



Trabajo de fin de grado  
Grado en Ingeniería Electrónica

# Aplicación de algoritmos de *machine learning* a problemas en astrofísica

Autor:  
Torrente Fernández, Álvaro  
Director:  
Echanove Arias, Francisco Javier

Leioa, 24 de junio de 2022



# Índice

<b>1. Objeto y desarrollo</b>	<b>3</b>
<b>2. Introducción</b>	<b>5</b>
2.1. <i>Machine Learning</i> (ML)	5
2.2. <i>Machine learning</i> aplicado a la astronomía y la investigación espacial	7
<b>3. Técnicas y herramientas</b>	<b>11</b>
3.1. Redes neuronales artificiales (ANN)	11
3.2. Máquinas de vectores de soporte (SVM)	15
3.3. Histograma de gradientes orientados (HOG)	17
3.4. Herramientas de análisis del rendimiento	19
<b>4. Detección de obstáculos en imágenes obtenidas con <i>Hazcam</i></b>	<b>21</b>
4.1. Descripción y creación de la base de datos	21
4.2. Análisis con redes neuronales artificiales	25
4.2.1. Características basadas en los valores de los píxeles	25
4.2.2. Características basadas en el cálculo de histogramas de gradientes orientados (HOG)	32
<b>5. Detección de pulsares</b>	<b>37</b>
5.1. Descripción de la base de datos	37
5.2. Detección con redes neuronales	38
5.3. Máquinas de vectores de soporte (SVM)	38
<b>6. Clasificación de objetos estelares</b>	<b>41</b>
6.1. Descripción de la base de datos	41
6.2. Análisis con redes neuronales	42
6.3. Análisis con máquinas de vectores de soporte (SVM)	44
<b>7. Resumen y conclusiones</b>	<b>45</b>
7.1. Resumen	45
7.2. Conclusiones	45
7.3. Futuras actividades derivadas del TFG	46
<b>Referencias</b>	<b>47</b>
<b>A. Resultados de los experimentos</b>	<b>51</b>
A.1. Detección de obstáculos en imágenes obtenidas con <i>Hazcam</i>	51
A.2. Detección de pulsares	64
A.3. Clasificación de objetos estelares	65
<b>B. Especificaciones técnicas de los equipos empleados</b>	<b>67</b>





# 1. Objeto y desarrollo

El *machine learning* engloba una serie de técnicas que son de inmensa utilidad para la resolución de problemas de gran escala que engloban análisis de datos. Con éstas ya no sólo se agiliza el procesado de dichos datos si no que en muchos otros caso se abre la posibilidad de obtener soluciones a problemas que anteriormente no se había conseguido resolver.

El objetivo de este trabajo es evaluar las posibilidades de resolución de distintos problemas relacionados con la astronomía y la investigación espacial empleando técnicas de *machine learning*. Con este propósito, se trabaja con dos bases de datos de observaciones, una para la detección de púlsares y otra relacionada con la clasificación de objetos estelares. Además, se empleará una tercera de imágenes capturadas con las cámaras de detección de peligro del rover *Curiosity*, denominadas Hazcams.

Con tal propósito se emplean distintas técnicas para la construcción de modelos de predicción y clasificación de objetos, redes neuronales artificiales o algoritmos SVM; así como el preprocesado de las imágenes, *histograma de gradiente orientados (HOG)*. Asimismo, dentro de cada uno de estos modelos se evalúan el rendimiento del ajuste de distintos parámetros presentes en cada una de las herramientas: topología de red, tamaño de celda donde calcular el histograma de gradientes, etc.

El trabajo comienza realizando una pequeña *introducción* al *machine learning*, a su historia y las técnicas más empleadas, así como a su aplicación a la investigación espacial y la astronomía, y en especial a las tres bases de datos estudiadas. Posteriormente se explican las diferentes *técnicas y herramientas* seleccionadas para la creación de modelos de predicción para las bases de datos elegidas. Con ello se pasa al *entrenamiento* de dichos modelos para las tres bases de datos, explicándose en primer lugar el contenido de las misma y en tratamiento de datos realizado si lo hubiera. En todos los casos se realiza toda una serie de experimentos analizando tasas de clasificación y ajustando parámetros para nuevas pruebas posteriores. Finalmente, se extraerán una serie de *conclusiones* y se seleccionará el mejor modelo para cada caso.



## 2. Introducción

### 2.1. *Machine Learning* (ML)

Lo que se conoce hoy en día como *machine learning* (ML), o aprendizaje máquina o automático si se ignora el anglicismo, es uno de esos conceptos que es difícil de explicar o definir de forma certera y concreta. Esto se debe principalmente tanto a los numerosos autores que han trabajado en el campo como a las distintas formas de realizar dicha definición. No obstante, casi todas ellas giran en torno a la idea de que ésta se basa en el desarrollo de algoritmos informáticos que hacen que los ordenadores sean capaces de acceder a información y aprender por sí mismos. Básicamente, los equipos acceden a una serie de datos, buscan patrones u otro tipo de información que se repita a lo largo de las distintas muestras con el objetivo final de emplear todo ello en la realización de predicciones y toma de decisiones.

