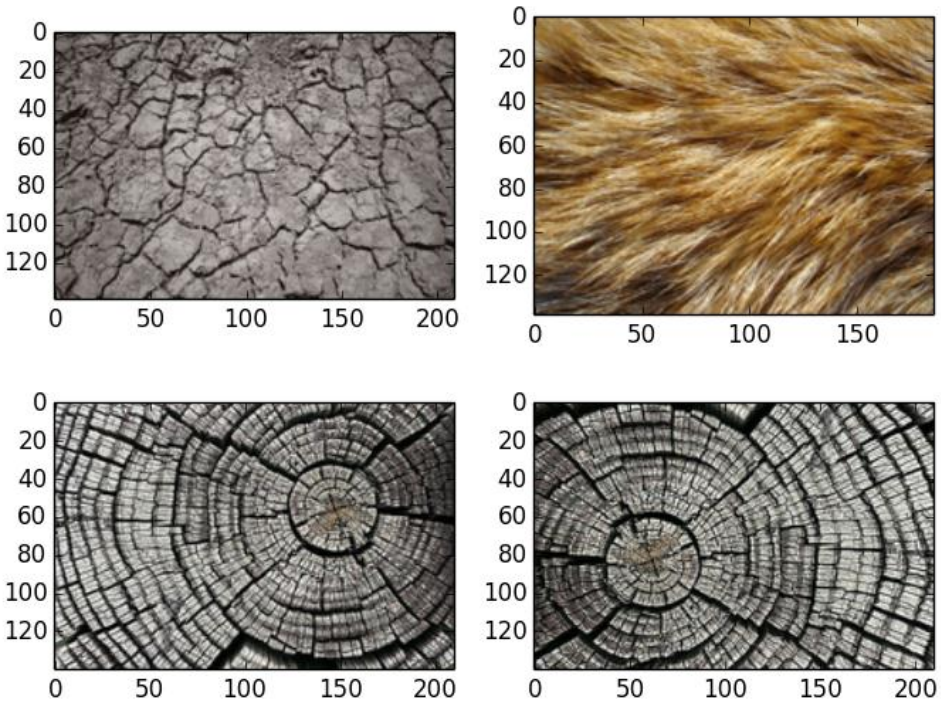


Figura IV-63: Diferentes imágenes de texturas reales

El segundo conjunto de imágenes de prueba, mostrado en la Figura IV-63, es más realista. Se pueden diferenciar cuatro imágenes que pretenden representar diferentes tipos de textura. Por un lado, las dos imágenes superiores buscan representar una textura, en el caso de la imagen de la izquierda es una textura con líneas bien definidas y grandes contrastes, y en el caso de la derecha es una textura con diferentes tipos de contraste y líneas menos claras. En la parte inferior se encuentran dos imágenes iguales, con una textura definida y circular. La diferencia es que la imagen de abajo a la derecha está rotada 180º y modificada con una lente de ojo de pez, por lo que se buscará que el modelo de córtex pueda detectar el mismo tipo de textura en ambos.

Al igual que en el caso anterior, se ha computado la orientación de cada píxel y los resultados se visualizan en la Figura IV-64.

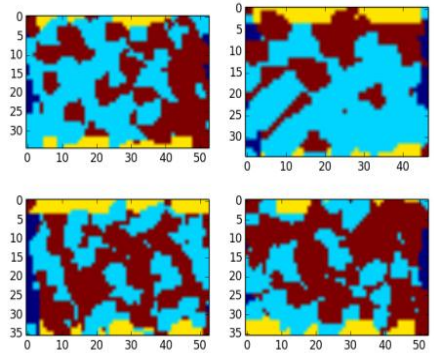
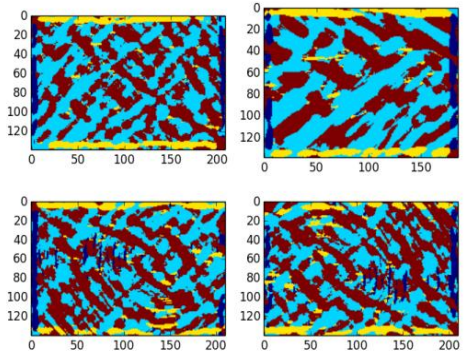
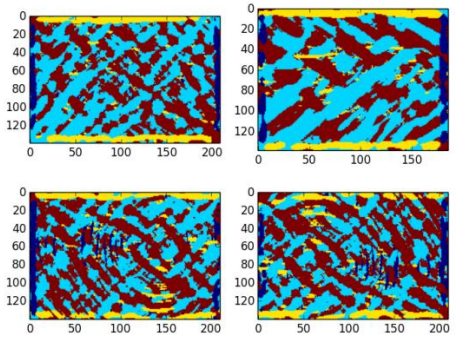
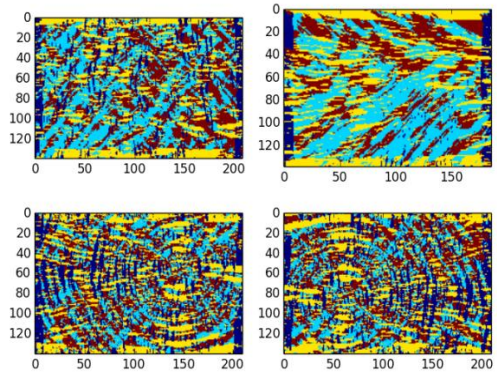
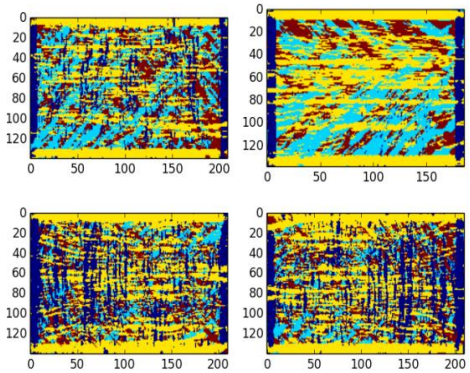
Serre	Sceniak	Bosking lineal
		
Cavanaugh	**Bosking normalizado**	
		

Figura IV-64: Respuesta de los filtros seleccionados ante el segundo conjunto de imágenes de prueba

La primera comprobación es que los modelos no lineales pueden detectar mejor la textura presente. De hecho, el modelo que más detalles posee es el *Modelo RG1*, ya que el *RGF* genera demasiados artefactos debido a la excesiva facilitación.

En este conjunto de imágenes lo que realmente importa no es la respuesta de cada sino ver en conjunto si se describen bien las imágenes y si todas las neuronas entre ellas de forma correcta. Para ver este efecto se va a generar un descriptor matemático que tenga cuatro elementos y cada uno de ellos contabilice el número repeticiones de una orientación. Así, en las dos imágenes inferiores se deberá dar mismo descriptor, mientras que en las dos superiores no. Los resultados gráficos dichos descriptores se muestran en la

Figura IV-65, donde la imagen *t1* es la superior izquierda, la *t2* es la superior derecha, la *t3* es la inferior izquierda y la *t4* es la inferior derecha.

Para visualizarlos, en el eje X se ven los cuatro valores del descriptor, mientras que en el eje Y se ve el valor del propio descriptor.

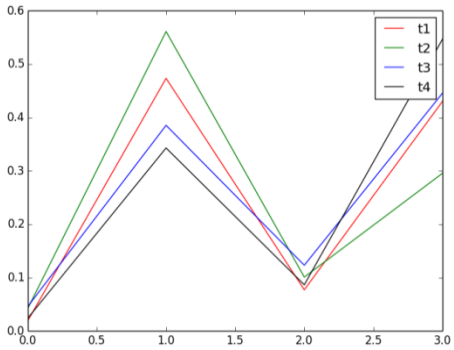
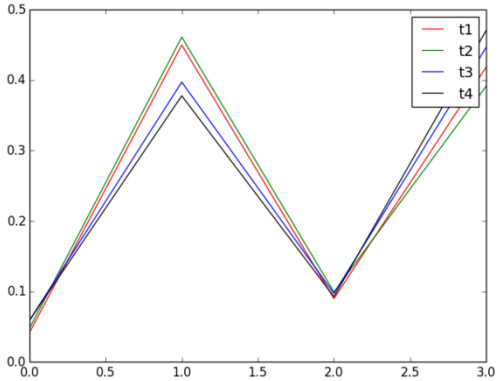
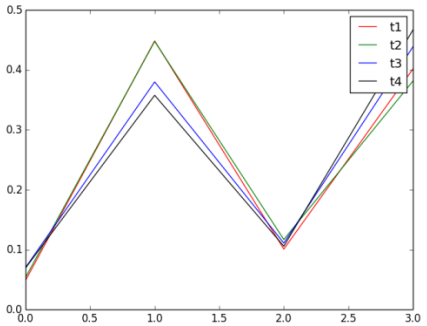
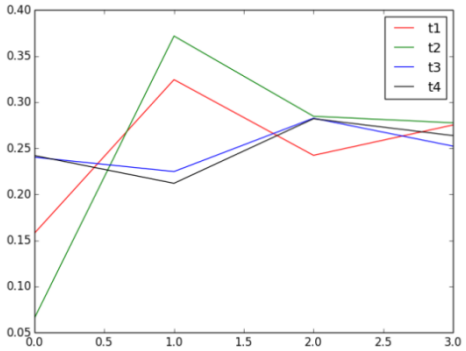
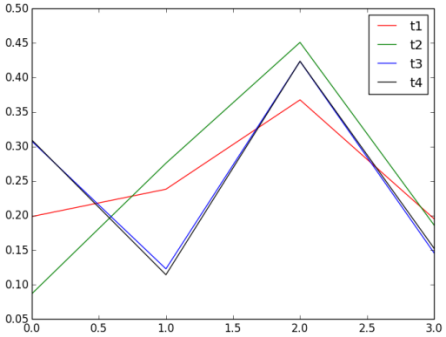
Serre	Sceniak	Bosking lineal
		
Cavanaugh	**Bosking normalizado**	
		

Figura IV-65: Descriptores generados para cada imagen del conjunto de prueba

En las gráficas anteriores se comprueba cómo el *Modelo RGF* genera descriptores aproximadamente iguales para las dos imágenes inferiores (*t3* y *t4*), por lo que el resultado obtenido es mejor que en el caso del *Modelo RG1*.

IV.6 Conclusiones

Tras un estudio exhaustivo de la literatura del campo de ciencias neurológicas sobre el funcionamiento del córtex visual de los macacos, se ha visto cómo las neuronas simples de la capa 2/3 de V1 poseen un campo receptivo clásico y uno extendido que interactúan bajo el concepto centro-entorno. Este concepto se puede modelar como ratio de gaussianas (RoG) o como diferencia de gaussianas (DoG), pero también se ha visto que existen otros efectos neuronales útiles para el campo de análisis de imágenes, que son:

- Selectividad con la orientación
- Supresión con la misma orientación si la excitación es cercana a la neurona
- Aumento de la excitación por la orientación cruzada a la óptima para una neurona
- Conexiones horizontales lejanas co-orientadas e co-alineadas permiten una facilitación de la respuesta
- Modificación de propiedades con el contraste:
    - Variación de la respuesta no lineal con el contraste
    - Modificación de la extensión de la neurona con el contraste
    - Modificación de la facilitación lejana con el contraste

En base a estos estudios hemos propuesto diferentes modelos, que modelan uno o varios efectos de los anteriores. Como resultado, se ha visto que los modelos no lineales son los que mejor modelan el comportamiento real del córtex y, entre ellos, el *Modelo RGF* es el que puede dar los mejores resultados potenciales en el campo de la visión artificial.

Una vez que se tiene un modelo neuronal seleccionado, el siguiente paso es introducirlo en el estándar MPEG7 y ver la mejora del mismo.

V. Nueva implementación del estándar MPEG7-EHD bioinspirada

V.1 Introducción

Esta tesis está enfocada a mejorar el modelo de vecinos más cercanos para la anotación automática de imágenes. Como se ha descrito, este modelo posee dos pasos: primero se aplica un procedimiento para obtener, de una base de datos de referencia, las imágenes visualmente más similares a la que se desea anotar; y segundo, se realiza una propagación de las etiquetas de las imágenes similares a la imagen de consulta en base a diferentes parámetros. Tanto en el capítulo IV como en este capítulo V nos centramos en mejorar el primero de los pasos.

Como se ha visto en el capítulo II, el trabajo seminal de Makadia et al.[MAKA10] ha sido seguido por el resto de investigadores en el estado del arte y esta tesis no es una excepción. En dicho trabajo se propone un método de búsqueda de las imágenes más similares en base a la descripción de todas las imágenes mediante una combinación de descriptores de Textura y Color, para posteriormente ejecutar un algoritmo de búsqueda en base a una combinación de distancias. En el capítulo III se ha mostrado que la combinación más óptima y compacta de descriptores de color y textura se basa en los descriptores Scalabe Color Descriptor (SCD) y Edge Histogram Descriptor (EHD) pertenecientes al estándar MPEG7.

En esta tesis se ha analizado el funcionamiento interno de ambos descriptores pertenecientes al estándar MPEG7 y se ha detectado una diferencia fundamental que debe ser recalcada (capítulo III): **mientras que el descriptor SCD utiliza el concepto de color perceptual tal y como lo perciben los humanos, el descriptor EHD utiliza varios filtros de bordes definidos arbitrariamente.** Por tanto, en este capítulo se pretende definir una nueva implementación del descriptor EHD basado también en la percepción humana para mejorar su funcionamiento.

Para tal fin, en el capítulo IV se ha comenzado a trabajar sobre esta característica y hemos propuesto varios modelos de córtex visual primario que permiten representar los bordes de una imagen de forma fiel a cómo los representa el cerebro de un primate. Gracias a los modelos definidos, en el presente capítulo se busca crear una nueva implementación del estándar MPEG7-EHD. La finalidad es lograr mantener las recomendaciones del estándar cambiando la forma de detección de los bordes de la imagen. Así, con la nueva implementación se tendrá un descriptor de bordes bioinspirado,

al igual que el descriptor MPEG7-SCD, y con un resultado más fiel a la realidad haciendo que las búsquedas por similitud de imagen puedan mejorar debido a que la representación de textura será mejor que la original del estándar.

De cara a ejecutar este trabajo, en el apartado V.2 de este capítulo se repasa la especificación del descriptor MPEG7-EHD. A continuación, se analiza en detalle el resultado que se obtiene con este descriptor y se trabaja en los puntos de integración del modelo del córtex en el estándar MPEG7-EHD. Tras la integración, en el apartado V.4 se realizan los ajustes necesarios de los parámetros del modelo de córtex para comprobar si los parámetros biológicos seleccionados en el capítulo IV eran adecuados para esta problemática. Finalmente, se muestran los resultados comparativos sobre la tarea de búsqueda de imágenes similares del descriptor estándar MPEG7-EHD y de la nueva implementación basada en el córtex de un primate.

V.2 El descriptor MPEG7-EHD

El descriptor MPEG7-EHD se centra en describir la textura de una imagen en niveles de gris en base a los bordes de la misma. Para tal fin, el estándar define cinco tipos de bordes (cuatro direccionales y uno no direccional). Así, ante la imagen de entrada, se aplican los cinco filtros, se computan una serie de histogramas que dan una idea de la distribución de los bordes, y finalmente se genera el descriptor final que se basa en una concatenación de dichos histogramas. Para información más detallada se referencia al apartado II.2.6.2 de esta tesis.

Los pasos generales que hay que dar para obtener el descriptor se presentan en la Figura V-1.

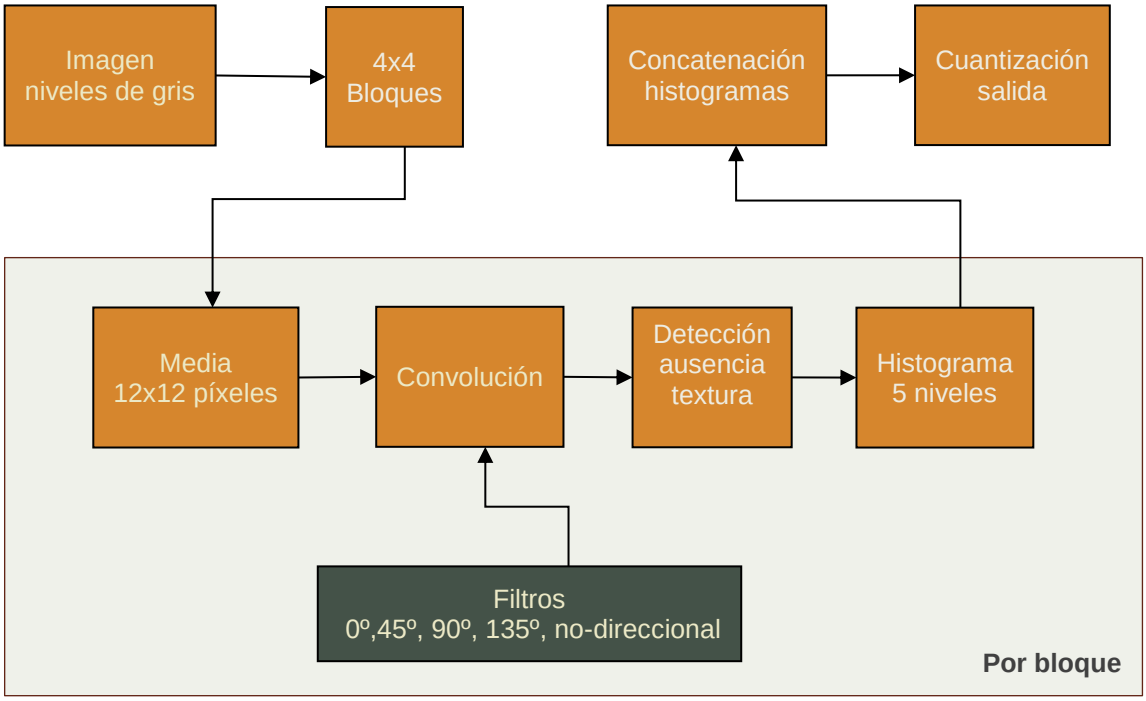


Figura V-1: Grafico de la implementación del descriptor MPEG7-EHD

En resumen, el descriptor EHD se basa en un histograma de orientaciones calculado con filtros sugeridos por el estándar. En el capítulo IV hemos propuesto un modelo de córtex llamado *Modelo RGF*, que es capaz de detectar las orientaciones de cada píxel, y el objetivo de este capítulo será introducir la salida de este modelo de córtex en el histograma del descriptor EHD, sustituyendo así los filtros definidos manualmente por MPEG7.

Esta integración se realizará en dos fases: en la primera se analizará la calidad de las salidas, tanto del estándar MPEG7 como del Modelo de córtex sobre imágenes reales. Esto permitirá conocer en qué puntos se puede mejorar el Modelo de cara a ajustar sus parámetros de configuración. En la segunda fase se utilizará una base de datos de imágenes para este ajuste y se obtendrán estadísticas que permitan guiar dicho ajuste.

V.3 Integración MPEG7-EHD y Modelo de córtex

La integración del modelo de córtex propuesto en esta tesis con el estándar MPEG7 es sencilla puesto que se ha realizado un diseño de córtex pensando en dicha integración. Así, la integración realizada entre ambos se muestra en la Figura V-2, donde en otro color se indican los módulos diferentes que se utilizan sobre el estándar (Figura V-1).

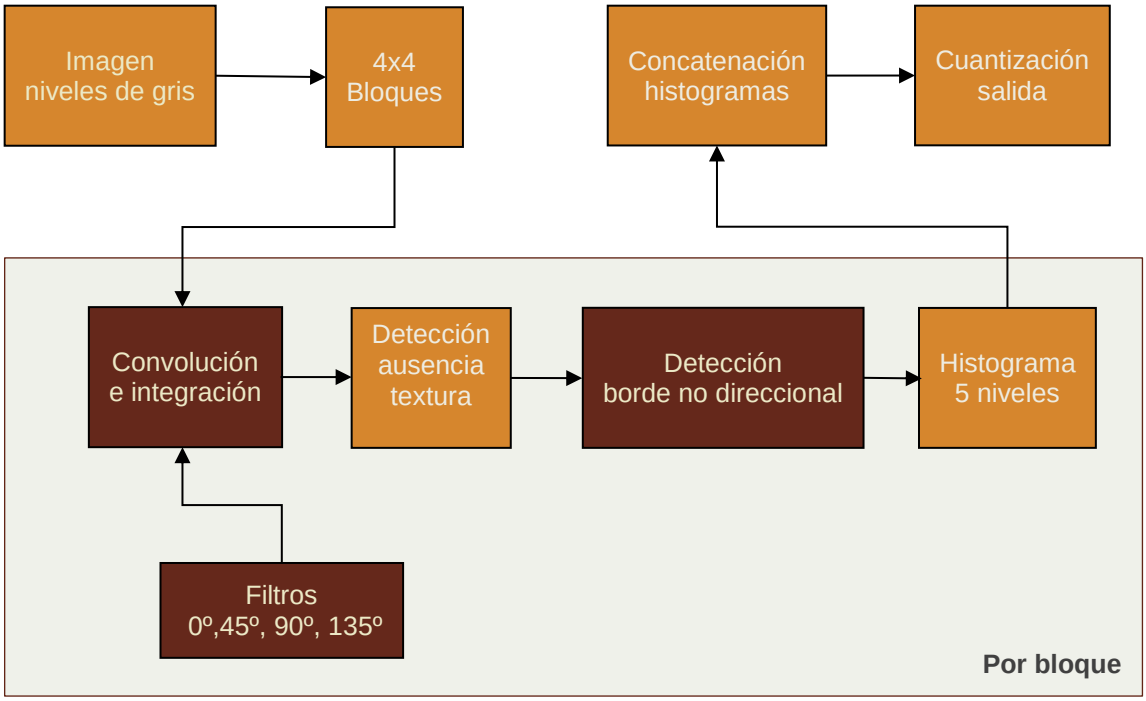


Figura V-2: Integración del modelo de córtex V1 propuesto en esta tesis con el estándar MPEG7-EHD

Se ve cómo han sido tres bloques los que se han sustituido. En primer lugar, tras computar la imagen de gris, para cada bloque de los 16 se ha aplicado el Modelo de córtex con los filtros definidos en el capítulo IV. Con las salidas para las 4 orientaciones se ha definido un umbral, de tal forma que si ninguna de las salidas supera el umbral se ha considerado que no existe textura, tal y como se sugiere en el propio estándar MPEG7. Para diferenciar la presencia de un borde sin orientación, se ha analizado la desviación estándar de los resultados de las cuatro orientaciones, así si esta desviación es baja, es porque no hay ningún tipo de borde, ya que ninguno de los filtros ha generado un resultado relevante. En cambio, si la desviación es alta, es porque los filtros están dando señales débiles y diferentes, por lo que hay textura no direccional.

Con esta integración, el primer paso a dar para comprobar si es correcta, es analizar visualmente de forma cualitativa qué resultados se obtienen usando el estándar MPEG7-EHD frente a los resultados obtenidos por la propuesta definida en esta tesis. Para mostrar los resultados del estándar, se ha trabajado con dos implementaciones diferentes:

por un lado se ha utilizado la implementación de referencia y, por otro, se ha modificado esa implementación para aumentar la resolución de las imágenes de salida. Esto se debe a que la implementación de referencia agrupa los píxeles de una vecindad de 12x12 píxeles en base a su media y sobre esta media a baja resolución aplica los filtros.

Como se ha dicho, la primera prueba a realizar es ver una comparativa entre el resultado de la detección de bordes del estándar MPEG7 y la propuesta realizada. Para hacer estas pruebas se van a usar cuatro imágenes reales que poseen diferentes características: zonas sin textura, zonas con líneas claras en diferentes sentidos, zonas más redondeadas,… Las imágenes se presentan en la Figura V-3.

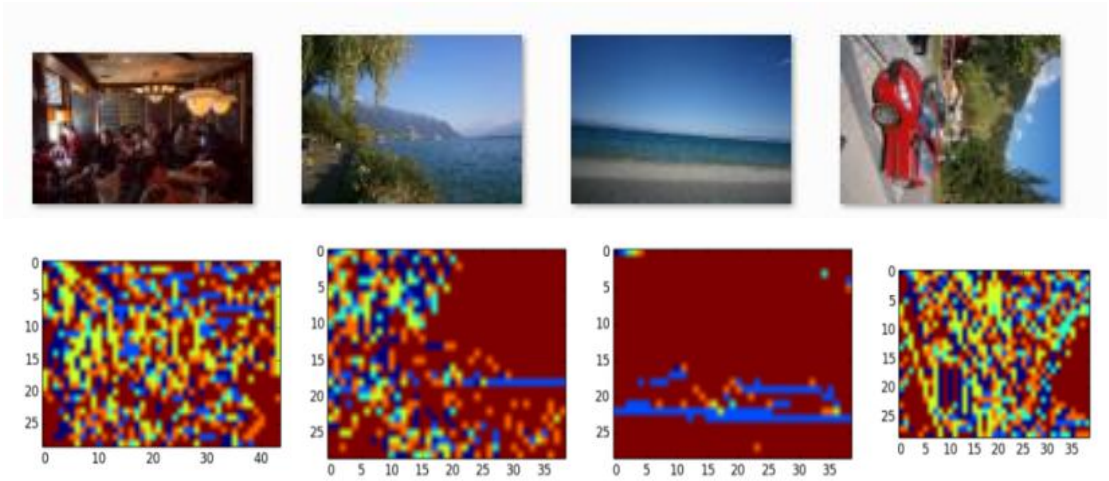


Figura V-3: Imágenes de prueba (arriba) y resultado del estándar MPEG7-EHD (abajo)

Aplicando la implementación de referencia de MPEG7-EHD sobre estas imágenes, el resultado se visualiza en la parte inferior de la Figura V-3. En ella se pueden ver diferentes colores representando a las diferentes orientaciones de cada píxel. El color azul oscuro indica una textura horizontal (0º), el color azul claro indica la presencia de una barra vertical (90º), el color azul verdoso indica 45º, el verde-amarillo muestra la presencia de elementos a 135º, el color naranja indica la presencia de una textura sin orientación clara, mientras que el color granate indica las zonas en las que no hay textura presente.

Para analizar este resultado en más detalle, también se ha aplicado la implementación de referencia modificada sobre las imágenes. Así se obtiene mejor resolución y es posible analizar y comparar los resultados visuales (Figura V-4).

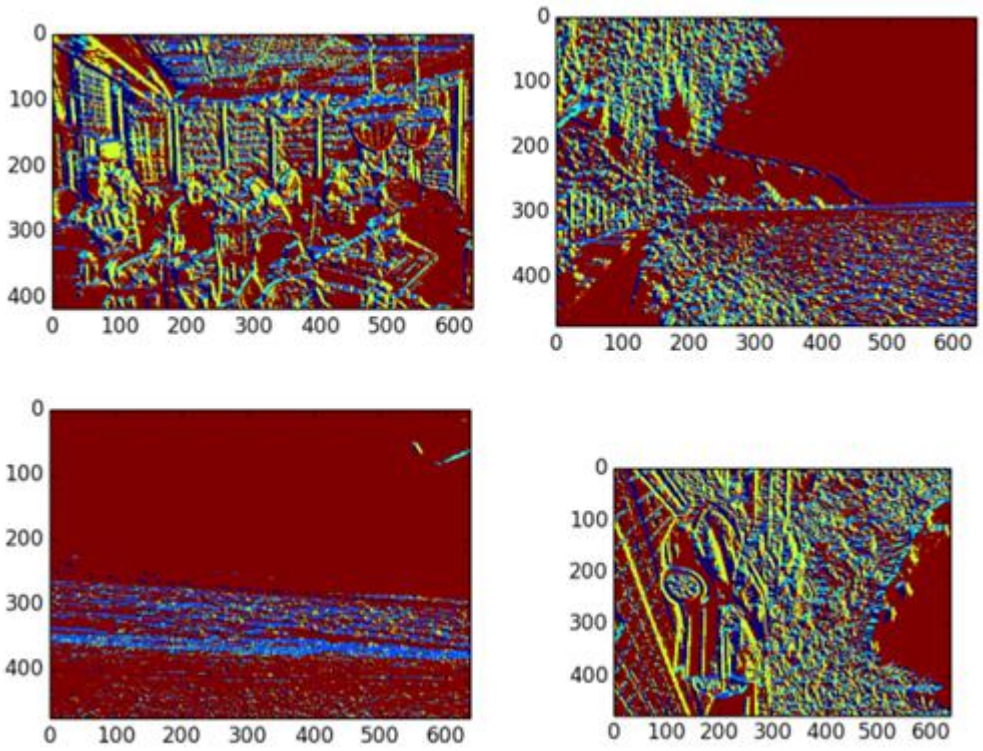


Figura V-4: Resultado con la implementación de referencia modificada para mayor resolución

Para calcular estas salidas se ha aplicado el *Modelo RGF* a una imagen directamente seleccionando un umbral determinado, para el cual si ninguna neurona tiene una respuesta mayor que dicho umbral se considera que no existe textura. Por ello, se obtienen 5 valores: cuatro de ellos para las cuatro orientaciones, más el color granate que indica la no presencia de textura.

En la Figura V-5 se ve el resultado del modelo de córtex *RGF* directamente aplicado para la imagen de entrada. Para ello, se han computado las salidas de las 4 capas de neuronas para cada píxel, y posteriormente se ha seleccionado el máximo de ellas para indicar la orientación del mismo. Por otra parte, en la Figura V-6 se ha incluido el Modelo dentro del estándar MPEG7 como hemos indicado en la Figura V-2, pero sin utilizar el umbralizado para la diferenciación de la textura no homogénea.

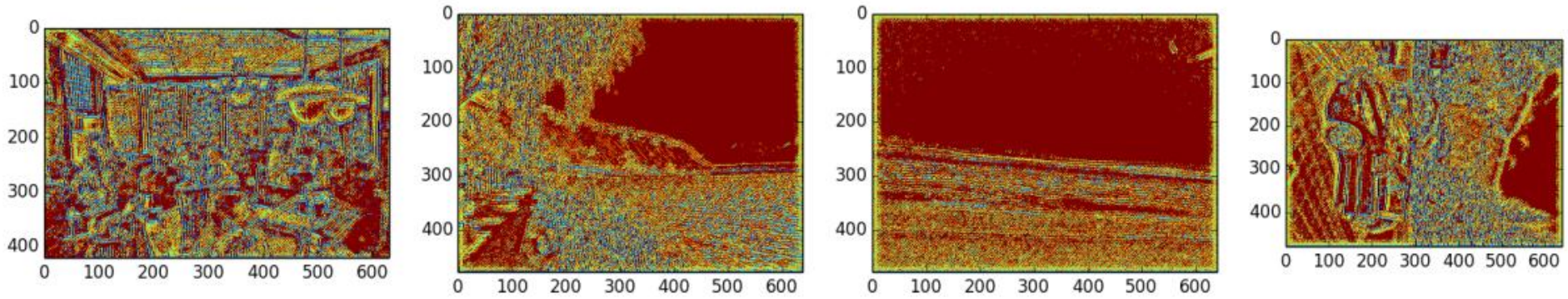


Figura V-5: Resultado del Modelo de cortex RGF

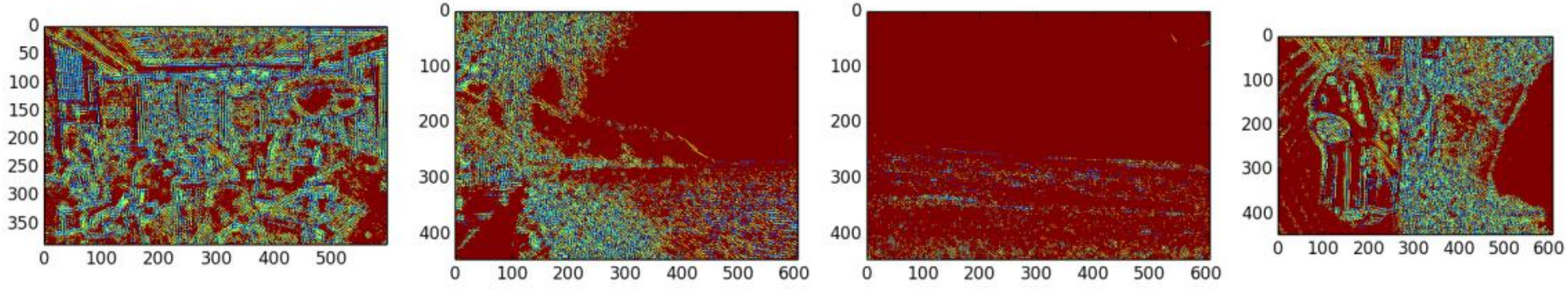


Figura V-6: Resultado del descriptor MPEG7-EHD usando el modelo de cortex RGF sin umbral direccional

Comparando la Figura V-5 y Figura V-6, se comprueba cómo el resultado del uso del Modelo de córtex tiene mucho más detalle que el propio estándar MPEG7 y que el estándar modificado. La salida de estas tres etapas es la que se muestra en Figura V-6, sin tener en cuenta la desviación estándar y, por tanto, sin tener en cuenta las texturas no direccionales. El resultado se asemeja al propio MPEG7, pero añade mucho más detalle en las texturas orientadas y fijándose en dicho detalle, se comprueba cómo el resultado es más certero que el propio estándar.

Añadiendo el análisis de la desviación estándar para las texturas no direccionales, se puede comprobar que el resultado mejora y se acerca a lo que un humano puede percibir. Para ello, se han seleccionado dos umbrales de la desviación típica, que son 5 y 50, y se han mostrado en la Figura V-7 y la Figura V-8 respectivamente.

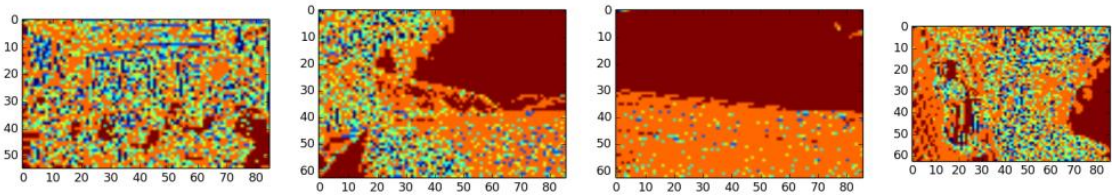


Figura V-7: En naranja se pueden ver las detecciones de textura no direccional usando un umbral basado en desviación estándar de valor 5

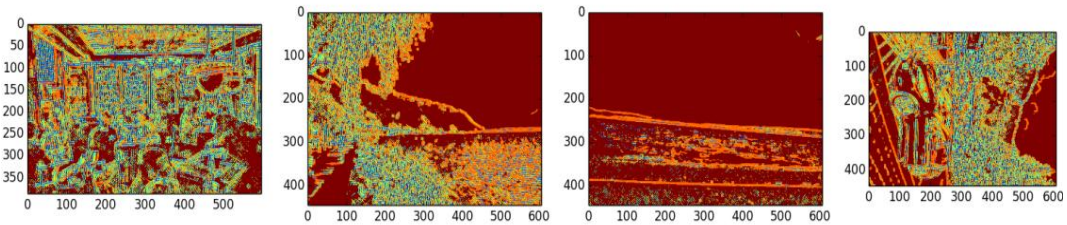


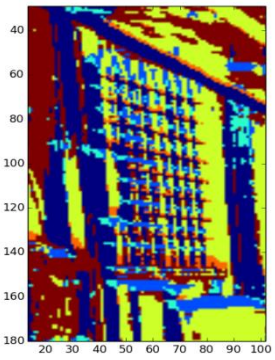
Figura V-8: En naranja se pueden ver las detecciones de textura no direccional usando un umbral basado en desviación estándar de valor 50

En estas ilustraciones se ve el efecto descrito anteriormente: cuanto mayor es la desviación, significa que existen más texturas no direccionales. Por ello, se comprueba que con un umbral alto se obtiene menos respuesta naranja, que es lo que tiene que suceder en la realidad.

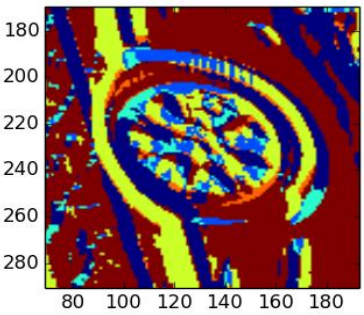
Para ver en conjunto los resultados cualitativos anteriores, se han seleccionado dos detalles en las imágenes de prueba: una ventana y una rueda de coche. Estos detalles poseen texturas orientadas en diferentes direcciones que generarán un resultado que puede ser directamente contrastado.

a - Ventana

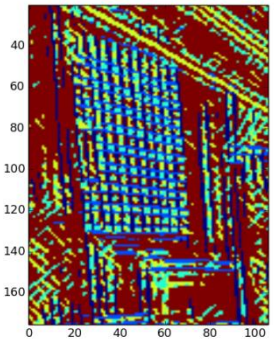
b - Rueda



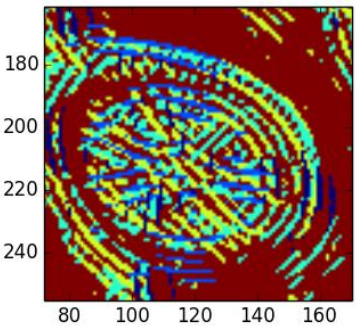
c - Ventana (MPEG7-EHD)



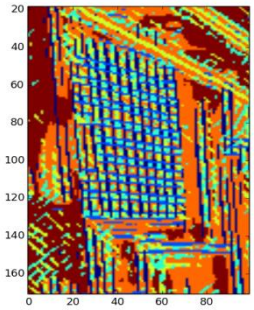
d - Rueda (MPEG7-EHD)



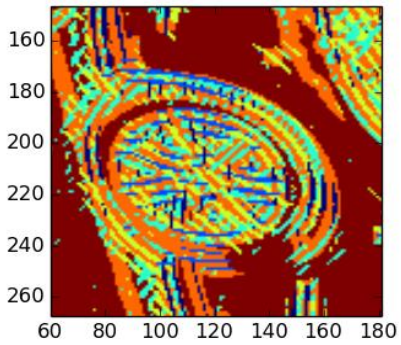
e - Ventana (MPEG7-EHD con *Modelo RGF* sin umbral no direccional)



f - Rueda (MPEG7-EHD con *Modelo RGF* sin umbral no direccional)



g - Ventana (MPEG7-EHD con *Modelo RGF*)



h - Rueda (MPEG7-EHD con *Modelo RGF*)

Figura V-9: Comparación de los resultados de varios modelos probados en dos imágenes.

Al usar directamente el estándar MPEG7-EHD, los resultados no se aproximan a la realidad. En cambio, al usar el Modelo de córtex *RGF* dentro del estándar MPEG7-EHD, se ve la realidad con más detalle: en la rueda se comprueban franjas azules claras en horizontal y oscuras en vertical, incluso en la sección redonda de la rueda. En oblicuo están presentes los otros dos colores, azul y verde, mientras que en otras zonas de la carrocería se muestra el color granate, al no tener textura. En cambio, el naranja aparece en zonas dudosas en cuanto a la direccionalidad de la textura, es más, quizá en algunas de esas zonas sea necesario marcarlas como granate. Dado que esta detección aparece tras ejecutar dos umbralizados de las respuestas de las neuronas artificiales, se hace necesario ajustar de forma fiable estos parámetros.

V.4 Ajuste de parámetros del *Modelo RGF*

Dado el gráfico de la Figura V-2, se puede intuir que en esta nueva implementación del estándar MPEG7-EHD existen numerosos parámetros a configurar. Por un lado, es necesario ajustar los parámetros del modelo neuronal y, por otro, los dos umbrales existentes en la integración con el estándar MPEG7-EHD. Para refrescar el *Modelo de córtex RGF*, a continuación se muestran de nuevo las ecuaciones del mismo:

$$\frac{Kc * L_c + Kf * L_f}{N + Ks * L_s}$$

siendo K_c, K_s y K_f las ganancias de los términos centro, supresión y facilitación respectivamente, y N es un término de normalización. L_c, L_s y L_f son los términos de integración de la respuesta neuronal:

$$L_c = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_c}\right)^2} dy \right)^2$$

$$L_s = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_s}\right)^2} dy \right)^2$$

$$L_f = \left(\int_{-x/2}^{x/2} e^{-\left(\frac{2y}{\sigma_f}\right)^2} dy \right)^{1.5}$$

donde σ_s,σ_c y σ_f representan el alcance espacial de sendas componentes. Además del valor empírico de N=2500 obtenido con anterioridad, para este modelo se habían obtenido una serie de parámetros de configuración en base a la literatura, y se resumen en la Tabla V-1.

Tabla V-1: Parámetros de configuración del modelo de córtex basados en los datos de macacos

Center	K_c	1
	σ_c	Tabla V-2
Surround – Supression	K_s	0.38
	σ_s	2.5 * σ_c
Surround – Facilitation	K_f	0.287
	σ_f	2.0 * σ_s

Tabla V-2: Parámetros de configuración de los filtros de Gabor que modelan el CRF [SERR07]

Banda Σ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Tamaño del filtro s	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
σ	28 & 3.6	4.5 & 5.4	6.3 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.2 & 22.8

A pesar de que estos parámetros de configuración han sido extraídos de la literatura neurocientífica, y todas las pruebas anteriores se han realizado en base a ellos, es necesario verificar que para aplicaciones de visión artificial estos parámetros son los adecuados. Por ello, en este apartado se van a validar estos parámetros usando una base de datos de imágenes complejas.

El protocolo de pruebas que definimos en este apartado establece que la base de datos a utilizar es la SAIAPRTC-12. Los valores que se van a testear son el *precision* y, en especial, el *recall* por cada imagen. Esto es debido a que este descriptor visual está orientado a la primera fase del algoritmo de anotación, que es la búsqueda de imágenes con contenido similar. En esa fase, lo ideal es que aparezcan el mayor número posible de etiquetas correctas, aunque esto haga que aparezcan etiquetas incorrectas, bajando el *precision*. Así, con todas ellas es posible afinar mejor en la segunda fase de propagación de etiquetas. Por ello, en esta fase se busca un *recall* alto, con independencia del *precision*, y en la fase de transferencia de etiquetas se busca un *precision* alta.

De esta forma, para evaluar los descriptores de imagen aplicados al algoritmo de similitud visual (ver *baseline* definido en el apartado III.5.1), se computará esta similitud y se hará una transferencia de todas las etiquetas de las 10 imágenes más cercanas. Con estas etiquetas es posible evaluar el recall y la bondad del algoritmo ya que, como se ha mencionado, un mayor recall indica que entre todas esas etiquetas de las 10 imágenes más cercanas existen más cantidad de etiquetas relacionadas con el *ground truth*.

Descartando σ_c, ya que otros trabajos del estado del arte ya han validado que ese valor es útil para los algoritmos de visión artificial [SERR07], las primeras pruebas se realizarán sobre los tres parámetros más importantes: σ_s, σ_f y N,. Las ganancias *K* se dejarán al

valor indicado, puesto que en el capítulo IV han sido validadas para poder tener una respuesta psicofísica correcta. En estas pruebas se fijarán los umbrales de detección de textura a un valor fijo que se ha determinado por validación cruzada. Además, cuando se modifique un parámetro, el resto de parámetros se fijarán al valor de la Tabla V-1.

El primer parámetro a modificar es σ_s. En la Tabla V-3 aparecen sus resultados para toda la base de datos SAIAPR-TC12.

Tabla V-3: Resultados para diferentes extensiones del campo de supresión

σ_s	1,25	1,67	1,58	2,08	2,5	2,8
Precision	0.084	0.086	0.085	0.086	***0.087***	*0.086*
Recall	0.536	0.549	0.546	0.552	***0.554***	*0.540*

Se comprueba que el mejor resultado es con el parámetro de configuración basado en la literatura.