En este sentido, el origen del nombre reside en el hecho de que el fin último es que los ordenadores aprendan cómo usar los datos y la información que tienen a su disposición de forma que puedan continuar haciéndolo automáticamente, idealmente sin que los seres humanos tengan que intervenir en dicho proceso de aprendizaje. De esta forma, serían las propias máquinas las que reconocerían los cambios o actualizaciones que se tendrían que aplicar a su propio código para realizar la tarea que se encomendó en un principio de la mejor manera posible. Como se ha mencionado anteriormente, esta definición puede no ser del todo general ya que existen distintos métodos de *machine learning*, algunos no supervisados como el mencionado antes, pero otros en los que los seres humanos siguen jugando un papel importante en proceso de aprendizaje. Las principales características y diferencias entre ambos modelos se explican más adelante [1].

#### Historia del *machine learning*

Cuando se habla de técnicas de aprendizaje autónomo o aprendizaje máquina se tiende a pensar que se tratan de ideas muy modernas, ideas que no se habían ni concebido hasta la aparición de las nuevas tecnologías. No obstante, la realidad es que el propio concepto de que una máquina pudiera aprender no es para nada nuevo, de hecho, el término *machine learning* fue empleado por primera vez en 1959 por Arthur Samuel cuando trabajaba en IBM. Sin embargo, uno de los primeros trabajos en el campo fue el realizado por Frank Rosenblatt unos años antes, en 1957, con el desarrollo del *perceptrón* para reconocer imágenes. Mientras que es a Samuel a quien se le atribuye la invención del concepto, fue Nils J. Nilsson quien posteriormente desarrolló más esta idea en su libro *Learning Machines* [1].

Durante la posterior década de los 70 llegó el denominado *invierno de la inteligencia artificial*, donde se dieron pocos avances en el campo y en el ambiente reinaba un sentimiento de pesimismo, debido al inmenso coste computacional de los algoritmos. Sin embargo, ya en los 80, dicho coste se volvió manejable debido al aumento de la capacidad de cálculo, y supuso el empujón definitivo al campo. De esta forma, la época dorada del ML se dio a partir de los 90, dándose importantes hitos en el campo como el desarrollo del

TD-Gammon, una máquina capaz de jugar contra humanos al Backgammon (haciendo uso de una red neuronal artificial), o la derrota del campeón mundial de ajedrez Gary Kasparov a manos de la supercomputadora de IBM Deep Blue. Hoy en día es inmenso el número de aplicaciones que las técnicas de ML están teniendo, entre las que se encuentra el impulso al desarrollo en muchas áreas de investigación científica, como la elección de la mejor zona de aterrizaje o el camino más seguro a seguir en una misión espacial en Marte.

### Aprendizaje supervisado y no supervisado

En cuanto a los distintos métodos de aprendizaje automático que existen en la actualidad, dos son los de mayor importancia, el aprendizaje supervisado y el no supervisado. Cada uno de estos métodos tiene sus propios beneficios, usos y aplicaciones, así como fuentes de debate, debido a la idea antes mencionada de muchos autores de establecer como fin último del ML el aprendizaje totalmente autónomo sin ningún tipo de implicación por parte de los seres humanos; o, frente a estos, los diversos problemas éticos o morales que esto puede llegar a suscitar.

Por un lado, el *aprendizaje supervisado* consta de una serie de algoritmos que construyen modelos matemáticos a partir de un cierto conjunto de datos que contienen tanto las entradas como las salidas deseadas ante dichos valores de entrada. Estos últimos constituyen lo que se denominan "datos de entrenamiento", donde cada ejemplo se conoce como vector y al conjunto de todos ellos como la matriz de entrenamiento. De esta forma, el algoritmo los emplea de forma iterativa para ir optimizando una función que permita la predicción de la salida asociada a futuros nuevos valores de entrada, idealmente no incluidos entre el conjunto de valores de entrenamiento.

Por otro lado, en el *aprendizaje no supervisado* la única información que se le da a la máquina son los datos de entrada. De esta forma, es el propio equipo el que debe ir desarrollando grupos de similitud y creando datos de salida en función de lo que va descubriendo, de manera que pueda llegar a ser intuitivo y inventivo de la forma más efectiva posible. El trabajo de los desarrolladores reside por tanto en generar un grupo de valores de entrada lo suficientemente completo y robusto para que de lugar a unos resultados mejores que a los que ellos mismo podrían haber llegado [2].

### Técnicas de *machine learning*

Dentro del *machine learning* se engloban toda una serie de técnicas para la obtención de los *modelos de predicción*. Éstos engloban el conjunto de reglas y "conocimientos" extraídos de la base de datos estudiada y que servirán para identificar y clasificar las nuevas entradas. La serie de técnicas que se pueden emplear para la creación de dichos modelos se pueden clasificar en: algoritmos de *clasificación* y *regresión* para el aprendizaje supervisado, y algoritmos de *agrupamiento* (o *Clustering*) y *asociación* en el caso del aprendizaje no supervisado.

En el caso del aprendizaje supervisado, las *regresiones* tratan predecir o interpretar un valor numérico concreto basándose en observaciones anteriores mediante el empleo de funciones. Dependiendo de la forma de estas funciones las regresiones pueden ser lineales,

no lineales, de lógica difusa... Asimismo, los algoritmos de *clasificación*, como su propio nombre indica, se utilizan para categorizar elementos. Dentro de este tipo de algoritmos se pueden encontrar las redes neuronales artificiales (ANN), las máquinas de vectores de soporte (SVM), etc.