El segundo de los elementos a modificar es N, ya que hasta ahora se está utilizando un valor empírico obtenido de la visualización gráfica de las neuronas independientes.

Tabla V-4: Resultados para diferentes valores de la constante de normalización

N	1	500	1000	1500	2000	2250	2500	3500	5000
Precision	0.087	0.087	0.087	0.086	0.087	0.087	0.087	0.087	0.087
Recall	0.549	0.554	0.552	0.555	***0.556***	0.555	0.554	0.554	0.553

En la Tabla V-4 se comprueba cómo el valor que se estaba utilizando de N=2500 no es el adecuado y se obtiene un mejor recall con N=2000.

Por último, se han realizado las pruebas mostradas en con σ_f.

Tabla V-5: Resultados para diferentes extensiones del campo de facilitación

σ_f	1,00	1,33	1,66	2,00	2,00
Precision	0.060	0.080	0.085	***0.087***	0.086
Recall	0.498	0.536	0.539	***0.554***	0.530

La tendencia es la misma que para el campo de supresión. Otro efecto que es interesante destacar es que se ve que cuanto menor es el campo de facilitación más perjudicial es para el sistema.

Con todos estos parámetros verificados, ya es posible pasar a obtener un resultado final en base al precision y recall sobre la base de datos SAIAPR-TC12. Existen dos parámetros adicionales a configurar. Por una parte la detección de presencia de textura se fija a un valor de 50, ya que los experimentos realizados indican que este valor es el óptimo. En cuanto a la definición del umbral de detección de bordes, se han probado varios valores y los diferentes resultados están presentes en la Tabla V-6.

Tabla V-6: Resultados ante diferentes umbrales de detección de bordes

th	0.1	0.15	0.2	0.23	0.25	0.27	0.3	0.35
Precision	0.092	0.098	0.099	0.098	0.098	0.097	0.095	0.094
Recall	0.574	0.595	0.605	0.613	***0.614***	0.610	0.600	0.598

El mejor resultado se obtiene con un umbral de 0,25. A pesar de ello, las pruebas realizadas se han llevado a cabo sobre exponente de L_f de 1,5. Es necesario comprobar si con un exponente de 2, de la misma forma que L_c y L_s, el resultado mejora. Los resultados con un exponente cuadrático se muestran en la Tabla V-7.

Tabla V-7: Resultados de diferentes umbrales de detección de bordes usando un exponente cuadrático

th	0.19	0.22	0.23	0.235	0.24	0.25	0.26	0.27	0.30	0.35
Precision	0.100	0.102	0.101	0.101	0.100	0.100	0.099	0.099	0.098	0.096
Recall	0.608	0.618	***0.622***	0.621	0.618	0.619	0.616	0.616	0.613	0.604

El resultado es mucho mejor, mejorando en un 1% el recall, por lo que se seleccionará este valor de exponente para L_f. De esta forma, los parámetros finales quedan configurado igual que en el sistema visual de los macacos, a diferencia del parámetro N, que será 2000, y el exponente de la facilitación, que será cuadrático al igual que L_c y L_s.

V.5 Comparativa con otros modelos de córtex

En el apartado V.3 se ha visto cómo se ha realizado la integración del *Modelo de córtex RGF* dentro del descriptor MPEG7-EHD. Este Modelo fue seleccionado en el capítulo IV ya que daba una respuesta más similar a la respuesta real del córtex de los macacos. En este apartado, se va a realizar una comparativa entre este modelo y el resto de modelos que se han definido en el mismo capítulo. Con los valores de parámetros validados en el apartado V.4 se van a aplicar los mismos modelos que en capítulo IV a la base de datos SAIAPR-TC12, para así completar la comparativa entre los modelos. Para su aplicación, se ha procedido a la integración de dichos modelos en el estándar MPEG7-EHD, tal y como se ha definido en el apartado V.3. La Tabla V-8, Tabla V-9, Tabla V-10 y la Tabla V-11 presentan los resultados para cada modelo en función de su umbral de detección de textura.

Tabla V-8: Resultados para el *Modelo SERRE*

th	250	300	350	400
Precision	0.081	0.079	0.079	0.076
Recall	***0.538***	0.534	0.537	0.511

Tabla V-9: Resultados para el *Modelo DG1*

th	10	20	30	50	70	80	100	125
Precision	0.099	0.099	0.099	0.096	0.095	0.093	0.093	0.091
Recall	0.598	0.599	***0.604***	0.601	0.603	0.597	0.593	0.586

Tabla V-10: Resultados para el *Modelo DGF*

th	4	8	10	20	50	80	100
Precision	0.101	0.100	0.099	0.097	0.095	0.093	0.092
Recall	***0.613***	0.613	0.610	0.601	0.596	0.591	0.590

Tabla V-11: Resultados para el *Modelo RG1*

th	0.05	0.10	0.15	0.20	0.25	0.30	0.40	0.5
Precision	0.089	0.091	0.098	0.098	0.098	0.095	0.093	0.091
Recall	0.562	0.571	0.595	0.610	***0.614***	0.599	0.595	0.586

Los valores de los modelos DG1 (Tabla V-9) y DGF (Tabla V-10) se pueden comparar entre sí para ver el efecto de la facilitación del córtex, ya que en el DGF está presente mientras que en el DG1 no. Se ve una mejora en el recall en un 1% al introducir este factor.

A su vez, los modelos RG1 (Tabla V-11) y RGF (Tabla V-7) también pueden ser comparados y la diferencia es el término de la facilitación. Al igual que en anterior caso, la introducción de la facilitación aporta cerca de un 1% de mejora sobre el conjunto del sistema.

Otra conclusión que se obtiene es que el modelo SERRE, basado completamente en el estado del arte [SERR07], tiene un resultado mucho peor que cualquiera de los modelos propuestos en esta tesis.

Para hacer un análisis global, podemos disponer toda la información anterior en un único gráfico en el que se muestre con diferentes puntos el precision y recall de todos los modelos independientemente de su umbral de detección de bordes. Esto es debido a la dependencia del resultado de un umbral de selección. Así, el mejor modelo será el que posea un precision y un recall mayor en todos sus puntos. El gráfico generado se muestra en la Figura V-10.

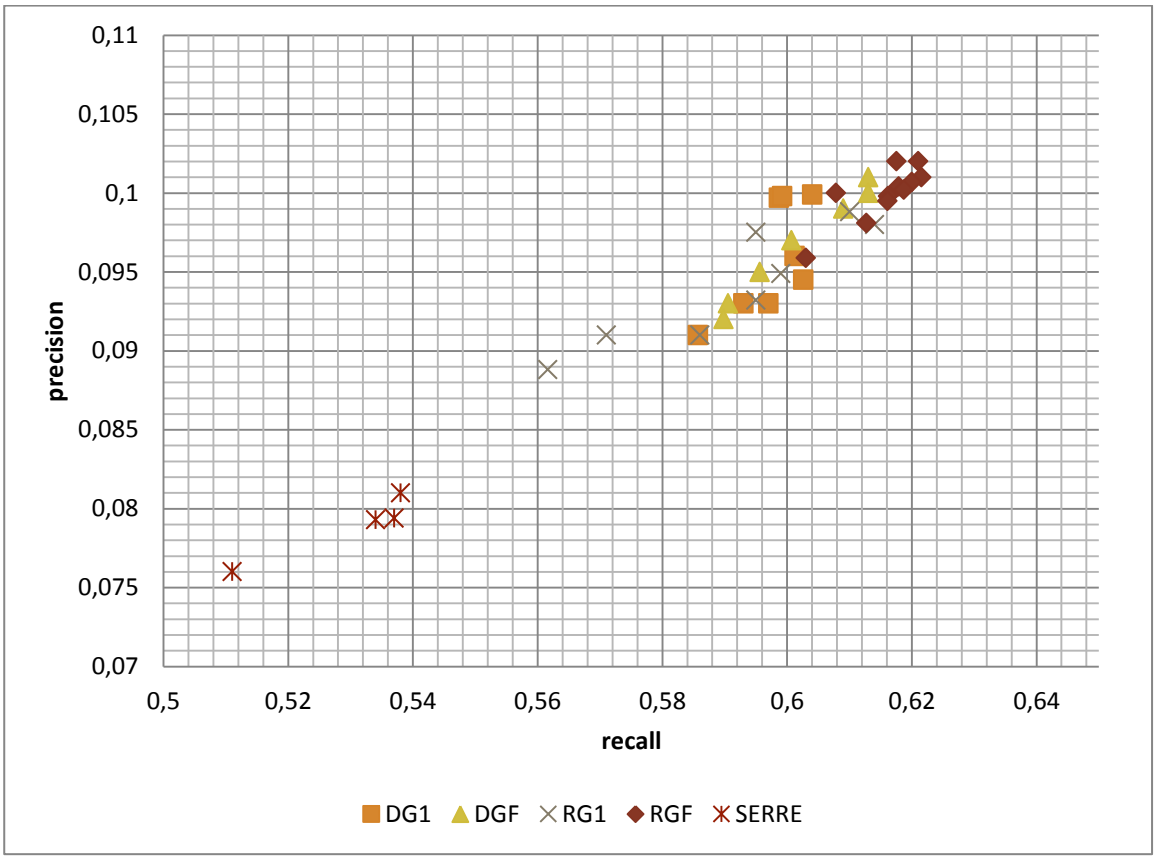


Figura V-10: Gráfico común de resultados para todos los modelos de cortex

Se ve claramente que el *modelo RGF* obtiene la mayor parte de sus puntos en la esquina superior derecha, por lo que es el mejor modelo. Adicionalmente, se puede agregar toda la información específica del recall en una tabla numérica, mostrando el recall medio de cada modelo. Dicha información se presenta en la Tabla V-12.

Tabla V-12: Recall medio para la integración del estándar MPEG7-EHD y los cinco modelos de córtex propuestos

Modelo de córtex	SERRE	DG1	DGF	RG1	RGF
Recall medio	0.530	0.597	0.601	0.591	**0.616**

En general se ve que todos los modelos tienen un recall peor que el *Modelo RGF* que fue el seleccionado en el capítulo IV. En ese capítulo, seleccionamos dicho modelo por tener

la mejor respuesta individual de una neurona, pero con las pruebas realizadas con todo el modelo de córtex sobre una base de datos de imágenes complejas se valida y verifica que la selección del modelo neuronal ha sido la correcta.

V.6 Resultados finales y conclusiones

A lo largo del capítulo IV se ha descrito cómo se ha generado un modelo de neuronas simples de la capa 2/3 del córtex visual V1 de un macaco, capaz de detectar orientaciones en la textura de las imágenes. A la vez, se ha visto cómo el descriptor de bordes EHD del estándar MPEG7 recomienda la utilización de un conjunto de filtros manualmente diseñados para la detección de estos bordes en imágenes.

En este capítulo se ha mostrado cómo hemos podido integrar diferentes modelos de neuronas en el estándar MPEG7-EHD, y se ha visto que el modelo que propusimos y seleccionamos en el capítulo IV es el que mejores resultados genera sobre una base de datos como SAIAPR-TC12.

Una vez se ha integrado y validado el modelo de córtex en el descriptor MPEG7-EHD, toca evaluarlo en su conjunto. Para ello, primero se debe comparar su precision y recall con el propio estándar MPEG7-EHD y segundo se deben comparar ambos resultados con el mejor modelo de córtex del estado del arte del campo de la visión artificial, también integrado en MPEG7; siendo este último modelo el *SERRE* (ver capítulos II y IV) que hereda del modelo propuesto en [SERR07]. Esta comparativa se presenta en la Tabla V-13.

Tabla V-13: Resultados comparativos entre el estándar (izquierda), el estado del arte (centro) y el modelo propuesto (derecha)

	Estándar MPEG7-EHD	**MPEG7-EHD + Modelo *SERRE* [SERR07]**	**MPEG7-EHD + Modelo *RGF***
Precision	0,094 ± 0,002	0,081 ± 0,001	**0,101** ± 0,002
Recall	0,589 ± 0,007	0,538 ± 0,007	**0,622** ± 0,007

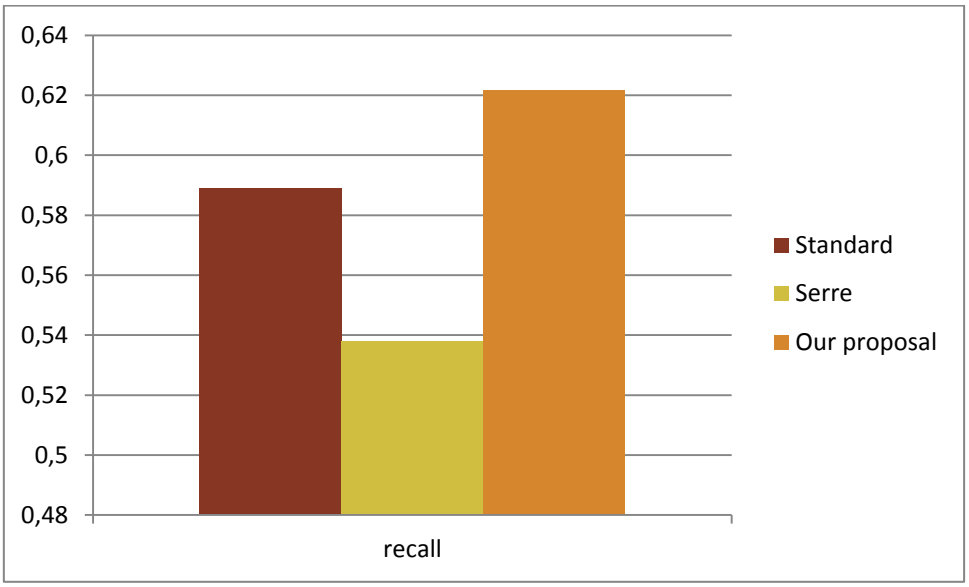


Figura V-11: Visualización gráfica del *recall* medio para las imágenes

En la Figura V-11 y la Tabla V-13 se comprueba que sobre esta base de datos la mejora de la nueva implementación del estándar propuesto en esta tesis es del 3,3%, y un resultado cercano al 10% mejor sobre el estado del arte actual en la aplicación de modelos de neuronas en visión artificial (*Modelo SERRE* basado en [SERR07]).

La evaluación anterior se ha realizado en base a imágenes, buscando que la anotación final de una imagen tenga un recall alto y, por tanto, que tras la búsqueda de imágenes similares las etiquetas de éstas posean el mayor número de etiquetas existentes en el *ground truth*.

En el estado del arte, también se evalúa este tipo de anotaciones en base a las propias etiquetas y no en base a la imagen en conjunto, buscando que el recall medio de las etiquetas a lo largo de toda la base de datos sea alto. En este caso, también se produce una mejora sustancial del recall, siendo del 2,4% (0,162 vs. 0,186), obteniendo los resultados comparativos mostrados en la Figura V-12.

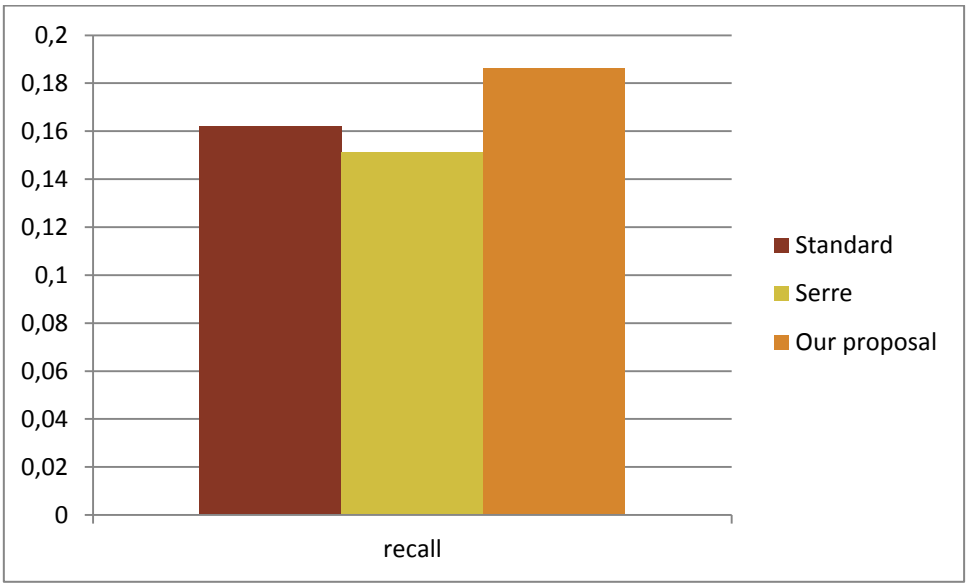


Figura V-12: Recall medio para las etiquetas de la base de datos IAPR-TC12

Además de esta prueba, se ha realizado otra utilizando una segunda base de datos de las utilizadas en el estado del arte (apartado III.1.1): la base de datos MIRFlickr con el conjunto de datos propuesto en ImageCLEF2011. Esta base de datos es totalmente diferente a la anterior, ya que tiene un conjunto mucho mayor de imágenes de prueba (~1.700 de IAPR-TC12 frente a 10.000 de MIRFlickr). Observando sólo en el recall, tanto por imagen como por etiqueta, el resultado comparativo se muestra en la Tabla V-14.

Tabla V-14: Recall del estándar y del modelo propuesto en la base de datos MIRFlickr

	Estándar MPEG7-EHD	**MPEG7-EHD con el Modelo *RGF***
Recall por imagen	0,855 ± 0,001	**0,867** ± 0,001
Recall por etiqueta	0,402 ± 0,028	**0,427** ± 0,028

Con una mejora del recall por imagen del 1,2%, una mejora por etiqueta del 2,5% y teniendo en cuenta que ningún parámetro ha sido optimizado para esta base de datos, el resultado obtenido es muy positivo.

Estas pruebas demuestran la eficacia del modelo neuronal propuesto para la detección de orientaciones en los bordes de las imágenes así como la nueva implementación del

descriptor MPEG7-EHD basado en dicho modelo. Como punto débil está el tiempo requerido para la computación: si bien al basarse en MPEG7 el descriptor es compacto, informativo y con una latencia baja en la búsqueda, el tiempo requerido para extraerlo es mucho más alto que la implementación de referencia. Futuros trabajos deberán ir encaminados a mejorar la eficiencia en la implementación, reduciendo el número de convoluciones realizadas, su implementación en plataformas más rápidas (multihilo o GPU), o su diseño en formato de filtro único.

VI. Propagación de etiquetas basada en información textual y visual

VI.1 Introducción

En esta tesis se está trabajando en proponer nuevos algoritmos que aporten mejoras a los diferentes pasos de los que se compone un sistema de anotación automática de imágenes basado en el modelo de vecinos más cercanos. En el capítulo II, se ha descrito este método como un algoritmo en dos pasos:

1. **Búsqueda de imágenes visualmente similares**, también denominadas vecinos más cercanos, donde cada imagen está acompañada de etiquetas que describen su contenido.
2. **Transferencia o propagación de las etiquetas** relevantes en base a la información de las imágenes del paso 1.

En el capítulo V se ha definido una mejora en un descriptor visual para poder obtener los vecinos más cercanos a la imagen que se pretende anotar. Para ello, se ha trabajado con el *baseline* establecido en el capítulo III y se ha propuesto una nueva implementación del descriptor MPEG7-EHD en base a un nuevo modelo de córtex de primate, propuesto en el capítulo IV, de forma que el recall obtenido es mayor que con la implementación estándar u otros algoritmos del estado del arte.

Tras esa fase, se tiene un conjunto de imágenes visualmente similares, por lo que el siguiente paso es utilizar un algoritmo de transferencia de etiquetas. En el capítulo II se ha descrito la existencia de diferentes métodos para llevar a cabo este fin. Por ejemplo, en [MAKA10] se utiliza un algoritmo denominado *GreedyTransfer* que genera las etiquetas finales seleccionándolas de la imagen más cercana, pero también seleccionándolas de otras imágenes similares en base a la frecuencia y coocurrencia de aparición de las etiquetas. Otro ejemplo es *TagProp* [GUILL09], donde se utiliza una aproximación discriminativa para cada etiqueta. Para cada modelo entrenado se le da un peso que coincide con la distancia a las imágenes similares. Además, *TagProp* también incluye una modulación sigmoidal específica para cada palabra que permite aumentar el recall de las palabras más raras.

Un trabajo reciente [FU12] ha mostrado que entrenar modelos individuales no es escalable, y propone un sistema basado en Random Forest y un esquema *tf-idf* para establecer un ranking de las etiquetas más cercanas.

En el capítulo III, también se ha comprobado cómo sencillos algoritmos mejoran algunos de los algoritmos del estado del arte, como por ejemplo el algoritmo *NearTransfer*. Además, se ha evidenciado que en los algoritmos planteados en el estado del arte no se utiliza de forma explícita toda la información existente: información visual y también textual.

En este capítulo se presenta un nuevo algoritmo que pretende solventar las limitaciones de otros sistemas de propagación de etiquetas mediante el uso de esta información de forma combinada. Para ello, se proponen dos mejoras:

1. **Algoritmo de propagación de etiquetas**: este algoritmo es el encargado de hacer la propagación propiamente dicha en base a la información visual y textual. La principal novedad de la idea será el uso de información textual y visual de forma explícita, a diferencia del estado del arte. La segunda novedad es que este algoritmo se centrará en un subconjunto del espacio y tendrá información local que permitirá optimizar el espacio de búsqueda de etiquetas a esa parte del espacio total de búsqueda. Por último, cabe destacar que este algoritmo se diseñará de forma que sea agnóstico a la tecnología de vecinos más cercanos utilizada.
2. **Algoritmo de aprendizaje eficiente**: Como todos los sistemas de propagación de etiquetas será necesario el aprendizaje de los parámetros de configuración que se deberá realizar de forma rápida y eficiente. Además, al adaptar el espacio de búsqueda a la imagen de test, será necesario hacer pequeños entrenamientos en tiempo de anotación. Este efecto hace que sea crítico un entrenamiento eficiente, tal y como el que se propone en este capítulo, para reducir el tiempo de respuesta de la anotación.

En los siguientes apartados se mostrarán en detalle ambos algoritmos así como las pruebas realizadas.

VI.2 Algoritmo de propagación de etiquetas

Dada una imagen de test que necesita ser anotada, un algoritmo de búsqueda de vecinos más cercanos obtendrá un conjunto S de K imágenes del total de imágenes de entrenamiento que son visualmente similares a la primera imagen. Cada imagen $s_i \in S$, posee un grupo reducido de etiquetas, entre las cuales se encuentran las etiquetas finales que deberán propagarse hacia la imagen de consulta. Así por ejemplo, para una imagen

de consulta *q*, se puede obtener un conjunto de *K=5* imágenes cercanas S = {s_1 , s_2 , s_3 , s_4 , s_5 }. Para cada imagen *i*, se tienen un conjunto de etiquetas $t_i = \{t_{i1}, t_{i2}, \ldots t_{iNi}\} / i=1 \ldots K$. Esto hace que para las cinco imágenes se tengan cinco conjuntos t_i y, por tanto, se tenga un conjunto T = {t_1 , t_2 , t_3 , t_4 , t_5 } de etiquetas, de tamaño $N = \sum_K N_i$, que es posible transferir a la imagen de consulta.

Para realizar esta transferencia, se propone la existencia de una función *f* que, dado este grupo de etiquetas *T*, es capaz de obtener un conjunto reducido de las mismas en base a unos parámetros de configuración *W* de la función:

$$\bar{Y}=f(T,W,q,S)$$

donde

- *T*: Conjunto de etiquetas inicial: $T = \{t_1 \cup t_2 \cup t_3 \cup t_4 \cup t_5\} / \ t_i = \{t_{i1}, t_{i2}, \ldots t_{iNi}\}$ $\forall i=1...K$
- $\bar{Y}$: Conjunto de etiquetas final: $\bar{Y} \subseteq T$
- *W*: Parámetros de configuración
- *q*: imagen de test
- S: conjunto de imágenes similares

En este punto surgen dos preguntas: ¿qué forma tiene la función *f* ?, y ¿cómo se pueden obtener los parámetros de configuración *W*?. En este apartado se trabajará sobre la función *f* y en el siguiente se mostrará la forma de obtener los parámetros de configuración *W*.

El primer paso es definir con más detalle la salida de la función *f*. $\bar{Y}$ es un vector columna que contiene un valor para cada etiqueta $t_{ij} \in T$, que idealmente será:

$$\bar{Y} = \begin{cases} \mathbf{1}, \forall t_{ij}\ relevante \\ \mathbf{0}, \forall t_{ij}\ no\ relevante \end{cases}$$

Para no forzar el entrenamiento del sistema, se va a suponer una condición más relajada, de tal forma que:

$$f(t_{ln},W,q,S)>f(t_{ij},W,q,S)$$

Para todo t_{ln} que sea más relevante para la imagen q que t_{ij}. Una vez se ha descrito de forma exacta el problema, se va a definir la siguiente función f:

$$f(T,W,q,S)=W\cdot X^{t}$$

donde:

- $W \in M_{1\times m}(\mathbb{R})$, contiene los parámetros de configuración
- $X \in M_{N\times m}(\mathbb{R})$, contiene la representación $\vec{x}_{ij} \in \mathbb{R}^m$ de cada etiqueta $t_{ij} \in T$

El objetivo de esta función de propagación es que el vector de características que representa cada etiqueta se proyecte sobre un vector W, de tal forma que todos los puntos proyectados sobre dicho vector tengan un orden sobre dicho vector.

Con esta definición de función de propagación sólo se necesita saber cómo se realiza la representación matemática de cada etiqueta disponible en T. Las etiquetas son elementos textuales que están asociados a cada una de las imágenes de la base de datos. De esta forma, no tienen una relación explícita con la imagen de entrada q. Así, no es posible usar información textual que permita a nuestra función f conocer qué etiquetas es necesario propagar.

Aun así, sí que existe una relación implícita que se propone explotar: una etiqueta está asociada a una imagen conocida, todas las imágenes conocidas tienen una representación matemática, y esa representación se puede asociar a la representación de la imagen de entrada gracias a una medida de similitud. Por ello, en la representación utilizada en la matriz X será necesario incluir toda esta información de la forma más eficiente posible.

Se va a definir una nueva función que mapea cada etiqueta a un espacio Euclídeo donde estará representada por un vector de características $\vec{x}_{ij} \in \mathbb{R}^m$. Esta función generará una representación de la etiqueta $t_{ij} \in T$ que contenga información semántica de la propia

etiqueta, así como información visual de la imagen de consulta *q* y de la imagen asociada a la etiqueta $s_i \in S$:

$$\vec{x}_{ij}=g(q,t_{ij},s_i)$$

Donde *g* denota que $\vec{x}_{ij}$ depende de la imagen de test *q*, de la imagen de entrenamiento s_i y de la etiqueta t_{ij}. Dada esta definición, si dos imágenes de entrenamiento en el set *S* se anotan con la misma etiqueta (y por tanto se introduce doblemente dicha etiqueta en el conjunto *T*), el vector $\vec{x}_{ij} \in \mathbb{R}^m$ será diferente para cada etiqueta ya que está asociada a diferentes imágenes *q* y s_i.

Así, para cada par de imagen *q* y tupla etiqueta-imagen (t_{ij}, s_i), la función de mapeo *g* generará las siguientes características:

- $\vec{x}_{ij}^{(1)}$ – *Distancia entre imágenes test-entrenamiento*: Esta característica define la distancia visual entre la imagen de test *q* y la imagen similar s_i obtenida del algoritmo de vecinos más cercanos. El objetivo de esta característica reside en que si dos imágenes son visualmente de forma global muy similares, entonces es muy probable que compartan la etiqueta.
- $\vec{x}_{ij}^{(2)}$ – *Distancia de color entre imágenes test-entrenamiento*: Como se ha visto en el capítulo II, el color juega un papel importante cuando se trata de clasificar imágenes. Esta característica mide la similitud de color entre la imagen de test *q* y la imagen similar s_i, y trata de utilizar la componente de color de forma independiente para propagar etiquetas que tienen un alto grado de correlación entre el color y la etiqueta, por ejemplo, el color verde y la etiqueta “hierba” como se ilustra en la Figura VI-1.

Figura VI-1: Algunas imágenes del synset "grass" de la base de datos ImageNet [DENG09]

- $\vec{x}_{ij}^{(3)}$ – *Distancia de bordes entre imágenes test-entrenamiento*: Los bordes existentes en las imágenes también son importantes durante la clasificación de imágenes. Es más, en el capítulo IV se ha visto cómo la función principal del córtex de los primates trata de representar los bordes y orientaciones de una forma óptima. Por ello, esta característica mide la similitud entre los bordes existentes en la imagen de test *q* y la imagen similar s_i, y al igual que en el caso anterior trata de fijar la propagación de las etiquetas para los casos en los que los bordes son muy importantes, como por ejemplo, la existencia de muchos bordes "aleatorios" y el concepto "árbol".
- $\vec{x}_{ij}^{(4)}$ – *Coocurrencia de etiquetas:* Esta característica semántica mide el número de veces que una etiqueta t_{ij} aparece en el conjunto de entrenamiento con el resto de etiquetas que pertenecen a *T.* Esta característica es ampliamente utilizada en el mundo del procesamiento del lenguaje natural, y su objetivo es que si hay ciertas etiquetas que normalmente aparecen juntas y alguna de ellas está presente en el conjunto de salida del algoritmo de vecinos más cercanos, quiere decir que es muy probable que las otras sean relevantes.
- $\vec{x}_{ij}^{(5)}$ – *Frecuencia de etiqueta:* Esta segunda característica semántica mide el número de veces que la etiqueta t_{ij} aparece en el conjunto *T*. De esta forma, se mide si varias de las imágenes similares están incluyendo la etiqueta para su

propagación y, por lo tanto, habría que tenerla más en cuenta, ya que hay varias imágenes que apoyan la "decisión".

- $\vec{x}_{ij}^{(6)}$ – *Probabilidad de etiqueta:* Esta característica mide con qué probabilidad aparece una etiqueta en la base de datos. Su función es tratar de modular el resto de informaciones. Así, si una etiqueta es muy probable que aparezca entonces el efecto de las dos características anteriores se debe potenciar.

El razonamiento de este vector de características, definido a través de diferentes conjuntos de pruebas, es que el algoritmo de aprendizaje que se propone deberá aprender la relación entre las características visuales de las imágenes de entrenamiento y de test (características $\vec{x}^{(1-3)}$), conjuntamente con la información de sus etiquetas (características $\vec{x}^{(4-6)}$).

VI.3 Aprendizaje de los parámetros de configuración

Retomando la formulación inicial del problema, se busca obtener un conjunto de etiquetas afines para una imagen de entrada, por medio de una función de este tipo:

$$\bar{Y} = f(T, W, q, S)$$

En el apartado anterior se ha definido como:

$$\bar{Y} = W \cdot X^t$$

De la anterior función, ya se ha descrito el procedimiento de mapeo de las etiquetas hacia la matriz de características *X*, por lo que es necesario definir la forma de aprendizaje de la matriz *W*.

Por norma general, los parámetros de configuración se obtendrán de un entrenamiento previo del sistema, usando alguna técnica de aprendizaje del estado del arte. Una aproximación lógica es que estos parámetros de configuración se obtengan entrenando el sistema con la base de datos de imágenes completa, y se podría realizar la suposición de

que dichos parámetros son válidos para la imagen de consulta. Pero esto no siempre tiene porqué ser cierto.

Supongamos un espacio en 2 dimensiones en el que las imágenes están representadas por un vector $z \in \mathbb{R}^2$ (Figura VI-2).

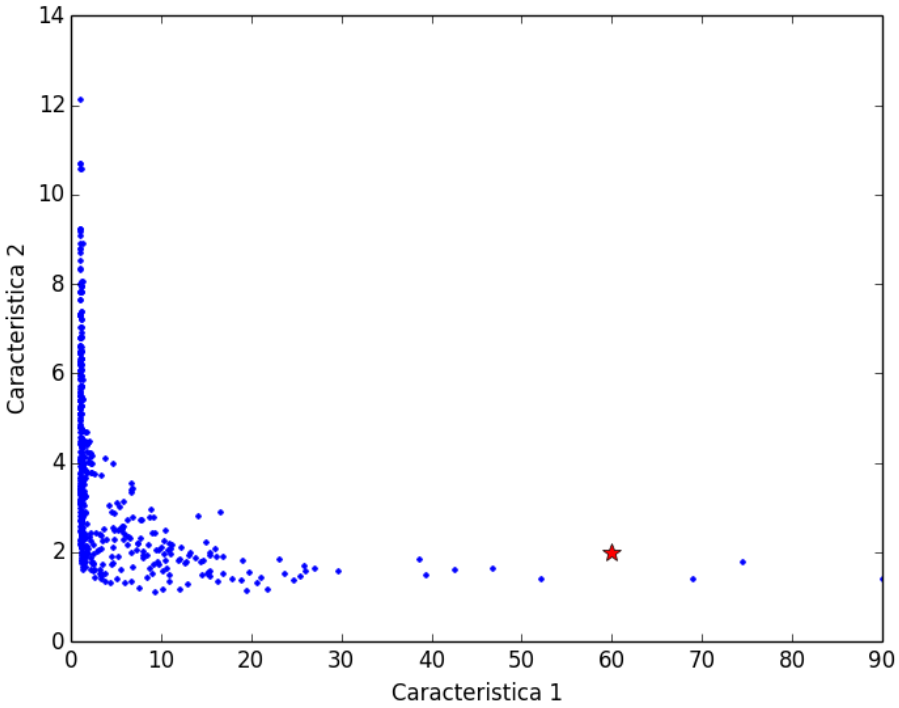


Figura VI-2: Espacio bidimensional en el que existe nuestra base de datos (azul) y nuestra imagen de consulta (rojo)

Si nuestra imagen de consulta está localizada en una "esquina" de este espacio $\mathbb{R}^2$, los parámetros de configuración contendrán información sobre el total del espacio $\mathbb{R}^2$. La lógica dice que si estos parámetros *W* están optimizados para esta "esquina" del espacio, entonces su información será mucho más adecuada a la imagen a anotar.

Para obtener los parámetros *W* se puede dividir el espacio en zonas y entrenarlo de forma específica para cada una de ellas. Un caso directo es dividir el espacio bidimensional en forma de cuadrícula, tal y como se muestra en la Figura VI-3.

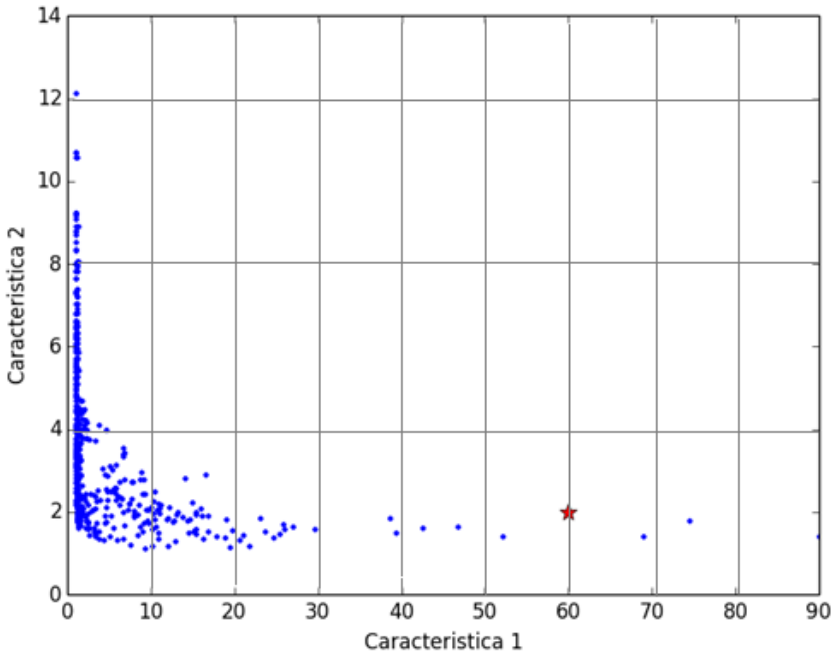


Figura VI-3: División homogénea del espacio bidimensional

Para cada parte del espacio podríamos tener unos parámetros *W*, pero existe el problema de imágenes que se encuentren muy cerca de los límites, y puede que no se relacionen correctamente con sus parámetros de configuración. Un problema más importante es que mediante esta técnica se produce el efecto no deseado de separar imágenes cercanas y, por tanto, visualmente similares; ya que pertenecerán a dos regiones con parámetros de configuración diferentes.

Debido a estos problemas, la propuesta que se hace en esta tesis es obtener estos parámetros lo más adaptados posible a la imagen de entrada. Para ello, se propone aprender la matriz de configuración *W* en tiempo de anotación. Entonces, durante la consulta se tendría la representación vectorial *x* de la imagen a anotar en el espacio $\mathbb{R}^2$, y se obtendrían estos parámetros de configuración para la propagación de etiquetas.

Visualmente, esta aproximación separaría el espacio de características en dos conjuntos, uno conteniendo imágenes que potencialmente tienen información útil para obtener los parámetros de configuración, y otra que tendrá menos información útil. Un ejemplo representativo en nuestro espacio bidimensional se muestra en la Figura VI-4.

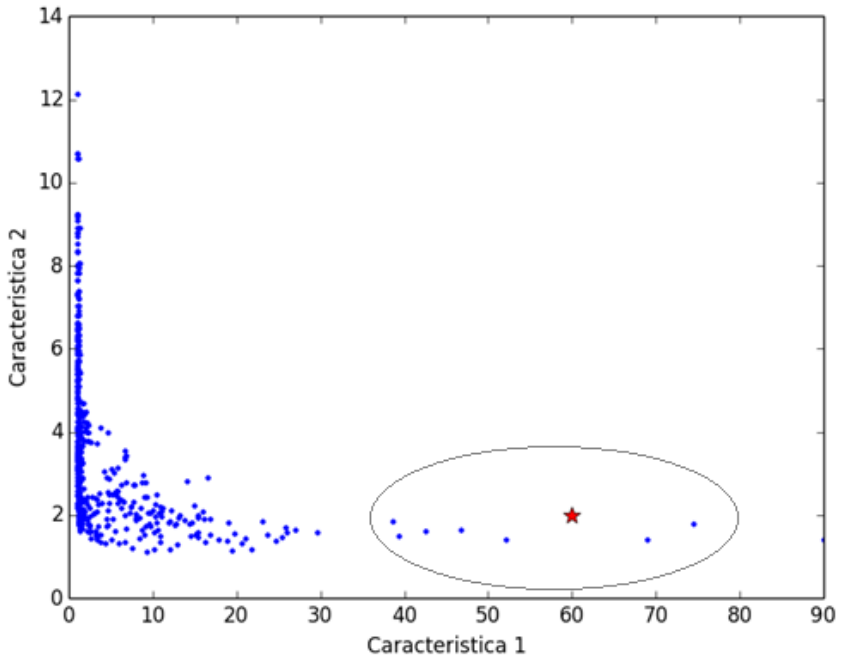


Figura VI-4: División del espacio adaptada a la consulta

A diferencia de otras ideas propuestas en la literatura, donde se entrena una única matriz *W* para todo el espacio, en este trabajo se propone la obtención de dicha matriz de pesos de forma local y adaptada a cada imagen de búsqueda.

Para la obtención de *W* pueden existir numerosas aproximaciones, pero en esta tesis se propone utilizar un nuevo concepto específico para este tipo de problemas relacionados con imágenes que hemos denominado *Image Networking*, cuya esencia es que todas las imágenes están “conectadas” (Figura VI-5). El objetivo de este método es obtener, en tiempo de consulta, los parámetros de configuración de la función de propagación de etiquetas.

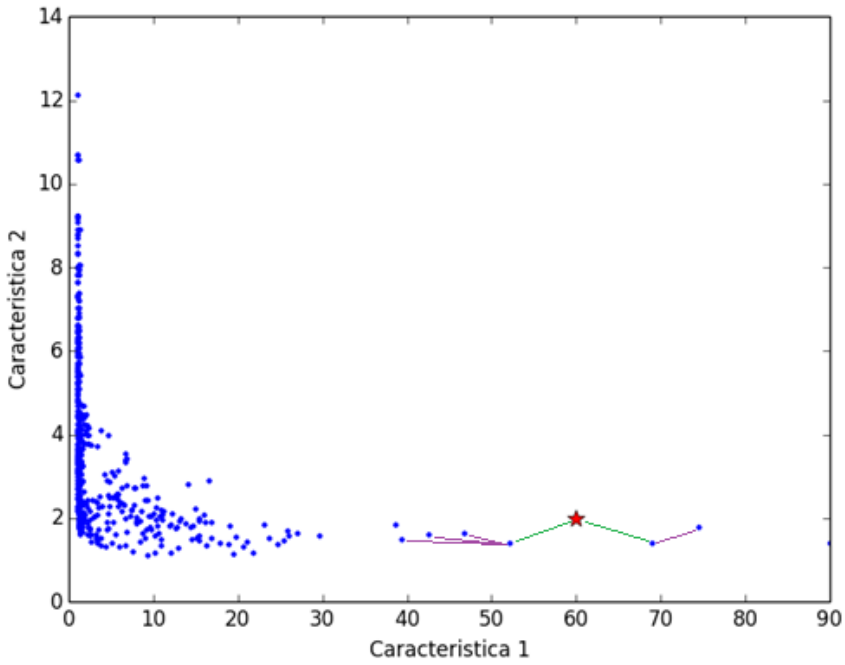


Figura VI-5: *ImageNetworking* en funcionamiento: la imagen de query está relacionada con dos imágenes de la base de datos y éstas a su vez con otras cuatro

Ante una imagen de consulta *q*, se dispone de un conjunto *S* con *K* imágenes similares de la base de datos de entrenamiento, y para cada una de esas imágenes $s_i \in S \ / i \in (1, \ldots, K)$ conocemos sus etiquetas reales t_i. Para cada imagen similar *i* podremos obtener su conjunto de imágenes similares $S^{(i)}$ que contiene $K^{(i)}$ imágenes similares, es decir, sus imágenes "padre". Además, dado $S^{(i)}$ también se puede extraer su conjunto correspondiente de etiquetas $T^{(i)}$. Usando la definición inicial del algoritmo de propagación de etiquetas tenemos lo siguiente:

$$\bar{Y}^{(i)} = f\left(T^{(i)}, W, s_i, S^{(i)}\right)$$

donde $\bar{Y}^{(i)}$ es el conjunto de etiquetas que se van a propagar a la imagen de entrada, en este caso s_i. Como se ha dicho anteriormente, s_i es una imagen de entrenamiento por lo que conocemos sus etiquetas reales t_i. Por ello, conocemos la correspondencia de la salida anterior $\bar{Y}^{(i)} = T_i$, donde T_i contendrá el valor 1 para las etiquetas de t_i, y el valor 0 para el resto.

Así, la función para el entrenamiento queda configurada de la siguiente forma:

$$T_i = f\left(T^{(i)}, W, s_i, S^{(i)}\right) \mid i = 1 \dots K$$

Debido a la restricción impuesta con anterioridad de la forma $f(t_{ln},W,q,S)>f(t_{ij},W,q,S)$ se busca una matriz W de tal forma que para todas las imágenes s_i se cumplan el mayor número de inecuaciones posible. Este problema es NP-Hard [JOAC02], pero se puede resolver relajando las inecuaciones mediante la introducción de *slack-variables* $\xi_{l,n,i,j}$ en las inecuaciones de la siguiente forma:

$$f(t_{ln}, W, q, S) > f(t_{ij}, W, q, S) + 1 - \xi_{l,n,i,j}$$

Gracias a ellas, el problema de optimización que se busca no es tan estricto, sino que se permite un cierto margen de error para que se cumpla la inecuación. En términos técnicos, se permiten violaciones de las restricciones anteriores, aunque posteriormente estas violaciones se penalizarán en la función de optimización, procedimiento comúnmente llamado *soft-margin SVM*.