Por otro lado, en la categoría de los algoritmos de aprendizaje no supervisado se pueden encontrar en primer lugar los algoritmos de *agrupamiento o Clustering*, donde se analizan los datos de entrada en busca de patrones comunes con el objetivo de agruparlos. Además, también se incluyen los algoritmos de *asociación*, donde se buscan correlaciones entre datos identificando reglas de asociación entre los mismos.

## 2.2. *Machine learning* aplicado a la astronomía y la investigación espacial

Uno de los campos de mayor desarrollo científico en los últimos años es el de la investigación espacial y la astrofísica. Desde las primeras observaciones del cielo realizadas en la antigüedad, la interpretación y el análisis de los datos arrojados por las mismas ha sido una tarea de suma importancia y gran trabajo. Hoy en día hemos sido capaces de desarrollar instrumentos capaces de realizar todo tipo de observaciones del cielo que recopilan ingentes cantidades de datos sobre todos los objetos presentes en nuestro universo. Debido al volumen y complejidad del análisis de dicha información las técnicas de *machine learning* expuestas anteriormente son de gran utilidad para el desarrollo de estas tareas, ayudando a resolver problemas antes imposibles o acelerando enormemente el análisis de datos. Ejemplos de dichas aplicaciones en la rama de la astrofísica pueden ser los recogidos en [3], de entre los que destacan el análisis de la evolución de galaxias con *deep learning* o la clasificación de supernovas, entre otras.

Además de ayudar en el procesado de los datos, todo este conjunto de técnicas de ML también se pueden emplear en actividades de exploración espacial. Uno de los motivos de mayor peso para dicha aplicación es el hecho de que dichos equipos se encuentran a muchos kilómetros de distancia y la comunicación con ellos es muy lenta. La implementación de modelos de predicción para la toma de ciertas decisiones en satélites o rovers de exploración permite que éstos puedan desarrollar actividades como el encendido o apagado de equipos de abordó para ahorrar energía o la selección del rumbo y la conducción autónoma.

En este trabajo se crearán modelos de predicción empleando bases de datos relacionadas con tres temas dentro del campo de la investigación espacial y la astrofísica: imágenes de cámaras del rover *Curiosity*, la detección de señales provenientes de púlsares o la clasificación de objetos estelares.

### Misión *Mars Science Laboratory* y el rover *Curiosity*

Marte ha sido durante mucho tiempo objeto de interés humano, ya no solo por la cierta mística que tiene el planeta rojo, sino por el conjunto de características con las que cuenta el planeta, desde a ser uno de los planetas más accesibles desde la Tierra, hasta su historia y evolución. En palabras del administrador en funciones de la NASA, Steve

Jurczyk, "Marte es el planeta más parecido a la Tierra de nuestro sistema solar, y eso es realmente intrigante, porque al estudiar la historia geológica y climática del planeta y cómo evolucionó, también podemos aprender sobre cómo ha evolucionado la Tierra y cómo evolucionará en el futuro" [4]. Hoy sabemos que Marte es un planeta rocoso, al igual que la Tierra, formados prácticamente al mismo tiempo, pero con sólo la mitad del diámetro de la Tierra y una atmósfera mucho más fina. Además, cuenta con una superficie fría y desértica, donde la temperatura media anual ronda los 215 K [5].

Desde que se formó en 1993, el programa *Mars Exploration Program* de la NASA ha englobado en todo este tiempo un amplio conjunto de misiones cuyos objetivos eran la investigación y la colección de datos acerca del planeta rojo, como el *Mars Global Surveyor* en 1996 o los rovers *Spirit* y *Opportunity* lanzados 2003. Tras el éxito cosechado, en 2011 se puso en marcha la misión *Mars Science Laboratory*, la cual puso sobre la superficie de Marte en 2012 al rover *Curiosity*.

El rover *Curiosity* (Figura 2.1) tiene 3 metros de largo y 2,7 de ancho, un tamaño aproximadamente dos veces superior a los anteriores *Spirit* y *Opportunity*, y tiene un número de instrumentos más amplio y más sofisticados. En general, el conjunto de instrumentos se trata de cámaras de alta resolución, 17 en total, 12 de ellas con fines ingenieriles (navegación en la superficie) y 5 científicas. Entre las primeras se encuentran las *Hazcams* (*Hazard avoidance cameras*), que son cuatro parejas de cámaras en blanco y negro localizadas la parte inferior delantera y trasera del equipo. Cada una de las cámaras tiene un campo de visión de 120°, de forma que el rover puede "ver" lo que tiene hasta 3 metros por delante y en 4 metros de ancho (en la zona más amplia de la "cuña"). De esta manera, con ellas se capturan imágenes en 3D que evitan que el rover se choque con obstáculos, e incluso le permiten al rover tomar decisiones sobre cuál es el camino más adecuado para continuar adelante de forma autónoma [7].