Sustituyendo f por la definición que hemos propuesto, las restricciones quedarían de la siguiente forma:

$$W\cdot\left(g(q,t_{ln},s_l)-g\left(q,t_{ij},s_i\right)\right)>1-\xi_{l,n,i,j}$$

En este caso, ya es posible obtener W. Para ello, de forma similar a una SVM clásica, se utilizará una función de optimización con un regularizador $\|W\|^2$ para la maximización del margen de clasificación. Si este regularizador es pequeño, entonces la función f (dependiente de W) variará de forma suave, permitiendo resultados más precisos. Por otro lado, también es necesario introducir el sumatorio de las *slack-variables* para minimizar su límite superior. El problema convexo de optimización a resolver es el siguiente y es similar al planteado en otros trabajos como [JOAC02]:

$$\min_{w,\xi_{l,i,k}\geq 0}\left\{\frac{1}{2}\|W\|^2 + C\sum_i \xi_{l,n,i,j}\right\}$$

s.t. $W\cdot\left(g(q,t_{ln},s_l)-g\left(q,t_{ij},s_i\right)\right)>1-\xi_{l,n,i,j}$ para todo par $((t_{ij},s_i),(t_{ln},s_l))$

Como se puede ver, esta configuración está preparada para permitir un mayor o menor grado de error de entrenamiento, mediante la modificación del parámetro *C* que actúa sobre el sumatorio de los márgenes introducidos por las *slack variables*.

El problema de optimización anterior es muy similar al que es necesario resolver para la optimización de un regresor basado en Máquinas de Vectores Soporte, y por lo tanto, se puede utilizar cualquier optimizador estándar para su resolución.

A pesar de ello, **si el número de elementos de entrenamiento es alto se plantea un problema que no es posible manejar en un tiempo limitado** por los optimizadores estándar [JOAC07]. Por ello, una solución es plantear este problema desde el punto de vista de resolución mediante un SVM de salidas estructuradas (SVM-Struct), y utilizando un algoritmo eficiente de resolución.

SVM-Struct ha sido planteado como forma de resolver tareas de predicción de objetos estructurados. A diferencia de un SVM clásico, en el que las salidas son binarias, en SVM-Struct la salida puede ser de cualquier tipo, y en el caso específico actual, se plantea una salida en la que se indiquen afinidades de etiquetas a una imagen. El primer problema que resuelve SVM-Struct es la representación de espacios de soluciones muy grandes. Para una salida binaria, las soluciones posibles son *y={1,0}*, pero para una salida como la que se busca en esta tarea puede tener valores infinitos. Con la formulación anterior, se encuentra el problema que resolviéndolo mediante un SVM clásico, el valor *W* se generalizará para cualquier entrada *x*. Pero nuestra salida es mucho mayor que dos valores binarios y es necesario que *W* no dependa del número de posibilidades de *y*, y generalice también para ella.

De esta forma, para resolver este problema, SVM-Struct propone el uso del denominado *joint feature map* $\Psi(x,y)$ en vez de usar directamente la función de mapeo $g(x)$. La diferencia fundamental es que la función $g(x)$ sólo depende de las imágenes, mientras que $\Psi(x,y)$ también depende del resultado esperado a la salida. Esta nueva función realiza un mapeo de la combinación de entrada salida de tal manera que la función

resultado queda de la forma $f(x,y) = w \cdot \Psi(x,y)$y no depende directamente del número de posibles soluciones existentes en el espacio de salida, sino que depende de las características de $\Psi(x,y)$.

Otro elemento a introducir en la formulación del problema es la *loss function* $\Delta(y,\bar{y})$.Al haber introducido las *slack-variables* estamos permitiendo la existencia de un cierto error en la salida para poder obtener un problema de optimización convexa. Pero en ningún momento se cuantifica si la calidad de la solución es buena, es decir, lo predicho con error es similar o no a la salida esperada. Por ello, se propone la introducción de esta función que permite medir esta calidad.

Usando ambos elementos introducidos, se plantea la siguiente reformulación (en forma de *quadratic program)* de nuestra tarea de optimización:

$$\min_{w,\xi_{i,j}\geq 0}\left\{\frac{1}{2}\|W\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_{i,j}\right\}$$

s.t. $W \cdot \Psi(x_i, y_i) - W \cdot \Psi\left(x_j, \bar{y}_j\right) > \Delta(y,\bar{y}) - \xi_{i,j}$

En la anterior reformulación se han descartado todas las dependencias con los las etiquetas *T* e imágenes *S*, y se ha utilizado la denominación genérica *i,j* para facilitar la lectura.

Con esta reformulación, sólo falta definir una forma óptima de resolverla, así como definir $\Psi(x,y)$ para nuestro problema de propagación de etiquetas.

Este problema se puede resolver de forma eficiente usando el algoritmo *cutting-plane* propuesto en [JOAC07]. La idea básica de este método se basa en utilizar un subconjunto de las restricciones anteriores en vez de todas al mismo tiempo. Iterativamente, el algoritmo construirá un conjunto de restricciones para trabajar con ellas y que sean equivalentes al conjunto total, hasta una precisión *e*. Para construir este set, se analizará la restricción para cada uno de los elementos de entrenamiento (x_i, y_i) y se verá si la restricción se supera en un valor *e*. De forma matemática, la comprobación es la siguiente:

$$W \cdot \Psi(x_i, y_i) - W \cdot \Psi\left(x_j, \bar{y}_j\right) > \Delta(y, \bar{y}) - \xi_{i,j} - e$$

Si esta restricción es sobrepasada, entonces se incluye en el conjunto de trabajo. Evidentemente, al ser un método iterativo, el punto de parada es cuando el conjunto de restricciones de trabajo no cambia en un número determinado de iteraciones.

Este número de iteraciones es polinomial e independiente de la cardinalidad del espacio de salida [JOAC09], lo que hace del método muy eficiente para salidas con gran número de dimensiones. Además, también es independiente del número de elementos de entrenamiento, por lo que más elementos de entrenamiento no hacen aumentar significativamente el tiempo de entrenamiento.

Por otra parte, está la definición de la función $\Psi(x, y)$ para la tarea de propagación de etiquetas planteada. La definición propuesta en esta tesis es la siguiente, eliminando las dependencias con las etiquetas *T* y las imágenes *S* para facilitar la lectura:

$$\Psi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{ij} y_{ij} \langle\, g(x_i) - \, g\left(x_j\right), \boldsymbol{W} \rangle$$

Teniendo en cuenta que $y_{ij}=1$ si x_i es más relevante que x_j. Gracias a esta función, los resultados generados por *f* se ordenarán de menor a mayor puntuación y_{ij}.

VI.4 Pruebas y resultados

Tras definir el algoritmo de propagación de etiquetas y su entrenamiento, en este apartado se detallarán las pruebas realizadas para dar validez a los mismos. Para ello, se utilizará el *baseline* definido en el capítulo III, modificando únicamente el algoritmo de propagación de etiquetas.

Como se ha visto anteriormente, es necesario tener una función de mapeo $g(x)$ entre una imagen de consulta *q* y las etiquetas de las imágenes similares a un espacio de características.

Para este mapeo se han seleccionado varias características, que son las siguientes:

- Características visuales
  - Distancia L1 normalizada del vector de características SCD+EHD de la imagen de consulta al vector de la imagen de la base de datos de entrenamiento.
  - Distancia L1 normalizada del descriptor SCD entre la imagen de consulta y la de entrenamiento.
  - Distancia L1 normalizada del descriptor EHD entre la imagen de consulta y la de entrenamiento.
- Características textuales
  - Coocurrencia de la etiqueta.
  - Frecuencia de aparición de la etiqueta en los KNN.
  - Probabilidad de aparición de la etiqueta en la base de datos.

Así, para una imagen de consulta, se obtendrán los K vecinos más cercanos, donde K es 10, computándolos mediante la técnica de K-NN propuesta en el *baseline*. Para cada etiqueta que se encuentra en cada una de las K imágenes, se genera un vector que contenga las 6 características anteriores.

En cuanto a las pruebas, éstas se han realizado en las dos bases de datos que se están usando en esta tesis y en el estado del arte: SAIAPR-TC12 y MIRFlickr. En los siguientes apartados se muestran los resultados de las mismas.

VI.4.1 Base de datos SAIAPR-TC12

El primer experimento a realizar es la verificación de la suposición realizada en apartados anteriores: el uso conjunto de información textual y visual para la transferencia de etiquetas es mejor que el uso individual de las mismas.

Para ello, construimos nuestro primer experimento sobre la base de datos SAIAPR-TC12, mostrando en la Figura VI-6 la precisión en la anotación de imágenes para diferentes tamaños de vecindarios, y comparando el algoritmo propuesto, en el que se usa información visual y textual (vector $\vec{x}_{i,j}^{(1\text{-}6)}$), información sólo visual (vector $\vec{x}_{i,j}^{(1\text{-}3)}$) e información sólo textual (vector $\vec{x}_{i,j}^{(4\text{-}6)}$).

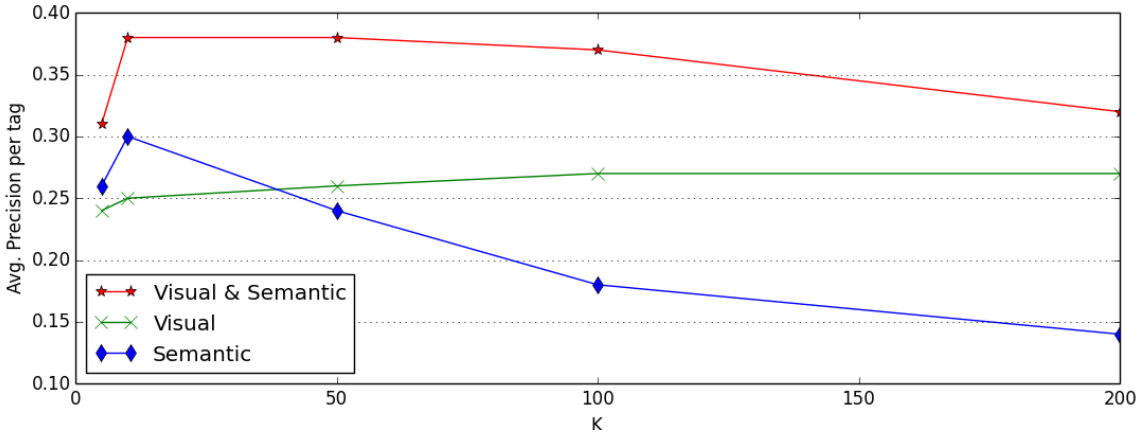


Figura VI-6: Precision media para tres diferentes configuraciones del algoritmo propuesto para diferentes tamaños de vecindario

Este experimento muestra cómo el algoritmo propuesto obtiene un mayor AP con la combinación de información que con las informaciones individuales. Además los mejores resultados se obtienen para vecindarios pequeños, lo que corrobora el hecho de que un vecindario pequeño con información local aprende mejor los parámetros de configuración y adaptación a la imagen de entrada que el uso de toda la base de datos, representado en la anterior imagen como un valor de vecindario grande.

La base de datos SAIAPR-TC12 ha sido históricamente una de las más usadas en el estado del arte de la anotación de imágenes multi-etiqueta. Por ello, también se ha seleccionado para mostrar y comparar los resultados del algoritmo propuesto frente al estado del arte. Para la primera comparativa sobre esta base de datos se han seleccionado tres modelos de propagación, usando la representación visual y la búsqueda de vecinos más cercanos del *baseline* propuesto en el capítulo III. En cuanto a la comparativa, se han testeado los algoritmos más relevantes definidos en el propio capítulo, que son el algoritmo *NearTransfer* y el *GreedyTransfer*. Además, se han testeado dos variantes de nuestra propuesta: un algoritmo con un número de etiquetas de salida fija e igual a 5 y otra con un número de etiquetas optimizado en función de las probabilidades obtenidas. Sus resultados se muestran en la Tabla VI-1, donde la configuración es común con 5 etiquetas de salida para cada imagen anotada.

Tabla VI-1: Comparativa del algoritmo propuesto con respecto a la base del capítulo III

Algoritmo	Precision	Recall	F-measure
MPEG7 features - NearTransfer	0,29	0,29	0,29
MPEG7 features - GreedyTransfer	0,31	0,21	0,25
Proposed algorithm (C=1.0,K=50)	0,40	0,15	0,22
Proposed algorithm + optimal number of tags	0,42	0,22	0,29

En esta fase del sistema de anotación, la precisión es más importante que el recall [TSAI11], por lo que es éste el valor de referencia. Se ve claramente que el resultado es muy superior a la base establecida.

Debido a que el sistema trata de anotar de la forma más precisa posible, se han centrado una serie de experimentos en conocer la precisión del algoritmo propuesto para una serie de etiquetas. Este valor de medida se conoce como “precisión a *X* etiquetas” y evaluar el precision obtenido al mostrar al usuario esas *X* etiquetas. En la Figura VI-7 se muestra un gráfico del precision evaluado a un mismo número de etiquetas (de 1 a 30) sobre los tres algoritmos seleccionados en la Tabla VI-1.

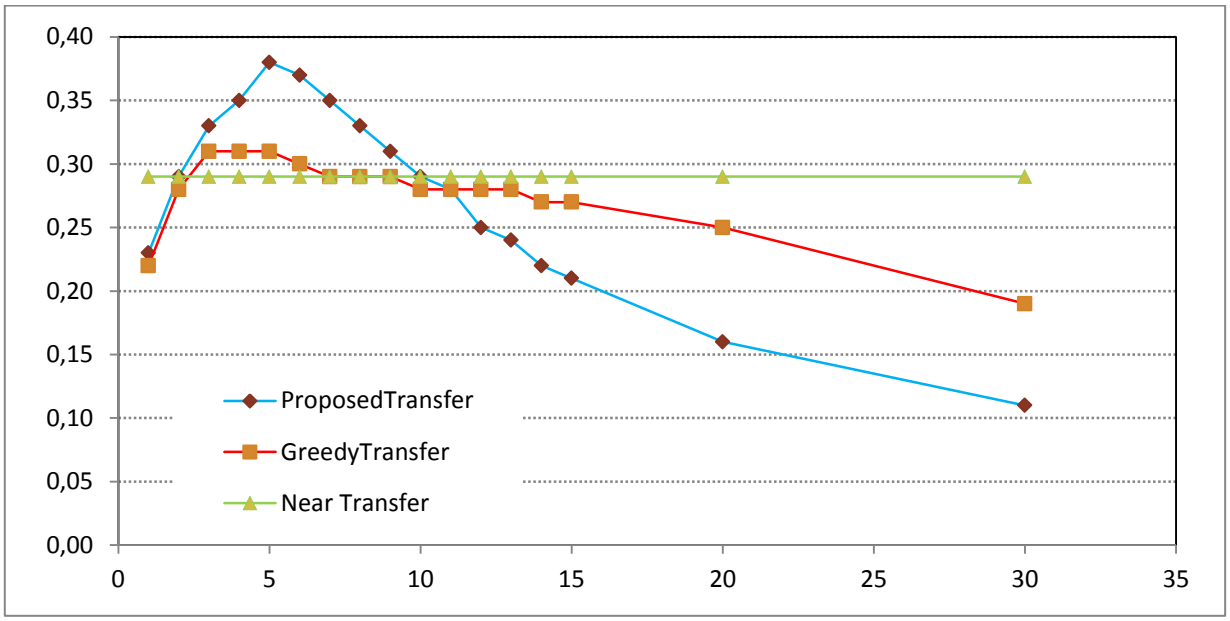


Figura VI-7: Curva de precision@ para un número de etiquetas desde 1 hasta 30

En la Figura VI-7 se ve cómo la propuesta realizada posee un pico de precisión cuando el número de etiquetas es pequeño mientras que el resto de algoritmos empeoran o se mantienen igual. Esto se debe al correcto funcionamiento de la parte de entrenamiento local.

Para ver el resultado del sistema de forma cualitativa, a continuación se presentan algunas imágenes con las etiquetas de resultado, con valores de probabilidad normalizados al valor máximo. Las etiquetas subrayadas se encuentran en el *ground truth*.

Tabla VI-2: Resultados de la anotación del algoritmo propuesto

Imagen	Etiquetas predichas
	Wall, room, front, table
	Tree, sky, house, front, building, people, man, woman, wall, roof
	Sky, tree, people, mountain, house, building, cloud, street
	Sky, mountain, cloud, tree, landscape, man, house, lake

Se ve que tanto en paisajes rurales como urbanos el resultado es bastante positivo, aunque existen problemas con la detección de objetos concretos.

Como se ha visto en el capítulo III, sobre esta base de datos se han testeado numerosos algoritmos del estado del arte. La pregunta en este punto es si la respuesta de la

propuesta desarrollada en comparación a estos algoritmos es mejor o no. Para ello, en la Tabla VI-3 se muestran los resultados más relevantes de los algoritmos testeados en esta base de datos en los últimos 4 años, utilizando los valores reportados en los artículos de investigación donde se describían los trabajos realizados. En el caso de *ImageNetworking* se están usando los parámetros de configuración K=10, C=20 con para el algoritmo con L=5 etiquetas de salida; y K=50, C=1 para el algoritmo con un número optimizado de etiquetas.

Tabla VI-3: Resultados de los algoritmos del estado del arte sobre la base de datos SAIAPR TC-12. Los resultados se miden con precision (P), recall (R), F-measure (F), y número de etiquetas con recall positive (N+)

			P	R	F	N+
Naïve Propagation Algorithms	*Near Transfer*		0.29	0.29	0.290	255
	Frequency Transfer		0.28	0.17	0.212	189
Previously Reported Results	*MBRM [FENG04]*		0.21	0.14	0.168	186
	JEC + Lasso [MAKA10]		0.26	0.16	0.198	199
	JEC + Greedy Transfer [MAKA10]		0.25	0.16	0.195	196
	SML+RF [FU12]		0.27	0.30	0.284	266
	RF_count^2 [FU12]		0.45	0.31	0.367	253
	*Group Sparsity [*ZHAN10*]*		0.32	0.29	0.304	252
	TagProp SD [GUILL09]		0.50	0.20	0.286	215
	TagProp sigmaSD [GUILL09]		0.41	0.30	0.346	259
	TagProp sigmaML [GUILL09]		0.46	0.35	0.398	266
Our Approach	*K=10*	*Fixed L=5*	0.38	0.18	0.244	210
		Optimized L	0.37	**0.29**	**0.325**	240
	K=50	*Fixed L=5*	0.40	0.15	0.22	231
		Optimized L	**0.42**	0.22	0.29	252

Como se ha dicho, los resultados mostrados en este punto son los reportados en los trabajos publicados. Es necesario recordar que en cada trabajo se ha modificado todo el algoritmo de anotación y, por tanto, la representación de la imagen es diferente, la

búsqueda de imágenes similares es diferente y el propio algoritmo de propagación de etiquetas es diferente. En esta situación, se ve cómo dos trabajos son superiores a los resultados mostrados en esta tesis, y son TagProp [GUILL09] y RF-count [FU12]. Pero la comparativa de nuestro algoritmo no puede ser realizada de forma directa contra ninguno de estos algoritmos. La problemática de esta comparativa es que el algoritmo descrito en esta tesis está basado en el trabajo de Makadia et al. [MAKA10] y sólo se puede comparar con él. Con respecto a ese trabajo, se tiene una ganancia del 13% en la precisión (para condiciones comunes de K=10).

Para tener una mejor visión de si la propuesta es adecuada o no, es necesario establecer un algoritmo de búsqueda de vecinos más cercanos común, y probar y comparar únicamente el algoritmo de transferencia de etiquetas. Esta prueba es la que se presenta en el siguiente apartado con una base de datos que no ha sido usada por ninguno de esos algoritmos.

VI.4.2 Base de datos MIRFlickr – ImageCLEF2011

La base de datos MIRFlickr es una base de datos que posee una característica idónea para poder testear los sistemas que se proponen: posee un gran número de imágenes de test.

Al igual que en el caso anterior, sobre esta base de datos también se ha propuesto una comparativa de algoritmo. En la primera prueba realizada sobre esta base de datos se han comparado tres modelos de propagación, usando como representación visual la misma propuesta que en el capítulo III. En cuanto a la comparativa, se han testeado los tres algoritmos más relevantes a la vista del capítulo III, capítulo II y la Tabla VI-1, que son los siguientes:

- El algoritmo propuesto en este capítulo VI.
- El algoritmo *NearTransfer*, que fue seleccionado en el capítulo III como mejor *baseline*.
- Un modelo discriminativo básico, ya que la mayor parte del estado del arte se centra en dichos modelos (capítulo II). Para ello, se han entrenado SVMs lineales para cada etiqueta de la base de datos.

Para estos tres algoritmos, los resultados de MAP, EER y AUC se presentan en la Tabla VI-4.

Tabla VI-4: Resultados de la base de datos ImageCLEF 2011

Algoritmo	MAP	EER	AUC
Proposed algorithm	**0.205**	**0.416**	**0.451**
MPEG7 features - NearTransfer	0.142	0.442	0.429
Concept SVM (lineal-real)	0.194	0.395	0.640

En la Tabla VI-4 se ve cómo el algoritmo propuesto obtiene resultados superiores a otros modelos base seleccionado en el capítulo III. Incidiendo más en los resultados obtenidos por nuestro algoritmo, también se ha realizado un análisis sobre cada uno de los conceptos testeados. Las etiquetas que han tenido un AP medio mayor son las mostradas en la Tabla VI-5.

Tabla VI-5: Mejores etiquetas de la base de datos de ImageCLEF 2011

Concepto	AP	EER	AUC
Neutral_Illumination	0.955	0.455	0.218
No_Persons	0.801	0.370	0.670
No_Blur	0.755	0.425	0.608
Day	0.748	0.32	0.742

En cuanto a los peores valores, se obtienen para las etiquetas que se muestran en la Tabla VI-6.