**Figura 2.1:** Auto-foto del rover Curiosity en la superficie de Marte [6].

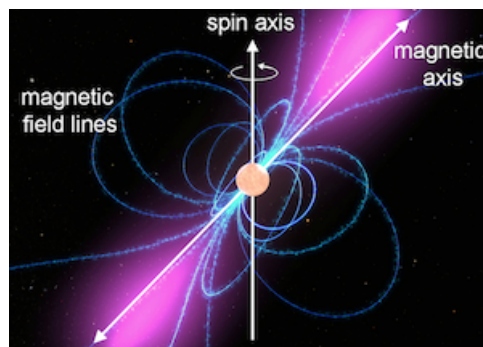
En esta última idea radica el principal interés en el unión de el *machine learning* con la exploración marciana. El hecho de contar con un equipo operando en una localización a la que la información tarda sobre 15 minutos en llegar hace que la realización de muchas tareas se complicada y peligrosa. Contar con software que permita al equipo tomar sus propias decisiones en función de los parámetros que se va encontrando e ir aprendiendo de éstas y sus consecuencias, es de un interés mayúsculo. De hecho, este tipo de técnicas ya se han empleado en la misión MSL. Por un lado, debido a la imposibilidad de controlar el proceso en tiempo real, la entrada en la atmósfera marciana y el aterrizaje del robot en la superficie fueron tareas realizadas por el ordenador de abordo, obteniéndose unos resultados magníficos. Por otro, el rover *Curiosity* cuenta con un software denominado *AutoNav*, responsable del análisis del terreno circundante y la posterior toma de decisión en cuanto al rumbo más seguro a tomar. De hecho, en el sol 376 de la misión (día marciano), el 27 de agosto de 2013, *Curiosity* anduvo de forma autónoma exitosamente durante 10 metros aproximadamente a través de un terreno que no se había podido catalogar como "seguro"

antes del inicio de la marcha.

## Detección de púlsares

Entre los objetos estelares más estudiados hoy en día se encuentran las estrellas de neutrones. Éstas se forman tras el colapso de una estrella supermasiva, donde en el proceso la región central de la estrella -el núcleo- se colapsa, aplastando todos los protones y electrones hasta convertirlos en neutrones. Este colapso da lugar a uno de los objetos más densos que se conoce: un objeto con la masa de un sol reducido al tamaño de una ciudad, con unos 20 kilómetros de diámetro. Dado que las estrellas de neutrones comenzaron su existencia como estrellas, se encuentran dispersas por toda la galaxia en los mismos lugares donde encontramos las estrellas. Muchas de ellas son indetectables porque simplemente no emiten suficiente radiación. Sin embargo, bajo ciertas condiciones, pueden observarse fácilmente. Éste es el caso de los *púlsares*.

Los púlsares son estrellas de neutrones en rotación en las que se observan pulsos de radiación, los cuales suelen darse a intervalos muy regulares que oscilan entre milisegundos y segundos. Asimismo, cuentan con campos magnéticos muy intensos que canalizan flujos de partículas a lo largo de los dos polos magnéticos, produciendo así haces de luz muy potentes. A menudo, el campo magnético no está alineado con el eje de giro, lo que hace que los haces de partículas giren con la estrella. Cuando el haz cruza nuestra línea de visión, vemos un pulso; de forma que desde la Tierra vemos cómo los púlsares se encienden y se apagan a medida que el haz barre nuestro planeta. Un esquema de un púlsar se muestra en la Figura 2.2. Estos objetos estelares son de gran interés científico debido a que su estudio ha arrojado luz en trabajos relacionados con el espacio-tiempo, el medio interestelar o los estados de la materia [8].



**Figura 2.2:** Esquema de un púlsar [8].

La radiación que emiten estos objetos estelares nos llega en forma de ondas de radio, que son detectables aquí en la Tierra. Es por ello por lo que la detección de púlsares se basa en la búsqueda de señales de radio periódicas. Sin embargo, cada púlsar tiene un patrón de emisión distinto, debido a que éste varía ligeramente con cada rotación. De esta manera, cuando se detecta una señal "candidata" a ser un púlsar, ésta se promedia en el tiempo, y en ausencia de información adicional, cada candidato podría describir un púlsar real. No obstante, en la práctica casi todas las detecciones están causadas por interferencias de radiofrecuencia (RFI) y ruido, lo que dificulta la detección de señales legítimas. Es por eso precisamente por lo que en la actualidad se utilizan técnicas de

*machine learning* para etiquetar automáticamente los púlsares candidatos y facilitar el análisis [9].

### Clasificación de objetos estelares

La forma que tenemos hoy en día de clasificar los objetos estelares es en función de su espectro de radiación. Simplemente atendiendo a la forma de dicho espectro y características derivadas del mismo se es capaz de clasificar galaxias, cuásares y estrellas. Según la ESA, hay aproximadamente un trillón de *estrellas* en el universo [10]. Durante mucho tiempo los astrónomos han intentado derivar propiedades de estas empleando lo único que tenían a su alcance sobre las mismas: la luz que emitían. Con el paso de los años, el desarrollo de nuevas tecnologías ha hecho que las observaciones sean cada vez mejores y capten mucha más información sobre estos objetos. Hoy en día somos capaces de saber la masa y la temperatura de una estrella simplemente a partir de sus espectros de radiación [11].

Además de estrellas, muchos otros objetos estelares también se pueden clasificar mediante el estudio y análisis de sus espectros. En el caso de las *galaxias*, al tratarse estas de un compendio de estrellas, nubes de gas, planetas, polvo cósmico, materia oscura y energía, los espectros de emisión serán un promedio de los espectros de todos los objetos que se encuentran en el interior de dicha galaxia. Éstos nos dan información valiosa sobre la composición y edad de la misma. Sin embargo, a día de hoy no existe una clasificación formal de galaxias por su espectro, pero sus espectros sí que se pueden distinguir de los provenientes de otro tipo de elementos del cosmos [12].

Finalmente, los *cuásares* se tratan de objetos estelares que son alimentados por agujeros negros supermasivos que brillan con tanta intensidad que eclipsan la luz de las galaxias que los contienen. Se piensa que puedan representar el proceso inicial de transformación de las galaxias. Las primeras detecciones de este tipo de objetos se dieron en los años 30 cuando se estudiaban las interferencias de las comunicaciones telefónicas. El estudio de los mismos, al igual que en el caso de las estrellas y galaxias, también se puede realizar atendiendo a sus espectros de radiación. En este proceso toma un papel fundamental el *machine learning*, debido a que atendiendo a una serie de características espectrales bien definidas, se puede obtener un modelo que sea capaz de clasificar objetos estelares atendiendo a la forma de los espectros de las observaciones realizadas [13].



## 3. Técnicas y herramientas

### 3.1. Redes neuronales artificiales (ANN)

A la hora de idear sistemas que sean capaces de procesar información y actuar en consecuencia, uno de los mejores ejemplos se encuentra en los seres humanos. Es esta precisamente una de las causas que impulsaron la creación de las *redes neuronales artificiales (ANN)*, las cuales son sistemas especialmente diseñados para tratar de imitar la forma en que el cerebro humano procesa información y aprende de ella. El sistema nervioso humano contiene células que se denominan neuronas, las cuales están conectadas entre sí mediante el mecanismo de comunicación que se denomina sinapsis. La intensidad de las conexiones sinápticas suele cambiar en respuesta a estímulos externos, y es este cambio precisamente la forma en que se produce el aprendizaje en los organismos vivos. En 1943 por Warren S. McCulloch y Walter Pitts [14] introdujeron por primera vez algoritmos sustentados por la idea de emplear modelos matemáticos basados en las redes neuronales biológicas, que darían lugar a las *redes neuronales artificiales (ANN)*. No obstante, como se ha mencionado anteriormente, no fue hasta muchos años posteriores, junto al desarrollo de las nuevas tecnologías, cuando se comenzara a explotar todo su potencial [15].

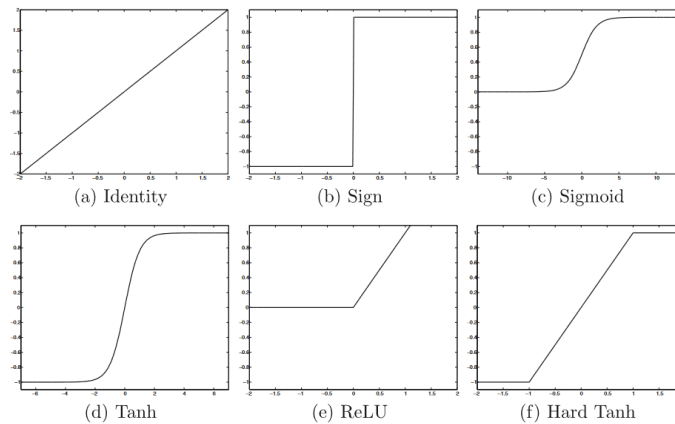
En las redes neuronales artificiales se cuenta con unidades computacionales a las que, por similitud con sus análogas biológicas, también se les denomina *neuronas*. Éstas están conectadas entre sí por *pesos*, que realizan el papel de la fuerza de las conexiones sinápticas entre neuronas en los organismos biológicos. De esta manera, cada entrada a cada neurona proviene de una conexión que está ponderada por un peso, de forma que no sólo las entradas sino que también el valor éstos tiene una gran importancia en el cálculo del valor que albergará ésta. Entre los elementos constituyentes de estas unidades computacionales (o neuronas) se encuentran los siguientes:

- **Entrada ( $x_i$ ):** Valor que recibe la neurona, ya sea proveniente de otra neurona o directamente de la entrada.
- **Peso ( $\omega_i$ ):** Valor que pondera la entrada que recibe la neurona y que se irá ajustando en el proceso de entrenamiento. Además, se puede interpretar como un parámetro característico de la importancia de la conexión a la que está asociado.
- **Bias (u *offset*) ( $b$ ):** Parámetro que da cuenta de la parte invariante de la predicción, de forma que se tiene siempre en cuenta, además de la entrada con el peso, independientemente del valor de la entrada. De esta manera, este parámetro indica la facilidad con la que se activará o no una neurona.
- **Función de propagación:** Como su propio nombre indica, se encarga de propagar la información de unas neuronas a las siguientes, de forma que establece el valor de una neurona a partir de sus entradas. La forma más común de representar esta matemáticamente es:

$$\sum_{i=1}^N \omega_i x_i + b \quad (1)$$

siendo  $N$  el número de conexiones entrantes a la neurona en cuestión,  $\omega_i$  los pesos de las conexiones,  $x_i$  el valor de las entradas y  $b$  el *bias*.