Tabla VI-6: Algunos de los conceptos con peor resultado obtenido en la base de datos de ImageCLEF 2011

Concepto	AP	EER	AUC
Skateboard	0.005	0.33	0.17
Rain	0.003	0.53	0.00
Birthday	0.008	0.54	0.14
Horse	0.006	0.49	0.03

Se observa la misma tendencia que en la base de datos SAIAPR-TC12, donde la información global de la imagen se representa mucho mejor que los conceptos locales como objetos.

Al igual que se ha hecho en el apartado VI.4.1 con la base de datos SAIAPR-TC12, en la

Tabla VI-7 se muestran algunas anotaciones de imágenes de ejemplo, donde las etiquetas subrayadas pertenecen al *ground truth*.

Tabla VI-7: Resultados de las anotaciones producidas por el algoritmo propuesto sobre la base de datos MIRFlickr

Imágenes	Anotaciones generadas
	Neutral_Illumination, No_Persons, No_Blur, natural, Day, Outdoor, cute
	Neutral_Illumination, No_Persons, No_Blur, Outdoor, natural, Day, cute, Visual_Arts, Sky
	Neutral_Illumination, Outdoor, Day, No_Persons, natural, No_Blur, cute, Plants, Visual_Arts, Sky, Landscape_Nature
	Neutral_Illumination, No_Blur, male, cute, natural, Visual_Arts, Single_Person, Adult

Una vez probado el algoritmo contra la base de datos y comparado con los algoritmos base, es hora de hacer la comparativa con respecto al estado del arte. Con este objetivo en mente, se ha utilizado una misma forma de computar las imágenes similares, utilizando el *baseline* propuesto y se ha ejecutado únicamente la etapa de transferencia de etiquetas de [MAKA10] [GUILL09] [FU12], que son los mejores trabajos sobre la base de datos SAIAPR-TC12 (Tabla VI-3). Para ello, se ha solicitado el código a sus autores para evitar

malinterpretaciones en la implementación. Con esto, los resultados obtenidos de precision y recall por imagen son los siguientes:

Tabla VI-8: Resultados comparativos de la base de datos MIRFlickr

	ImageCLEF 2011 dataset		
	Precision (%)	***Recall (%)***	***F1-Measure***
Greedy Transfer [MAKA10]	63.4±0.24	29.5±0.13	0.403
TagProp σSD [GUILL09]	46.1±0.26	21.3±0.13	0.291
TagProp SD [GUILL09]	61.5±0.25	28.6±0.13	0.390
RF_count^2 [FU12]	26.1±0.25	10.9±0.10	0.154
Proposed algorithm	**69.9±0.22**	**32.8±0.13**	**0.446**

En este experimento se está utilizando la misma técnica de vecinos más cercanos, así que realmente es posible comparar la etapa de transferencia de etiquetas. Lo primero que se ve es que en términos de precisión, los algoritmos del estado del arte [FU12] [GUILL09] no son mejores que el baseline propuesto en [MAKA10]. Esto puede ser debido a que estos trabajos se centraron en mejorar la etapa de búsqueda de imágenes similares en vez de la etapa de transferencia de etiquetas. En este escenario, para un valor de imágenes similares de K=10, nuestra propuesta tiene mejor funcionamiento con una mejora del 6% de precisión sobre el segundo algoritmo en términos de precision, indicando que **la combinación de características semánticas y visuales en la etapa de transferencia de etiquetas tiene un impacto significativo en el resultado final**.

Para corroborar estos resultados, se ha seleccionado una segunda métrica para evaluar el algoritmo. Utilizando el script oficial provisto en ImageCLEF2011 se ha computado el MAP, EER y AUC. Es necesario mencionar que no se ha podido calcular estos valores para [MAKA10], ya que no devuelve estimaciones de probabilidad, tal y como requiere el script oficial.

Tabla VI-9: Resultados de los algoritmos del estado del arte computado con el script de evaluación de ImageCLEF2011

	F	MAP	EER	AUC
TagProp sigmaSD [GUILL09]	Text	0.177	0.423	0.433
TagProp SD [GUILL09]	Text	0.187	0.421	0.457
RF_count^2 [FU12]	Text + Visual	0.182	0.418	0.460
Our proposal	**Text + Visual**	**0.205**	**0.416**	**0.451**

Como ya se ha visto, lo más importante en esta fase es que la precisión sea la mayor posible, por lo tanto, el valor de referencia es el del Mean Average Precision (MAP). Se comprueba cómo el resultado de Image Networking supera a los otros algoritmos por más de un 2%.

Para finalizar con la validación de la mejora de este funcionamiento, y ya que las condiciones del número de vecinos similares de cada trabajo es diferente, se ha computado el resultado del MAP para diferentes tamaños de vecindad y se presentan en la Figura VI-8.

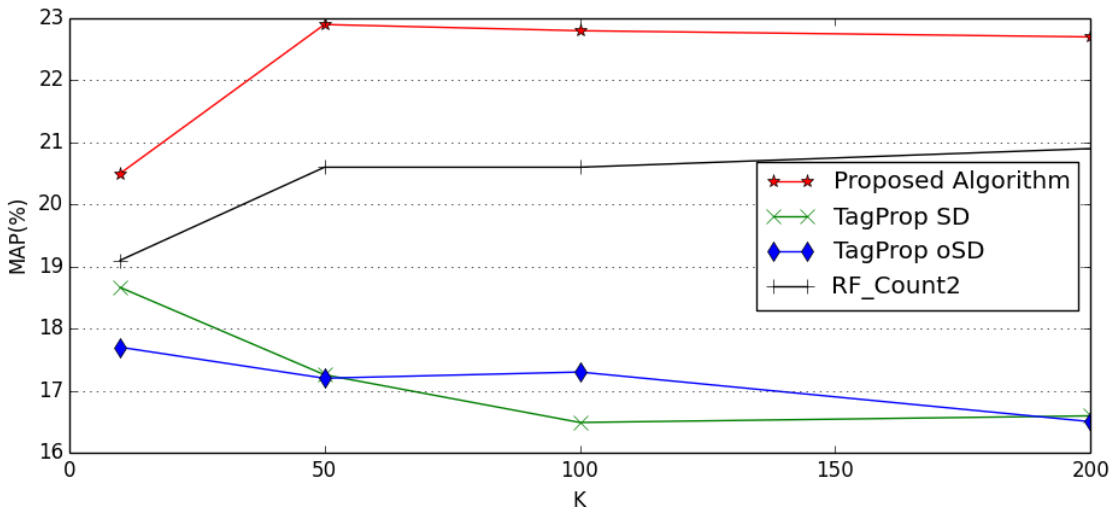


Figura VI-8: Resultado de los algoritmos comparados usando diferente tamaño de vecindad

Se comprueba cómo, independientemente del tamaño del vecindario, nuestra propuesta mejora el funcionamiento del resto de algoritmos, contando exactamente con un 2,3% mejor MAP que [RF_count] (22,9% contra 20,6%) en la mejor configuración (K=50).

Además, los resultados anteriores demuestran que la propuesta realizada predice correctamente etiquetas raras (mejor MAP), así como predice correctamente las etiquetas predichas en una imagen (mayor precisión por imagen).

En cuanto a una evaluación cualitativa de los resultados, en la Tabla VI-10 se muestran dos imágenes aleatorias de ejemplo. En ellas se muestran los resultados de anotación de los mejores algoritmos probados anteriormente, comprobando cómo nuestro algoritmo devuelve más etiquetas que están presentes en el *ground truth* que otros (estas etiquetas están señaladas en negrita y subrayadas).

Tabla VI-10: Etiquetas más relevantes para dos imágenes de la base de datos

	TagPropSD [GUILL09]	**RF_count^2 [FU12]**	**Our approach**
	Landscape_Nature Outdoor, Sky, Night, Neutral_Illumination, No_Blur, No_Persons, cute, calm, bodypart	Underexposed, Night, melancholic, Shadow, Single_Person, scary, Adult, male, bodypart, Indoor	No_Blur, Underexposed, No_Persons, Visual_Arts, cute, melancholic, Neutral_Illumination, Night, Outdoor, male
	Landscape_Nature Outdoor, Plants, Flowers, Trees, Sky, Water, Day, Sunset_Sunrise, Neutral_Illumination	Night, Underexposed, Indoor, Citylife, Single_Person, melancholic, Water, Visual_Arts	No_Blur, Neutral_Illumination, natural, No_Persons, Outdoor, Visual_Arts, Day, Sky, Indoor

VI.5 Conclusiones y trabajo futuro

En este capítulo se ha propuesto un nuevo algoritmo para la etapa de propagación de etiquetas en un modelo de anotación de imágenes basado en los vecinos más cercanos. La principal novedad de este algoritmo es que incorpora información semántica de las etiquetas candidatas e información visual de las imágenes de test y entrenamiento. El algoritmo se ha comparado contra tres algoritmos del estado del arte, utilizando dos bases de datos de tamaño medio ImageCLEF2011 y SAIAPR-TC12. Primero, se ha demostrado

cómo los algoritmos del estado del arte se centran en mejorar la etapa de vecinos más cercanos y no la etapa de transferencia de etiquetas, ya que se ha visto cómo, en diferentes condiciones de búsqueda por similitud en SAIAPR-TC12, el resultado es un poco peor, pero con las mismas condiciones sus resultados son inferiores.

Ante las mismas condiciones de entrada, el mejor algoritmo de transferencia de etiquetas es el propuesto en esta tesis y se ha corroborado con dos medidas diferentes.

De forma adicional, también se ha propuesto una técnica de aprendizaje local, *ImageNetworking*, que entrena el algoritmo en tiempo de consulta. Se ha demostrado en la base de datos SAIAPR-TC12 que esta técnica de aprendizaje estructurado aporta información local útil para ejecutar la transferencia de etiquetas, en vez de dar información genérica global como hacen otros sistemas.

Por último, hay que destacar que este método se puede usar con cualquier técnica de vecinos más cercanos, así que el resultado mostrado puede ser mejorado usando otra técnica de búsqueda de vecinos, por ejemplo, usando *metric learning*. Por ello, futuros pasos en esta línea pasarán por comprobar el funcionamiento de este algoritmo con este tipo de técnicas.

Además, también se ha visto cómo las etiquetas mejores son las de conceptos generales de la imagen, no objetos concretos de la misma. Una línea de trabajo futuro es mejorar este defecto, mejorando la representación espacial de la imagen, así como introduciendo modificadores para cada etiqueta, de forma que las etiquetas raras no se queden por debajo de las etiquetas más genéricas.

VII. Conclusiones, contribuciones y trabajo futuro

VII.1 Conclusions

This thesis fits in the automatic image annotation field and its main objective is to generate labels which describe accurately the content of digital images.

To this end, we have studied the vast state of the art devoted to this theme in the computer vision field. In addition, we have also analysed further scientific fields that are relevant to this task, i.e. multimedia analysis, artificial intelligence and information retrieval. This **detailed analysis of the most relevant works in the state of the art** has been carried out with the objective of defining a baseline algorithm which achieves the same performance as the state of the art in the nearest neighbours model based algorithms (Figure VII-1). Moreover, this algorithm has serve as starting point for the rest of the research done in this thesis.

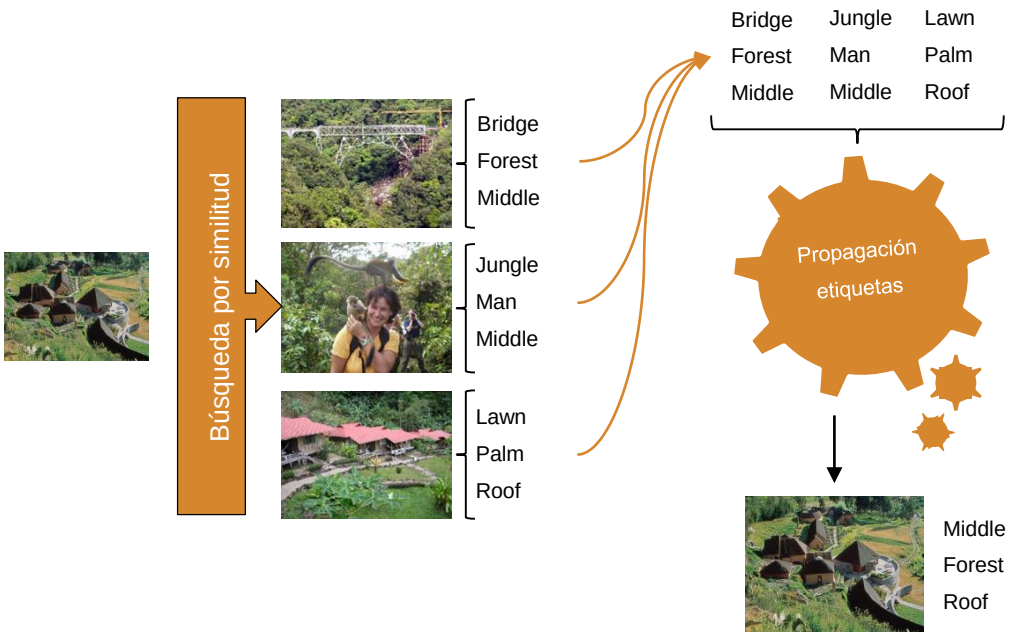


Figure VII-1: Phases of a generic nearest neigbor based annotation model

The tasks performed in this thesis have been focused on the two main stages of the nearest neighbours model. These stages are the *similarity based image search* and the *label propagation stage*. The outcome of the research done in these two areas is the following:

- The proposal of a **novel visual edge descriptor**, based on the 2/3 layer of the primate visual cortex and compatible with the MPEG7 standard.
- The proposal of a **new label propagation algorithm** and its **training algorithm** based on the explicit combination of textual and visual information.

The conclusions reached in these areas have been shown through the different chapters of this thesis but in the next paragraphs we summarize all of them.

The state of the art in this field is gigantic and the most part of computer vision research groups in the field have worked or are working in the area of image annotation or in similar themes, as object detection. The existing works cover practically all the possible research areas in the field. The main topic of these works is researching novel computer vision algorithms but also researching novel algorithms on large scale metric learning or analysis of the biological functions of the human brain. All these works are not compact so researchers can find wholes in between research and it is mandatory to fill these wholes to deeply understand the results generated by some of the state of the art works. Filling these wholes will allow to the research community to improve them and to move forward the complete area of automatic image annotation.

The first conclusion we obtain in this thesis was that, despite all the work that has been carried out, little attention has been paid to the MPEG7 standard in the computer vision field. This is not the case of the multimedia analysis area where it is one of the benchmarks. Even though, we have found little evidence, in terms of research publications, that researchers are trying to improve the reference implementation of the MPEG7 standard and the research community has opted for generating novel descriptors that presumably perform better than the standard but they are not compatible with them. The analysis carried out in this thesis has demonstrated that the MPEG7 standard obtains results similar to or better to the best algorithms in the state of the art, having the advantage of being more compact than them.

The second conclusion obtained in this thesis is related with the standard itself. MPEG7 defines a colour descriptor based on the human perception of the colours, but the defined edge descriptor is based on handcrafted filters. In this thesis we have demonstrated that improving these filters it is possible to improve the results obtained by the standard descriptor in the similarity based image search task. On top of that, in this thesis we

propose a new implementation of these filters that uses a neuron model, based on the functioning of the visual cortex during the edge recognition step in a primate brain, while being compatible with the standard

The third conclusion is related to the primate visual cortex. There are few references in the literature that have tried to apply correctly the primate visual cortex models in the computer vision field. We have seen many effects in the cortical neurons that have been studied and modelled by the neuroscientific literature, but mainly for single neurons and almost never extended to work with two dimensional images. On the other side, we have seen numerous computer vision publications that claim to use a biologically inspired models while they only use a biological concept and propose a novel computational model adapted to the problem they try to solve. After researching almost all the biological behaviours of the primate neurons the conclusion we have reached is that the existing behaviours are and will be very useful in the computer vision field.

The fourth and last conclusion is that, despite the large amount of works that exist in the image annotation field using the nearest neighbours model, very few of them propose novel algorithms for label propagation. Furthermore, among these few works that propose a novel algorithm the main contribution of them is the image similarity search algorithm, not the label propagation stage. In this thesis it has been demonstrated that improving the second stage of the algorithm can lead to much better results in terms of precision and recall of the final complete algorithm. In addition, we have also demonstrated that using not only visual information but textual information from the labels at the same time we can leverage a better result than using these features independently.

VII.2 Contributions and publications

This thesis has generated multiple contributions obtained from the conclusions of the completed work. These contributions are summarized below, divided by each chapter of this document.

The chapter II is focused on the study of the state of the art of different technologies related to the automatic image annotation field. It has produced the following outcomes:

1) Global study of the visual image descriptors, from the ones that generate simple information of the image given pixel statistics to the most complex descriptors that aggregate high level information.

2) Presentation of the different computation techniques of distances between vectors for similarity search tasks. In the presentation it can be found low level metrics as well as other metrics that use semantic information or use automatic learning to adapt the metrics to the problem.
3) Exhaustive study of the most relevant discriminative models for object detection in the last 15 years. The conceptual improvements of each algorithm has been shown as well as the result improvements.
4) Detailed analysis of the automatic annotation algorithms based on the nearest neighbours models. Its main stages has been studied, including their image search algorithms and label propagation stages.
5) Presentation of additional methodologies that allow automatic algorithms to annotate images based on the images context in a web environment.

The chapter III has targeted the state of the art algorithms and contains all the work done to generate a baseline for this thesis. The main contributions have been the following:

6) Study of the scientific image databases used in the computer vision and multimedia fields.
7) Presentation of the most relevant evaluation metrics used in the image annotation field.
8) Proposal of an improvement over the 2004 bag-of-words algorithms based on the SURF feature descriptor and the SVM classifier.
9) Detailed analysis of the discriminative models, demonstrating that the algorithm proposed in 2004 with the improvements included in 8) has still a state of the art performance.
10) Deep analysis of the different stages involved in the nearest neighbours based model and, more specifically, the computation of distances between images, the fast image similarity search, the description of the image content and the label propagation phase.
11) Proposal of a method for label propagation based on the most similar image in the database, demonstrating that it outperforms more complex state of the art algorithms.

12) Proposal of a new set of visual descriptors based on the MPEG7 standard which obtain similar performance as the state of the art while requiring a 98% less of space in disk to store them.
13) Proposal of a novel architecture for fast searching of images based on the use of the Local Sensitive Hashing based descriptors, binary tree search algorithms and searching distribution over different nodes using the Map-Reduce framework.

After a complete study of the literature, this thesis has focused on working on top of the nearest neighbours model. For that, the first stage of the model was studied in detail in the **chapter IV,** with a special emphasis in the edge detection mechanism in the primate visual cortex. In that chapter the following contributions have been proposed:

14) Confirmation that the MPEG7 standard has a colour descriptor based on the human perception while the edge descriptor was generated manually.
15) Global analysis of the workstreams of the macaque visual system, especially the static image path in the brain.
16) Presentation of a detailed study of the neurons present in the visual cortex, the signals that module them response and the signals that the neurons generate.
17) Review of the functional and computational models of the V1 layer of the cortex, with special attention to the effects that can be potentially used in the computer vision field.
18) Proposal of multiple neuronal models, based in the primary visual cortex, which can be used in a computer vision system.
19) Thorough study of the configuration parameters required by the proposed models based on the biological knowledge of real neurons.
20) Detailed study of the individual responses of the different neural models proposed in 18). Comparison with real responses.
21) Detailed study of the complete responses of the different neural models proposed in 18) with multiple neurons acting as a single cortex. Comparison of the real responses and selection of the neural model with the most faithful output.

After the definition of the neural model most suitable for the edge detection task, the **chapter V** has proposed the replacement of the standard implementation of the MPEG7-EHD descriptor by a novel implementation based in the proposed neural model. This proposal has generated the following contributions:

22) Integration of the cortex model proposed in 18), 20) and 21) in the MPEG7 standard with the proposal of a new implementation of the EHD descriptor.
23) Validation and tuning of the biological parameters of the cortex defined in 19).
24) Comparison of the models defined in 18) with respect to the state of the art using several image databases described in 6).

Once the new visual descriptor has been proposed, this thesis has centred its focus into the tag propagation stage of the nearest neighbours model. In this theme, the **chapter VI** has proposed the following contributions:

25) Proposal of a new method for textual and visual information integration during the generation of label rankings.
26) Proposal of a new machine learning algorithm trainer adapted to the learning task at a query time for the stage of label propagation.

All these individual contributions proposed in this thesis have generated several **research publications**. The contributions are combined in multiple publications that are presented in the next paragraphs divided by each chapter.