- **Función de activación ( $f$ ):** Es la función encargada de decidir si se activa la neurona o no en función de la información que le llegue ( $x_i$ ), de los pesos ( $\omega_i$ ) y del bias ( $b$ ). Son varias las funciones que se suelen emplear como funciones de activación, entre las cuales las más frecuentes son: la función signo, la sigmoide, la arco-tangente hiperbólica, ReLU... La forma de estas se recoge en la Figura 3.1.



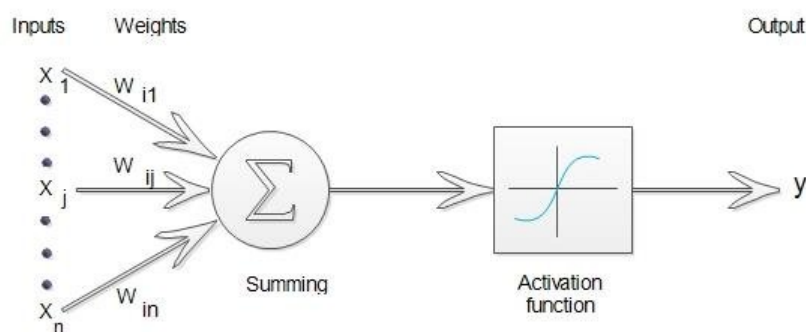
**Figura 3.1:** Forma funcional de las funciones de activación más empleadas [15].

Teniendo en cuenta todos los citados elementos, de forma matemática la salida de una neurona  $y$  se puede expresar de la forma recogida en la ecuación (2).

$$y = f\left(\sum_{i=1}^N \omega_i x_i + b\right) \quad (2)$$

siendo  $f$  la función de activación,  $N$  el número de conexiones entrantes a la neurona en cuestión,  $\omega_i$  los pesos de las conexiones,  $x_i$  el valor de las entradas y  $b$  el *bias*.

El esquema de una neurona donde se señalan los elementos arriba descritos se muestra en la Figura 3.2.



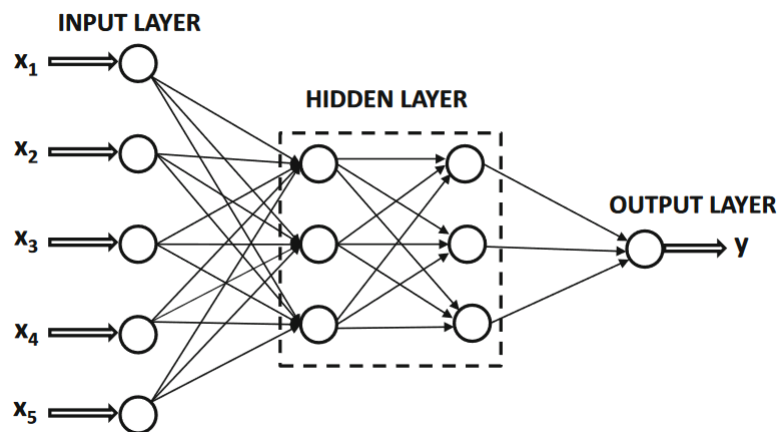
**Figura 3.2:** Esquema de la forma de una neurona de una red neuronal artificial [16].

Dentro de los numerosos tipos de estructuras de redes neuronales que existen en la actualidad, una de las más empleadas son las denominadas *redes feedforward*. En éstas, atendiendo al propio nombre, la información se va propagando por la red, desde las neuronas de entradas hasta las de salida. Dicha información fluye por la red a través de las

denominadas *capas*. De esta manera, la estructura de una red neuronal se divide como sigue:

1. **Capa de entrada:** Es la primera capa de la red, visible desde fuera. Cuenta con un número de neuronas igual al número de valores de entrada con los que trabaja la red.
2. **Capas ocultas:** Grupos de neuronas que siguen a la de entrada y son invisibles desde fuera. Estas pueden estar formadas por cualquier número de neuronas y la red puede contar con una capa oculta, varias o ninguna.
3. **Capa de salida:** Conjunto de neuronas que proporcionan los valores de las salida. De esta forma, el número de neuronas en esta capa será igual al número de salidas que proporcione la red.

Un esquema de dicha arquitectura se muestra en la Figura 3.3.

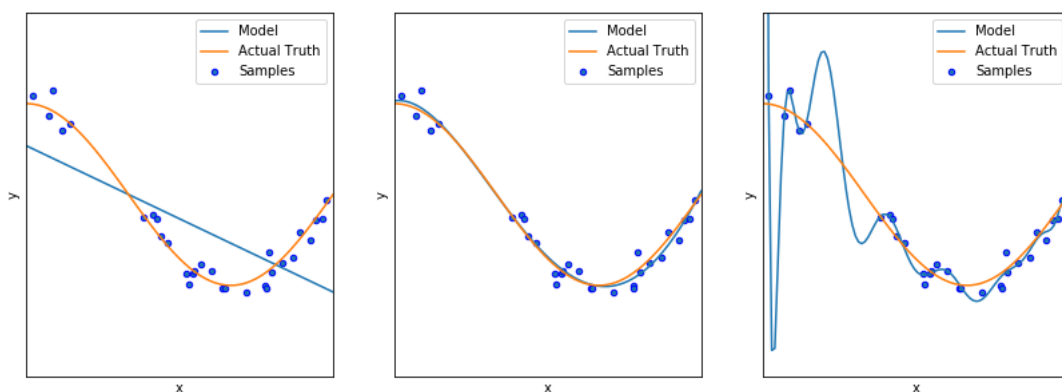


**Figura 3.3:** Esquema de una red neuronal artificial [15].

Con todo esto, el objetivo principal de una red neuronal *feedforward* es conseguir calcular una función de los datos de entrada propagando los valores calculados desde las neuronas de entrada, pasando por las capas intermedias (u ocultas) hasta las de salida, utilizando para ello los pesos como parámetros intermedios. Por tanto, el proceso aprendizaje consiste en que al presentarle una entrada y sabiendo cuál debe ser la salida, se ajustan los pesos de las distintas conexiones de forma que se obtenga dicho valor de salida para esa entrada concreta. De esta manera, los pesos de la red se ajustan en respuesta a los errores de predicción, es decir, a lo mucho o poco que se ha alejado la respuesta de la red de la salida real para una entrada dada. Posteriormente, con otro valor de entrada cuyo valor a la salida también es conocido, se realiza de nuevo dicho proceso sobre los pesos ajustados para la entrada anterior. El objetivo es que con el paso de las muestras y el continuo ajuste de los pesos de la red se vayan consiguiendo salidas cada vez más próximas a las deseadas. A este proceso y a la capacidad de la red para calcular una salida a partir de una entrada no vista anteriormente se denomina *generalización del modelo*. Ésta es la gran ventaja y utilidad del empleo de técnicas de *machine learning*, la capacidad de generalizar lo aprendido con ejemplos a nuevas entradas no vistas anteriormente.

Para su implementación en este trabajo se emplea el programa *MatLab* y la biblioteca *Deep Learning Toolbox*, dentro de la cual se encuentra la herramienta *NNTrainingTool* que se encarga de entrenar redes neuronales. De cara a la realización de los entrenamientos son varios los parámetros de dicha herramienta a configurar:

- **Método de entrenamiento:** Se elige el método del *gradiente conjugado escalado*, ya que es el que normalmente mejores resultados arroja en este tipo de tareas.
- **Función de división de las muestras:** Se trata de la forma en la que la herramienta elige qué muestras emplear para entrenar y validar la red. Se elige la función *aleatoria*.
- **Modo de división de las muestras:** Se elige el modo por defecto, es decir, *sample*.
- **Función de rendimiento:** Establece la forma en la que se va viendo la mejora o no de la red tras cada iteración. En este caso se elige evaluar el error como *mínimos cuadrados (mse)*.
- **Ratio de muestras:** Porcentaje de la muestras empleadas para cada uno de los tres procesos de entrenamiento:
  - **Ratio de muestras de entrenamiento:** Marca el porcentaje de las muestras de entrada empelado para entrenar la red. Dicho proceso de entrenamiento consiste en ir empleando las muestras seleccionadas para ajustar los distintos parámetros de la red, como los pesos etc., en función del método y la función de rendimiento elegidas.
  - **Ratio de muestras de validación:** Porcentaje de las muestras de entrada empelado para validar la red. Ésta validación se encarga de evitar el sobreentrenamiento del modelo. Una red sobreentrenada obtiene una función que clasifica a la perfección los datos de entrenamiento pero no se ajusta al resto. Un ejemplo se muestra en la Figura 3.4.



**Figura 3.4:** Ejemplo de un modelo infraentrenado (izquierda), uno bien entrenado (centro) y uno sobreentrenado (derecha) [17].

- **Ratio de muestras de prueba:** Porcentaje de las muestras de entrada empelado para probar la red. Se emplean dichas muestras para verificar el comportamiento antes datos de entrada no "vistos" durante el entrenamiento.

- **Máximo de fracasos de validación:** Fija el número máximo de iteraciones en las que, si no mejora el error, se para el entrenamiento. Se establece un valor para esta cantidad de 100.
- **Máximo de número de iteraciones:** Establece el número máximo de iteraciones para el entrenamiento. Se fija este parámetro en 1000.

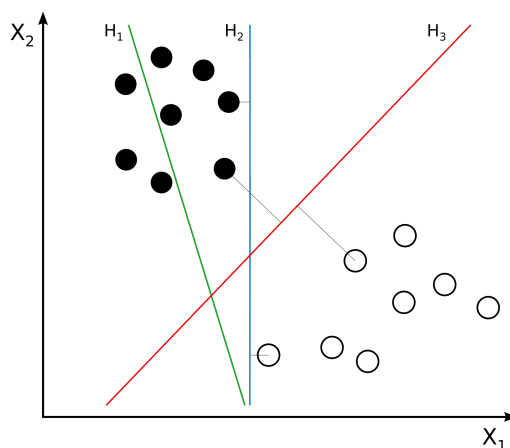
Debido a que se ha elegido una selección aleatoria de las muestras que se emplearán para entrenar la red así como los valores de los pesos iniciales, con el objetivo de minimizar el efecto de esta aleatoriedad en los parámetros que medirán el rendimiento de los algoritmos, es necesario realizar una pequeña estadística. Es decir, se repetirá cada entrenamiento un número de veces  $N_{promedio}$  y se realizará la media de los resultados, ya sean tasa de clasificación, tiempo de entrenamiento o la matriz de confusión. En este caso, con el objetivo de obtener unos buenos resultados pero en unos tiempos de entrenamiento razonables, se elige un valor de  $N_{promedio} = 5$ .