Specifically, the contributions proposed in the *chapter II* are present in the following journal and conference papers:

- *"Content-Based Image Annotation: Current Trends and Application" Sergio Rodriguez-Vaamonde. 5th International Conference on Image Processing and Machine Vision (HANDS-ON IMAGE PROCESSING 2011 – HOIP11). Zamudio. Spain. November 2011.*
- *"Joint Use of Semantics and Visual Analysis for Automatic Image Annotation and Retrieval" Sergio Rodriguez-Vaamonde, Pilar Ruiz-Ibañez, Marta Gonzalez Rodriguez. El Profesional de la Informacion. 2012, enero-febrero, v. 21, n. 1, pp. 27-33. (JCR)*
- *"Multimedia Information Interpretation. Is it still a challenge?" Round Table. Juan M. Cigarrán, Antonio Albacete, Sergio Rodríguez , Antonio Matarranz ,Frank Guijarro. VII MAVIR conference "Advances in Languages Technologies and Informations Access".Madrid November 26,27 (2012)*

The main contributions proposed in the *chapter III*, related to the state of the art analysis, are present in the next publications:

- *"Image analysis technologies on large data environments", A. Picon, A. Bereciartua, S. Rodriguez, A. Lopez,E. Muñoz, F. Gandon, F. Moscone, P. H.J. Riegman, S. García, R. Bilbao, European Data Forum (EDF), Copenhagen 2012.*
- *"Algoritmo de reconocimiento de expresiones faciales".Alberto Isasi-Andrieu, Sergio Rodriguez-Vaamonde, Jesus Herrero. tourGUNE Journal of Tourism and Human Mobility. June, year 2014 , Issue 2, pp. 35-40. ISSN 2340-6410*
- *"Plataforma de búsqueda de imágenes histológicas por similitud visual" C. L. Saratxaga, A.Picón, S. Rodriguez-Vaamonde, A. López-Carrera, J. Echazarra, A. Bereciartua, E. Garrote. Oral presentation at XVII Congreso Nacional de Informática de la Salud – 2014.*
- *"Tecnologías Big Data para el análisis y la recuperación rápida de imágenes en entornos web en base a similitud de imágenes" To be published in El Profesional de la Información "2015"*

Regarding the *chapter VI*, three are the main articles that show the developed technology and its application to the image annotation:

- *"What Can Pictures Tell Us About Web Pages? Improving Document Search using Images." Sergio Rodriguez-Vaamonde, Lorenzo Torresani and Andrew Fitzgibbon. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13). ACM, New York, NY, USA, 849-852. DOI=10.1145/2484028.2484144*
- *"Improving Tag Transfer for Image Annotation using Visual and Semantic Information", Sergio Rodriguez-Vaamonde, Lorenzo Torresani, Koldo Espinosa, Estibaliz Garrote , 12th International Workshop on Content Based Multimedia Indexing (CBMI), June 18-20 2014, Klagenfurt – Austria*
- *"What Can Pictures Tell Us About Web Pages? Improving Document Search using Images." Sergio Rodriguez-Vaamonde, Lorenzo Torresani and Andrew Fitzgibbon. To be published in Transactions on Pattern Analysis and Machine Intelligence 2015.*

Finally, only one publication has been accepted so far in relation to the *chapters IV and V*:

- *"Modelización bayesiana de preferencias visuales en el ámbito turístico".Sergio Rodriguez-Vaamonde , Jesus Herrero, Estibaliz Garrote. tourGUNE Journal of*

Tourism and Human Mobility. December, year 2013 , Issue 1, pp. 27-33. ISSN 2340-6410

In addition to this last article, we are in process of writing and sending a detailed article related to the model proposed in these chapters.

Moreover, the knowledge obtained in this thesis, as well as some of their contributions, has helped Tecnalia to achieve the EARTO INNOVATION AWARD 2014 by the project BIOSIMIL.

VII.3 Future work

This thesis is not an oasis in the middle of the dessert. As it has seen in the chapter II of this thesis, there is a huge amount of research in the image annotation field or related field. The work done in this thesis is only a grain of sand laid in the research pathway that will bring novel and better scene understanding systems. Inside this pathway there are multiple possibilities that have not been analysed in detail by this thesis but they have been spotted as useful for the future of the field.

One of these possibilities is centred on improving the computation of the distances between images in order to find a “better” similar image. In the chapter III.4.2.4 it has been proposed a novel algorithm to compute these distances faster based on the visual descriptors. As it has been demonstrated by this thesis, the combination of textual and visual information is a benefit for other tasks, so one of the future work lines will be to combine this information in the similarity search stage instead of using it in the label propagation stage. A starting point can be the ImageNet distance (II.3.1.10) combined with the architecture proposed in the chapter III.4.2.4.

In relation to the cortex model, it has been shown in the chapter IV.2.2 that there exist multiple effects that modulate the cortex neuron response. Given all the studied effects, this thesis only has been focused on the suppression and facilitation effects generated by the horizontal connexions of the neurons (IV.4). It is clear that one of the future working areas will be adding more *feedback* information from neurons that reside in cortical layers different from V1. Another fascinating work can be modelling the variation of the neural configuration parameters (for example, depending on the contrast of the image one neuron can behave differently, as the real neurons do) or even connect the proposed cortex model with a retinal computational model, as proposed in [GARR11].

In relation to the cortex modelling, nowadays the research community is living in the golden age of the neural networks (II.4.2.6). In fact, some of the latest models of artificial neural networks for object detection present a convolution layer followed by a max-pooling layer. This particular structure is not so far from the model proposed in the chapter IV.4, so one work area will be to introduce the models proposed by this thesis as layers in a general purpose artificial neural network. This will allow to better train our model, and to improve the results obtained by this thesis.

The proposed cortex model is the core of the new MPEG7-EHD implementation described in the chapter V. As it is at the core of the implementation, the model is also the bottleneck of the implementation. The use of multiple neurons to cover 2D images requires the computation of multiple convolutions on top of the convolutions required by the EHD descriptor itself. So even though this model obtains better results than the standard in terms of recall, the proposed method is slightly slower than the standard. This can be easily solved using processing architectures adapted to computation of convolutions, and one of the major exponents of this technology are the Graphical Processing Units (GPU).

Finally, another research area which can continue the work started by this thesis is the improvement of the label propagation algorithm proposed in the chapter VI. Despite we have demonstrated that the proposed algorithm outperforms the state of the art, it has been noticed that this algorithm and others from the state of the art tend to recognize easily the most common labels while failing detecting the uncommon concepts. This is because the common labels have more statistical influence in the final evaluation metrics, so the automatic algorithms tend to improve the results of these ones. Further work must be done to modify this tendency in the proposed algorithm and train the system more homogenously over all the labels.

As the reader can realize, in all chapters of this thesis there is pending work that can improve even more the work done on them. This fact presents a future with multiple open research lines that start in this thesis and they will be explored by different research groups, including those involved in the development of this thesis, which are the *Multimedia Group at the University of the Basque Country*, the *Visual Learning Group* at *Dartmouth College* and the *Computer Vision Group* at *Tecnalia*.

Referencias

A

[ADES10] Adesnik, H., & Scanziani, M. (2010). Lateral competition for cortical space by layer-specific horizontal circuits. Nature, 464(7292), 1155-1160.

[AHN04] von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: ACM CHI. (2004)

[ANGE03] Angelucci, A., & Bullier, J. (2003). Reaching beyond the classical receptive field of V1 neurons: horizontal or feedback axons?. Journal of Physiology-Paris, 97(2), 141-154.

B

[BARN03] K. Barnard, P. Duygulu, D. Forsyth, N. D. Freitas, D. M. Blei, J. K, T. Hofmann, T. Poggio, and J. Shawe-taylor. Matching words and pictures. Journal of Machine Learning Research , 3:1107–1135, 2003

[BAJC13] Bajcsy, P., Vandecreme, A., Amelot, J., Nguyen, P., Chalfoun, J., & Brady, M. (2013). “Terabyte-sized image computations on Hadoop cluster platforms.” In Big Data, 2013 IEEE International Conference on, pp. 729-737.

[BAY08] Bay, Herbert; Ess, Andreas; Tuytelaars, Tinne and van Gool, Luc (2008). SURF: Speeded up robust features. Computer Vision and Image Understanding 110(3): 346-359.

[BEHJ10] Behjat Siddiquie and Abhinav Gupta. Beyond Active Noun Tagging: Modeling Contextual Interactions for Multi-Class Active Learning IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2010.

[BELO02] Belongie, S., J. Malik, and J. Puzicha (2002, April). Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (4), 509-522.

[BENG06] Bengio, P. Lamblin, D. Popovici and H. Larochelle, Greedy Layer-Wise Training of Deep Networks, in Advances in Neural Information Processing Systems 19 (NIPS'06), pages 153-160, MIT Press 2007.

[BENG09] Bengio, Y. (2009). Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), 1-127.

[BERG08] Karl Berggren & Pär Gregersson. "Camera focus controlled by face detection on GPU". Master Thesis 2008. Lund University

[BERG11] Bergamo, A., Torresani, L., & Fitzgibbon, A. W. (2011). Picodes: Learning a compact code for novel-category recognition. In Advances in Neural Information Processing Systems (pp. 2088-2096).

[BERG14] BERGAMO, Alessandro; TORRESANI, Lorenzo. Classemes and Other Classifier-based Features for Efficient Object Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, p. 1.

[BILV14] BilVideo. http://www.cs.bilkent.edu.tr/~bilmdg/bilvideo-7/MPEG7Profile.html. Accessed July 2014.

[BOSC06] Bosch, Anna; Zisserman, Andrew and Munoz, Xavier (2006). Scene classification via pLSA. Proc. 9th European Conference on Computer Vision (ECCV'06) Springer Lecture Notes in Computer Science 3954: 517~530.

[BOSC07] A. Bosch, A. Zisserman & X. Munoz. 2007. Representing shape with a spatial pyramid kernel. In Proceedings of CIVR '07

[BOSK97] Bosking, W. H., Zhang, Y., Schofield, B., & Fitzpatrick, D. (1997). Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. The Journal of Neuroscience, 17(6), 2112-2127.

C

[CARA05] Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., ... & Rust, N. C. (2005). Do we know what the early visual system does?. The Journal of neuroscience, 25(46), 10577-10597.

[CARA12] Matteo Carandini (2012) Area V1. Scholarpedia, 7(7):12105., revision #126411

[CARA12b] Carandini, M., & Heeger, D. J. (2011). Normalization as a canonical neural computation. Nature Reviews Neuroscience, 13(1), 51-62.

[CARA97] M. Carandini, D.J. Heeger and J.A. Movshon, Linearity and normalization in simple cells of the macaque primary visual cortex. J Neurosci, 1997. 17: 8621-44.

[CARA99] Carandini M, Heeger DJ, Movshon JA (1999) Linearity and gain control in V1 simple cells. In: Cerebral Córtex, Vol 13: Models of cortical circuits (Ulinski PS, Jones EG, Peters A, eds), pp 401-443. New York: Kluwer Academic/ Plenum.

[CARB04] P. Carbonetto, N.D. Freitas, K. Barnard, A statistical models for general contextual object recognition, in: European Conference on Computer Vision, 2004.

[CAVA02] Cavanaugh, J. R., Bair, W., & Movshon, J. A. (2002). Nature and interaction of signals from the receptive field center and surround in macaque V1 neurons. Journal of neurophysiology, 88(5), 2530-2546.

[CHAT08] S. A. Chatzichristofis and Y. S. Boutalis, “CEDD: COLOR AND EDGE DIRECTIVITY DESCRIPTOR – A COMPACT DESCRIPTOR FOR IMAGE INDEXING AND RETRIEVAL.”, «6th International Conference in advanced research on Computer Vision Systems (ICVS)», Lecture Notes in Computer Science (LNCS), pp.312-322, May 12 to 15, 2008, Santorini, Greece.

[CHANG01] Chang, S. F., Sikora, T., & Purl, A. (2001). Overview of the MPEG-7 standard. Circuits and Systems for Video Technology, IEEE Transactions on, 11(6), 688-695.

[CHEC10] Chechik, G., Sharma, V., Shalit, U., & Bengio, S. (2010). Large scale online learning of image similarity through ranking. The Journal of Machine Learning Research, 11, 1109-1135.

[CHEE02] Chee Sun Won , Dong Kwon Park , and Soo-Jun Park , "Efficient Use of MPEG-7 Edge Histogram Descriptor," ETRI Journal, vol. 24, no. 1, Feb. 2002, pp. 23-30. http://dx.doi.org/10.4218/etrij.02.0102.0103

[CHIS03] Chisum, H. J., Mooser, F., & Fitzpatrick, D. (2003). Emergent properties of layer 2/3 neurons reflect the collinear arrangement of horizontal connections in tree shrew visual cortex. The Journal of neuroscience, 23(7), 2947-2960.

[CHOI10] Myung Jin Choi, Joseph Lim, Antonio Torralba, and Alan S. Willsky. Exploiting Hierarchical Context on a Large Database of Object Categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, June 2010.

[CLAU00] Clausi, D. A., & Jernigan, M. E. M. E. (2000). Designing Gabor filters for optimal texture separability. Pattern Recognition, 33(11), 1835-1849.

[CSUR04] Csurka, Gabriella; Dance, Christopher; Fan, Lixin; Willamowski, Jutta; Bray, Cédric. "Visual categorization with bags of keypoints". Workshop on Statistical Learning in Computer Vision, ECCV. 2004

[CSUR07] Csurka, G., Dance, C. R., Perronnin, F., and Willamowski, J. (2007). Generic visual categorization using weak geometry. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A., editors, Toward Category-Level Object Recognition, pages 207–224, Springer, New York.

D

[DAHL13] Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013, May). Improving deep neural networks for LVCSR using rectified linear units and dropout. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 8609-8613). IEEE.

[DALA05] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[DALA12] Dalal, N. "Histograms of oriented gradients for object detection". Lecture in CS8690, Computer Vision, Spring 2012. University of Missouri.

[DAVI07] Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007, June). Information-theoretic metric learning. In Proceedings of the 24th international conference on Machine learning (pp. 209-216). ACM.

[DEAN08] Dean, J., & Ghemawat, S. (2008). "MapReduce: simplified data processing on large clusters." Communications of the ACM, 51(1), 107-113.

[DEAN12] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Ng, A. Y. (2012). Large scale distributed deep networks. In Advances in Neural Information Processing Systems (pp. 1223-1231).].

[DEAN95] DeAngelis GC, Ohzawa I, Freeman RD (1995) Receptive-field dynamics in the central visual pathways. Trends Neurosci 18:451-458.

[DESE08] Deselaers, T., Keysers, D., & Ney, H. (2008). Features for image retrieval: an experimental comparison. Information Retrieval, 11(2), 77-107.

[DESE10] Thomas Deselaers and Vittorio Ferrari. Global and Efficient Self-Similarity for Object Classification and Detection. CVPR 2010. IEEE Computer Vision and Pattern Recognition, San Francisco, June 2010.

[DESE11] Deselaers, Thomas; Ferrari, Vittorio. “Visual and Semantic Similarity in ImageNet”, IEEE Computer Vision and Pattern Recognition (CVPR) Conference. 2011.

[DENG09] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[DENG10] Deng, J., Berg, A. C., Li, K., & Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us?. In *Computer Vision–ECCV 2010* (pp. 71-84). Springer Berlin Heidelberg.

[DICK09] Dickinson, Sven J.; Leonardis, Ale; Schiele, Bernt; Tarr, Michael J.. “A Strategy for Understanding How the Brain Accomplishes Object Recognition”. Object Categorization. Cambridge University Press, 2009. http://dx.doi.org/10.1017/CBO9780511635465

[DORK05] Dorkó, G. and Schmid, C. 2005. Object class recognition using discriminative local features. Technical Report RR-5497, INRIA - Rhône-Alpes.

[DOUZ09] Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., & Schmid, C. (2009, July). Evaluation of gist descriptors for web-scale image search. In Proceedings of the ACM International Conference on Image and Video Retrieval (p. 19). ACM.

[DUYG02] Duygulu, P., Barnard, K., de Freitas, J. F., & Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Computer Vision—ECCV 2002 (pp. 97-112). Springer Berlin Heidelberg.

E

[ELHA09] Elahi, R. Karlsen; Akselsen , S. “A Context Centric Approach for Semantic Image Annotation and Retrieval”. Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD ’09. Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD ’09. Computation World: IEEE, págs. 665-668.

[ELHA11] Elahi, R.Karlsen, W. Younas. “Image Annotation by Leveraging the Social Context", ACM ICUIMC 2011, The 5th International Conference on Ubiquitous Information Management and Communication. 2011.Seoul, South Korea, 21-23 February.

[ENGE10] Karin Engel, Klaus D. Toennies, Hierarchical vibrations for part-based recognition of complex objects, Pattern Recognition, Volume 43, Issue 8, August 2010, Pages 2681-2691, ISSN 0031-3203

[EVER10] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

F

[FEIF04] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In IEEE CVPR Workshop on Generative-Model Based Vision, 2004.

[FEIF07] Fei-Fei, L., Fergus, R., and Perona, P. 2007. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. 106, 1 (Apr. 2007), 59-70.

[FELL91] Felleman, D.J. and Van Essen, D.C. (1991). Distributed hierarchical processing in the primate cerebral córtex. Cereb. Córtex 1:1-4.

[FELZ05] Felzenszwalb, P. F. and D. P. Huttenlocher (2005, January). Pictorial structures for object recognition. Int. J. Comput. Vision 61 (1), 55-79.

[FELZ08] P. Felzenszwalb, D. McAllester, D. Ramaman. A Discriminatively Trained, Multiscale, Deformable Part Model. Proceedings of the IEEE CVPR 2008.

[FELZ10] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9), 1627-1645.

[FENG04] Feng, S. L., Manmatha, R., & Lavrenko, V. (2004, June). Multiple bernoulli relevance models for image and video annotation. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Vol. 2, pp. II-1002). IEEE.

[FERG03] Fergus, R., Perona, P., and Zisserman, P. 2003. Object class recognition by unsupervised scale-invariant learning. In Proc. CVPR.

[FERR04] Vittorio Ferrari, Tinne Tuytelaars, Luc Van Gool. "Integrating Multiple Model Views for Object Recognition". IEEE Computer Vision and Pattern Recognition (CVPR), Washington, USA, June 2004

[FERR10] V. Ferrari, F. Jurie, and C. Schmid. "From Images to Shape Models for Object Detection". International Journal of Computer Vision (IJCV), March 2010.

[FU12] Fu, H., Zhang, Q., & Qiu, G. (2012). Random forest for image annotation. In Computer Vision–ECCV 2012 (pp. 86-99). Springer Berlin Heidelberg.

[FUKU07] Kunihiko Fukushima (2007) Neocognitron. Scholarpedia, 2(1):1717., revision #91558

[FUKU80] K. Fukushima: "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", Biological Cybernetics, 36[4], pp. 193-202 (April 1980).

G

[GALL10] Galleguillos, C., McFee, B., Belongie, S., Lanckriet, G.R.G. (2010). Multi-Class Object Localization by Combining Local Contextual Interactions. Proceedings of the 23rd IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA.

[GARC11] Garcia-Serrano, Ana. “UNED-UV experiments using Multimodal information approaches”. ImageCLEF 2011. Amsterdam.

[GARR11] Garrote E. (2011). Algorithms for colour image processing based on Neurological Models, PhD Thesis.

[GAO10] Gao, Shenghua; Chia, Liang-Tien; Cheng, Xiangang. “Web image concept annotation with better understanding of tags and visual features”. Journal of Visual Communication and Image Representation 2010, v21, no. 8, p806-814.

[GEHE09] Gehler, P. V. and S. Nowozin: On Feature Combination for Multiclass Object Classification. Proceedings of the Twelfth IEEE International Conference on Computer Vision (ICCV 2009), 1-8, IEEE Computer Society, Los Alamitos, CA, USA (10 2009)

[GEHL09] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in ICCV , 2009.

[GEME10] J. C. van Gemert,C. G. M. Snoek, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Comparing Compact Codebooks for Visual Categorization. Computer Vision and Image Understanding, in press, 2010.

[GRIFF06] Griffin, G. Holub, AD. Perona, P. The Caltech 256. Caltech Technical Report. (2006)

[GRUB06] Grubinger M., Clough P., Müller H., and Deselaers T.. The IAPR Benchmark: A New Evaluation Resource for Visual Information Systems. International Conference on Language Resources and Evaluation, 24/05/2006, Genoa, Italy, (2006).

[GU09] Recognition using Regions Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik. CVPR 2009, Miami, Florida.

[GUILL09] Guillaumin, M., Mensink, T., Verbeek, J., & Schmid, C. (2009, September). Tagprop: Discriminative metric learning in nearest neighbor models for image auto-

annotation. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 309-316). IEEE.

[GUPT08] Gupta, A. and Davis, L. S. 2008. Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. In Proceedings of the 10th European Conference on Computer Vision: Part I (Marseille, France, October 12 - 18, 2008). D. Forsyth, P. Torr, and A. Zisserman, Eds. Lecture Notes In Computer Science, vol. 5302. Springer-Verlag, Berlin, Heidelberg, 16-29.

H

[HAAR14] Haar wavelet. (2014, May 26). In Wikipedia, The Free Encyclopedia. Retrieved 08:29, July 10, 2014, from http://en.wikipedia.org/w/index.php?title=Haar_wavelet&oldid=610192192

[HARR54] Harris, Z. S. (1954). Distributional structure. Word.

[HEIT08] Heitz, G. and Koller, D. 2008. Learning Spatial Context: Using Stuff to Find Things. In Proceedings of the 10th European Conference on Computer Vision: Part I (Marseille, France, October 12 - 18, 2008). D. Forsyth, P. Torr, and A. Zisserman, Eds. Lecture Notes In Computer Science, vol. 5302. Springer-Verlag, Berlin, Heidelberg, 30-43.

[HEYD84] R. von der Heydt, E. Peterhans and G. Baumgartner, Illusory contours and cortical neuron responses. Science, 1984. 224: 1260-1262.

[HINT06] G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

[HITC41] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. J. Math. Phys., 20:224-230, 1941

[HONG10] Honglak Lee. NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning. Co-organizers: Yoshua Bengio, Geoff Hinton, Yann LeCun , Andrew Ng, and Marc'Aurelio Ranzato

[HORW12] Horwitz GD, Hass CA (2012) Nonlinear analysis of macaque V1 color tuning reveals cardinal directions for cortical color processing. Nat Neurosci. doi: 10.1038/nn.3105.

[HSL14] HSL and HSV. (2014, June 1). In Wikipedia, The Free Encyclopedia. Retrieved 08:49, July 10, 2014, from http://en.wikipedia.org/w/index.php?title=HSL_and_HSV&oldid=611123087

[HUAN10] Huang, Z., & Leng, J. (2010, April). Analysis of Hu's moment invariants on image scaling and rotation. In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Vol. 7, pp. V7-476). IEEE.

[HUBE59] Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate córtex. J Physiol 148:574-591.

[HUBE62] Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual córtex. J Physiol 160:106-154.

[HUBE98] D.H. Hubel and T.N. Wiesel, Early exploration of the visual córtex. Neuron, 1998. 20: 401-12.

[HUISK08] M. J. Huiskes, M. S. Lew (2008). The MIR Flickr Retrieval Evaluation. ACM International Conference on Multimedia Information Retrieval (MIR'08), Vancouver, Canada

[HUPE01] Hupe, J. M., James, A. C., Girard, P., Lomber, S. G., Payne, B. R., & Bullier, J. (2001). Feedback connections act on the early part of the responses in monkey visual cortex. Journal of Neurophysiology, 85(1), 134-145.

[HWAN10] S. J. Hwang and K. Grauman. Reading Between The Lines: Object Localization Using Implicit Cues from Image Tags. To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, June 2010. (Oral)

I

[IAKO14] Iakovidou, Chryssanthi; Anagnostopoulos, Nektarios; Kapoutsis, Athanasios Ch.; Boutalis, Yiannis; Chatzichristofis, Savvas A, "Searching images with MPEG-7 (&

MPEG-7-like) Powered Localized dEscriptors: The SIMPLE answer to effective Content Based Image Retrieval," Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on , vol., no., pp.1,6, 18-20 June 2014. doi: 10.1109/CBMI.2014.6849821

[ICHI07] Ichida, J. M., Schwabe, L., Bressloff, P. C., & Angelucci, A. (2007). Response facilitation from the “suppressive” receptive field surround of macaque V1 neurons. Journal of Neurophysiology, 98(4), 2168-2181.

[INST14] Instagram (2014). “Instagram Press Stats”, http://instagram.com/press/

[ISO02] ISO. "ISO/IEC 15938-1:2002 - Information technology -- Multimedia content description interface -- Part 1: Systems".

[ITO04] M. Ito and H. Komatsu, Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. J Neurosci, 2004. 24: 3313-24.

J

[JAMI10] M. Jamieson, A. Fazly, S. Stevenson, S. Dickinson, and S. Wachsmuth, Using Language to Learn Structured Appearance Models for Image Annotation,

IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32., No. 1, January 2010, pp 148--164.

[JEGO10] Jégou, H., Douze, M., & Schmid, C. (2010). Improving bag-of-features for large scale image search. International Journal of Computer Vision, 87(3), 316-336.

[JEON03] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. SIGIR, 2003

[JIA13] Yangqing Jia. {Caffe}: An Open Source Convolutional Architecture for Fast Feature Embedding (2013). http://caffe.berkeleyvision.org/

[JOAC02] Joachims, T. (2002, July). Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 133-142). ACM.

[JOAC07] Joachims, T. (2006, August). Training linear SVMs in linear time. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 217-226). ACM.

[JOAC09] Joachims, T., Hofmann, T., Yue, Y., & Yu, C. N. (2009). Predicting structured objects with support vector machines. Communications of the ACM, 52(11), 97-104.

[JONE87] Jones, J. P. and Palmer, L. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology ,58:1233–1258.

[JUNQ05] Junqueira, L.C., Carneiro, J. (2005). Basic Histology (11th ed.). New York: McGraw-Hill

K

[KASY13] Reza Kasyauqi Sabhara, Chin-Poo Lee, and Kian-Ming Lim. “Comparative Study of Hu Moments and Zernike Moments in Object Recognition”. Smart Computing Review, vol. 3, no. 3, June 2013.

[KEND84] Kendall, David G. "A Survey of the Statistical Theory of Shape." Statistical Science. Vol. 4, No. 2, 1989, pp. 87–99.

[KHOT90] A. Khotanzad, A. Y. Hong, —Invariant image recognition by Zernike moments,‖ IEEE Transaction on Pattern Analaysis and Machine Intelligence. vol. 12, no. 5, May 1990.

[KIPE02] Kiper, D. C., & Carandini, M. (2002). Neural basis of pattern vision.

[KOST12] Kostinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2012, June). Large scale metric learning from equivalence constraints. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2288-2295). IEEE.

[KREI08] Gabriel Kreiman (2008) Biological object recognition. Scholarpedia, 3(6):2667., revision #91063

[KRIZ12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[KULI12] Kulis, B., & Grauman, K. (2012). Kernelized locality-sensitive hashing. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(6), 1092-1104.

L

[LAMM00] Lamme, V. A., & Roelfsema, P. R. (2000). The distinct modes of vision offered by feedforward and recurrent processing. Trends Neurosci, 23(11), 571-579. doi:10.1016/S0166-2236(00)01657-X.

[LAZEB06] Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Vol. 2, pp. 2169-2178). IEEE.

[LEE10] Lee, Y. J; Grauman, K. “Object-graphs for context-aware category discovery”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, p1-8.

[LEIB08] Leibe, B., Leonardis, A., and Schiele, B. 2008. Robust Object Detection with Interleaved Categorization and Segmentation. Int. J. Comput. Vision 77, 1-3 (May. 2008), 259-289.

[LEVI10] Levi, D. and Ullman, S. 2010. Learning to classify by ongoing feature selection. Image Vision Comput. 28, 4 (Apr. 2010), 715-723.

[LI09] Li-Jia Li and Li Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. International Journal of Computer Vision (IJCV), 2009.

[LIEN02] An extended set of Haar-like features for rapid object detection. Proceedings. 2002 International Conference on In Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 1 (2002), pp. I-900-I-903 vol.1.

[LIU02] Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. Image processing, IEEE Transactions on, 11(4), 467-476.

[LIU12] Liu, W., Wang, J., Ji, R., Jiang, Y. G., & Chang, S. F. (2012, June). Supervised hashing with kernels. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2074-2081). IEEE.

[LOWE04] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

M

[MAKA10] Makadia, A., Pavlovic, V., & Kumar, S. (2010). Baselines for image annotation. International Journal of Computer Vision, 90(1), 88-105.

[MART13] Martín, D., López-de-Ipiña, D., Alzua-Sorzabal, A., Lamsfus, C., & Torres-Manzanera, E. (2013). “A methodology and a web platform for the collaborative development of context-aware systems”. Sensors, 13 (5), 6032-6053

[MCFE10] McFee, B., & Lanckriet, G. R. (2010). Metric learning to rank. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 775-782).

[MIKO06] Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1 (2006), pp. 26-36.

[MOHA01] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 349-361, 2001.

[MONA04] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: Constraining the latent space. In ACM Multimedia, pages 348–351, 2004

[MOVS78] Movshon JA, Thompson ID, Tolhurst DJ (1978c) Spatial summation in the receptive fields of simple cells in the cat's striate córtex. J Physiol 283:53-77

[MURP06] K. Murphy, A. Torralba, D. Eaton, W. T. Freeman. Object detection and localization using local and global features. In Toward Category-Level Object Recognition. Springer-Verlag Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman (eds.), 2006.

N

[NASS09] Nassi, J. J., & Callaway, E. M. (2009). Parallel processing strategies of the primate visual system. Nature Reviews Neuroscience, 10(5), 360-372.

[NEWM14] New Mexico Tech University. “Color Theory”. http://infohost.nmt.edu/tcc/help/pubs/colortheory/web/hsv.html . Last accesses July 2014.

[NUCL14] Núcleo geniculado lateral. (2014, 23 de abril). Wikipedia, La enciclopedia libre. Fecha de consulta: 08:15, julio 11, 2014 desde http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_geniculado_lateral&oldid=73978349.

O

[OHM01] Ohm, J. R., Cieplinski, L., Kim, H. J., Krishnamachari, S., Manjunath, B. S., Messing, D. S., & Yamada, A. (2001). The MPEG-7 Color Descriptors. Introduction to MPEG-7: Multimedia Content Description Interface, Wiley.

[OJAL96] T. Ojala, M. Pietikäinen, and D. Harwood (1996), "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59.

[OLIV01] Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision, 42(3), 145-175.

[OMME09] Björn Ommer and Jitendra Malik, Multi-Scale Object Detection by Clustering Lines, in: ICCV'09, IEEE, 2009.

[OMME10] Ommer, B. and Buhmann, J.M. “Learning the Compositional Nature of Visual Object Categories for Recognition”. Trans o Pattern Analysis and Machine Learning, vol 32, pp. 501-516, March. 2010.

[OPEL06] Opelt, A., Pinz, A., Fussenegger, M., Auer, P., 2006. Generic object recognition with boosting. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (3), 416-431.

[OLSH96] B.A. Olshausen and D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 1996. 381: 607-9.

[OZEK04] Ozeki, H., Sadakane, O., Akasaki, T., Naito, T., Shimegi, S., & Sato, H. (2004). Relationship between excitation and inhibition underlying size tuning and contextual response modulation in the cat primary visual cortex. The Journal of neuroscience, 24(6), 1428-1438.

P

[PAPA00] Papageorgiou, C., Poggio, T., 2000. A trainable system for object detection. Int. J. Comput. Vision 38 (1), 15-33.

[PASU99] A. Pasupathy and C. Connor, Responses to contour features in macaque area V4. Journal of Neurophysiology, 1999. 82: 2490-2502

[POGG13] Tomaso Poggio and Thomas Serre (2013) Models of visual cortex. Scholarpedia, 8(4):3516., revision #131433

[POLL81] Pollen, D. A. and Ronner, S. F. (1981). Phase relationships between adjacent simple cells in the visual cortex. Science, 212:1409–1411

R

[RAND99] Randen, T.; Husoy, J.H., "Filtering for texture classification: a comparative study," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.21, no.4, pp.291,310, Apr 1999. doi: 10.1109/34.761261

[RED14] Red neuronal artificial. (2014, 22 de junio). Wikipedia, La enciclopedia libre. Fecha de consulta: 14:20, julio 10, 2014 desde http://es.wikipedia.org/w/index.php?title=Red_neuronal_artificial&oldid=75170825

[RIED93] Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Neural Networks, 1993., IEEE International Conference on (pp. 586-591). IEEE

[RODR14] Rodriguez-Vaamonde, S., Torresani, L., Espinosa, K., & Garrote, E. (2014, June). Improving tag transfer for image annotation using visual and semantic information. In Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on (pp. 1-4). IEEE.

[RUSS14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge 2014”, arXiv:1409.0575

[RUST05] Rust, N. C., Schwartz, O., Movshon, J. A., & Simoncelli, E. P. (2005). Spatiotemporal elements of macaque v1 receptive fields. Neuron, 46(6), 945-956.

S

[SARA14] Saratxaga, C. L.; Picón, A.; Rodriguez-Vaamonde, S.; López-Carrera, A.; Echazarra, J.; Bereciartua, A.; Garrote, E. (2014). “Plataforma de búsqueda de imágenes histológicas por similitud visual” En: XVII Congreso Nacional de Informática de la Salud.

[SAVA07] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. IEEE Intern. Conf. in Computer Vision (ICCV). 2007.

[SCEN99] M.P. Sceniak, D.L. Ringach, M.J. Hawken, R. Shapley, Contrast’s effect on spatial summation by macaque V1 neurons, Nat.Neurosci. 2 (1999) 733–739

[SCEN01] Sceniak, M. P., Hawken, M. J., & Shapley, R. (2001). Visual spatial characterization of macaque V1 neurons. Journal of Neurophysiology, 85(5), 1873-1887.

[SCHW06] Schwabe, L., Obermayer, K., Angelucci, A., & Bressloff, P. C. (2006). The role of feedback in shaping the extra-classical receptive field of cortical neurons: a recurrent network model. The Journal of Neuroscience, 26(36), 9117-9129.

[SERI03] Series, P., Lorenceau, J., & Frégnac, Y. (2003). The “silent” surround of V1 receptive fields: theory and experiments. Journal of physiology-Paris, 97(4), 453-474.

[SERR05] Serre, T., L. Wolf and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In: Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Press, San Diego, June 2005.

[SERR07] Serre, Thomas; Wolf, Lior; Bileschi, Stanley; Riesenhuber, Maximilian; Poggio, Tomaso. “Robust object recognition with cortex-like mechanisms”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, v29, p411-426.

[SHEC07] E. Shechtman and M. Irani, “Matching local self-similarities across images and videos,” in CVPR, 2007.

[SHOT06] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), LNCS: Vol. 3951. Proceedings of European conference on computer vision (pp. 1-15). May 2006. New York: Springer.

[SHOT08] Jamie Shotton, Andrew Blake, and Roberto Cipolla, Multi-Scale Categorical Object Recognition Using Contour Fragments, in Trans. on PAMI, July 2008

[SHOT09] J. Shotton, J.Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV, 81(1):2–23, 2009.

[SILL06] Sillito, A. M., Cudeiro, J., & Jones, H. E. (2006). Always returning: feedback and sensory processing in visual cortex and thalamus. Trends in neurosciences, 29(6), 307-316.

[SLAN08] Slaney, Malcolm; Casey, Michael (2008). "Locality-sensitive hashing for finding nearest neighbors". Signal Processing Magazine, IEEE v.25, n.2, pp. 128-131.

[SLAN12] Slaney, M., Lifshits, Y., & He, J. (2012). Optimal parameters for locality-sensitive hashing. Proceedings of the IEEE, 100(9), 2604-2623.

[SMOL86] Smolensky, Paul (1986). "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory". In Rumelhart, David E.; McLelland, James L. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press. pp. 194–281. ISBN 0-262-68053-X

[SRIV13] Srivastava, N. (2013). Improving neural networks with dropout (Doctoral dissertation, University of Toronto).

[STON08] Stone, Z.; Zickler, T.; Darrel , T. “Autotagging Facebook: Social Network Context Improves Photo Annotation”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. 2008

[SZEG14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions” ArXiv: 1409.4842

T

[TANG11] Tang ; Yang, S.; Chua, T.; Jaim, R. “Label-specific training set construction from web resource for image annotation”. CoRR. 2011.

[TECH10] TechCrunch (2010). “Eric Schmidt: Every 2 Days We Create As Much Information As We Did Up To 2003”. http://techcrunch.com/2010/08/04/schmidt-data/

[TSAI11] Tsai, D., Jing, Y., Liu, Y., Rowley, H. A., Ioffe, S., & Rehg, J. M. (2011, November). Large-scale image annotation using visual synset. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 611-618). IEEE.

[TSOC04] Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004, July). Support vector machine learning for interdependent and structured output spaces. In Proceedings of the twenty-first international conference on Machine learning (p. 104). ACM.

[TONG03] Tong, F. (2003). Primary visual cortex and visual awareness. Nature Reviews Neuroscience, 4(3), 219-229.

[TORR03] Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. 2003. Context-based vision system for place and object recognition. In Proceedings of the Ninth IEEE international Conference on Computer Vision - Volume 2 (October 13 - 16, 2003). ICCV. IEEE Computer Society, Washington, DC, 273.

[TORR04] Torralba, A., Murphy, K.P., and Freeman, W.T. 2004. Contextual models for object detection using boosted random fields. NIPS.

[TORR07] Torralba, A., Murphy, K. P., and Freeman, W. T. 2007. Sharing Visual Features for Multiclass and Multiview Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 29, 5 (May. 2007), 854-869.

[TORR10] Torresani, L., Szummer, M., & Fitzgibbon, A. (2010). Efficient object category recognition using classemes. In Computer Vision–ECCV 2010 (pp. 776-789). Springer Berlin Heidelberg.

[TOUS12] Tousch, Anne-Marie ; Herbin, Stéphane ; Audibert, Jean-Yves. “Semantic hierarchies for image annotation: A survey”. Pattern Recognition, 2012, v45,p333-345.

U

[ULGE11] Ulges; Worring, M.; Breuel, T. “Learning Visual Contexts for Image Annotation From Flickr Groups” Ieee Transactions on Multimedia, 2011, v13.

V

[VALO82] De Valois, R. L., Yund, W., and Hepler, N. (1982b). The orientation and direction selectivity of cells in macaque visual cortex.Vision Research, 22:531–544

[VAN10] Van De Sande, K. E., Gevers, T., & Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9), 1582-1596.

[VAR03] Varma, M., & Zisserman, A. (2003, June). Texture classification: Are filter banks necessary?. In Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on (Vol. 2, pp. II-691). IEEE.

[VEDA09] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In International Conference on Computer Vision, 2009.

[VIOL01] Viola, Paul and Jones, Michael. Robust Real-time Object Detection. International Journal of Computer Vision (2001)

[VIOL04] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

[VISU14] Visual cortex. (2014, July 10). In Wikipedia, The Free Encyclopedia. Retrieved 10:20, July 11, 2014, from http://en.wikipedia.org/w/index.php?title=Visual_cortex&oldid=616452025

W

[WEB1] http://benanne.github.io/2014/04/05/galaxy-zoo.html

[WEB2] http://media.utoronto.ca/media-releases/u-of-t-neural-networks-start-up-acquired-by-google/

[WEB3] http://insidesearch.blogspot.com.es/2013/05/finding-your-photos-more-easily-with.html

[WEB4] https://www.facebook.com/yann.lecun/posts/10151728212367143

[WEB5] http://www.technologyreview.com/news/527301/chinese-search-giant-baidu-hires-man-behind-the-google-brain/

[WEB6] http://techcrunch.com/2013/10/23/yahoo-acquires-startup-lookflow-to-work-on-flickr-and-deep-learning/

[WEB7] http://www.usatoday.com/story/tech/2014/01/06/pinterest-nabs-visualgraph-for-image-recognition/4341747/

[WEB8] http://peekaboo-vision.blogspot.com.es/2010/11/restricted-boltzmann-machine-on-cuda.html

[WEBE00] Weber, M., Welling, M., and Perona, P. 2000. Unsupervised Learning of Models for Recognition. In Proceedings of the 6th European Conference on Computer Vision-Part I (June 26 - July 01, 2000). D. Vernon, Ed. Lecture Notes In Computer Science, vol. 1842. Springer-Verlag, London, 18-32.

[WEST11] Weston, J., Bengio, S., & Usunier, N. (2011, July). Wsabie: Scaling up to large vocabulary image annotation. In IJCAI (Vol. 11, pp. 2764-2770).

X

[XIAO10] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010, June). Sun database: Large-scale scene recognition from abbey to zoo. In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on (pp. 3485-3492). IEEE.

Y

[YANG08] L. Yang, R. Jin, R. Sukthankar, F. Jurie. Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition. Proceedings of Computer Vision and Pattern Recognition, 2008.

[YU14] Yu, J., Jeon, M., & Pedrycz, W. (2014). Weighted feature trajectories and concatenated bag-of-features for action recognition. Neurocomputing, 131, 200-207.

Z

[ZERN14] Zernike polynomials. (2014, June 17). In Wikipedia, The Free Encyclopedia. Retrieved 08:15, July 10, 2014, from http://en.wikipedia.org/w/index.php?title=Zernike_polynomials&oldid=613337044

[ZHAN07] Wei Zhang, G. Zelinsky, D. Samaras .Real-time Accurate Object Detection using Multiple Resolutions. Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (2007), pp. 1-8.

[ZHAN10] Zhang, S., Huang, J., Huang, Y., Yu, Y., Li, H., & Metaxas, D. N. (2010, June). Automatic image annotation using group sparsity. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 3312-3319). IEEE.

[ZHAN12] Zhang, D., Islam, M. M., & Lu, G. (2012). A review on automatic image annotation techniques. Pattern Recognition, 45(1), 346-362.

[ZHI10] Zhi Li ; Guizhong Liu ; Xueming Qian ; Chen Wang; Scale and rotation invariant Gabor texture descriptor for texture classification. Proc. SPIE 7744, Visual

Communications and Image Processing 2010, 77441T (August 04, 2010); doi:10.1117/12.863447.

[ZHOU13] Zhou, L., Zhou, Z., & Hu, D. (2013). Scene classification using a multi-resolution bag-of-features model. Pattern Recognition, 46(1), 424-433.

[ZHU06] Zhu, Q., Yeh, M., Cheng, K., and Avidan, S. 2006. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (June 17 - 22, 2006). CVPR. IEEE Computer Society, Washington, DC, 1491-1498.

Glosario

BOW

Algoritmo de agrupación de parches locales de imagen para generar un descriptor global de la misma. Más información en el apartado II.2.10.2

CRF

Acrónimo de Classical Receptive Field. Este concepto se refiere a la zona del campo receptivo de una neurona que actúa directamente sobre la salida de la neurona.

DoG

Método de computación utilizado en múltiples ámbitos. Su característica principal es la computación de una diferencia de funciones gaussianas.

ERF

Acrónimo de Extendend Receptive Field. Este concepto se refiere a la zona del campo receptivo de una neurona que no se encuentra dentro de la definición clásica.

F-measure

Métrica de evaluación de resultados de recuperación de información. Más información en el apartado III.3

GIST

Descriptor de imagen propuesto en [OLIV01] [TORR04] que representa el conjunto de una imagen en baja dimensionalidad. Más información en el apartado II.2.11

HOG

Descriptor de imagen propuesto en [DALA05] que representa los gradientes de una imagen mediante un histograma. Más información en el apartado II.2.5

HSV

Espacio de color compuesto por las componentes de tonalidad de color (H), saturación del color (S) y valor (V). Más información en el apartado II.2.6.1

K-means

Algoritmo de agrupación automática de puntos multidimiensionales. El valor de “K” define el número de grupos de puntos generados.

KNN

Acrónimo de K-Nearest Neighbors. Es un algoritmo de computa los “K” puntos más cercanos a un punto dado.

Lab

Espacio de color compuesto por las componentes de la luminancia (L) y los canales de oponencia de color definidos (a) y (b) .

LBP

Descriptor compacto de pequeños vecindarios de un píxel en base a cambios binarios de intensidad. Más información en el apartado II.2.9

Quadratic Program

Problema matemático de optimización de una función cuadrática en base a varias variables restringidas por funciones lineales.

Precision

Métrica de evaluación de resultados de recuperación de información. Más información en el apartado III.3

Recall

Métrica de evaluación de resultados de recuperación de información. Más información en el apartado III.3

RGB

Espacio de color compuesto por las componentes Rojo (R), Verde (G) y Azul (B).

RoG

Método de computación utilizado en múltiples ámbitos. Su característica principal es la computación de una división de funciones gaussianas.

SIFT

Descriptor local de parches de imagen propuesto en [LOWE04], invariante a rotación y escala. Más información en el apartado II.2.8.

SURF

Descriptor local de parches de imagen propuesto en [BAY08], invariante a rotación y escala, que busca mejorar la velocidad de SIFT.

SVM

Algoritmo de clasificación, acrónimo de Support Vector Machine. Se basa en la formulación matemática de la separación de un hyperplano ideal, permitiendo un cierto grado de falsas clasificaciones.

Tf-idf

Algoritmo de búsqueda inversa de documentos (inverse document frequency) en base a la frecuencia de sus términos (Term-frequency).