### 3.2. Máquinas de vectores de soporte (SVM)

Las *máquinas de vectores de soporte* (SVM del inglés *Support Vector Machines*) son uno de los algoritmos de clasificación binaria supervisada más empleados. Aunque las primeras versiones del mismo llevan en circulación desde 1963, la versión empleada actualmente fue propuesta por Cortes y Vapnik en 1995 [18]. Para un correcto estudio del conjunto de datos de entrada es esencial realizar una buena análisis de las características más importantes presentes en el mismo. Con este fin se emplean los *descriptores de características*, los cuales consisten en realizar una representación de la información contenida en un conjunto de datos mediante la extracción de la información más relevante e ignorar toda aquella que no lo es. Esta información relevante se organiza y se introduce a la red como un vector de características.

Dado un vector de características de  $n$  dimensiones, se pueden imaginar estas como un espacio de  $n$  dimensiones, donde los algoritmos SVM buscan la forma de establecer fronteras entre grupos que definan clases. Debido a la dificultad de representar espacios de  $n$  dimensiones, como ejemplo se supone un vector de solamente dos dimensiones. De esta manera, se tiene un espacio bidimensional donde cada punto está asociado a la clase A o B. En la Figura 3.5 se muestra una representación de puntos pertenecientes a las dos clases en este espacio bidimensional. Durante el entrenamiento del modelo la tarea consiste principiante en ir diciéndole al algoritmo las coordenadas de cada punto de los valores de entrada en este espacio, además de la clase, A o B, a la que pertenece.

A partir de aquí, existen diferentes algoritmos que tratan de buscar la forma de separar los elementos correspondientes a estas dos clases en el espacio. Los algoritmos *SVM lineales* tratan de realizar esta tarea buscando una línea recta que establezca la frontera entre los elementos pertenecientes a una clase u a otra. En el ejemplo mostrado en la Figura 3.5 H1, H2 y H3 representan dichas fronteras. En el caso de H1 esta recta no consigue separar ambas clases, por lo que no es un buen clasificador. Por otro lado, tanto H2 como H3 consiguen clasificar bien los elementos. Sin embargo, a primera vista H3 parece un mejor clasificador que H2, ya que este parece representar mejor lo que se entendería como frontera



**Figura 3.5:** Ejemplo de un espacio bidimensional con elementos perteneciente a una clase A (puntos negros) y B (puntos blancos) [19].

entre ambas clases. Se dice que este clasificador separa *máximamente* ambas clases, debido a que la distancia de esta recta a las muestras de las dos clases es la máxima. Con todo ello, una vez encontrada dicha recta de separación entre clases, cuando se quiere evaluar la pertenencia a una clase u otra de una muestra nueva no empleada para el entrenamiento, simplemente se debe ver a qué lado de la recta pertenecen las coordenadas de la muestra en el espacio de características.

El proceso explicado anteriormente se puede generalizar para vectores de características de más de dos dimensiones. De esta forma, en 3 dimensiones el algoritmo SVM buscará un plano de separación entre los puntos pertenecientes a cada clase. Para  $n$  dimensiones, el algoritmo buscará el hiperplano correspondiente.

El ejemplo mostrado en la Figura 3.5 contaba con objetos de clases que eran claramente separables por una recta. Pero, ¿qué ocurre si esto no es posible? En este caso el algoritmo buscará también el hiperplano que mejor separe las clases, pero esta vez lo hará resolviendo el problema de optimización que trata de aumentar la distancia del hiperplano de las dos clases al tiempo que intenta asegurarse de que muchos ejemplos de entrenamiento se clasifican correctamente. Es importante recalcar cómo la distancia entre muestras está directamente relacionada con el margen de error del clasificador a la hora de categorizar nuevos elementos. Este compromiso entre distancia y clasificaciones en el entrenamiento se controla mediante un parámetro llamado  $C$ . Si dicho parámetro es pequeño, se elige un hiperplano de margen grande a expensas de un mayor número de clasificaciones erróneas. Por el contrario, cuando  $C$  es grande, se elige un hiperplano más cercano a las muestras pero que clasifique correctamente muchos más ejemplos. En la práctica, este valor de  $C$  se elige de forma que se obtenga el mejor rendimiento en cada caso a la hora de validar el modelo [19].

### Clasificadores SVM multiclase

Los clasificadores explicados anteriormente creados a partir de algoritmos SVM únicamente realizan clasificaciones de datos pertenecientes a dos clases diferentes. Ahora bien, éste mismo método se puede extender para poderse emplear para la clasificación elemen-

tos multiclase. En la actualidad son dos los principales métodos de clasificación multiclase que se pueden encontrar: *One-vs-Rest (OvR)/One-vs-All (OvA)* ó *One-vs-One (OvO)*. En todo este tipo de algoritmos se dividen los datos multiclase como datos de dos clases para que los mismos algoritmos de clasificación binaria puedan aplicarse.

En el caso de los primeros *One-vs-Rest (OvR)/One-vs-All (OvA)*, como se puede entender del propio nombre, se van agrupando los elementos de las diferentes clases de forma que se tengan dos grupos, uno de una clase de las originales y el otro del resto. Como ejemplo se puede tomar el caso de tres clases (A), (B) y (C), de manera que las agrupaciones antes mencionadas quedarían: (A), (B,C); (B), (A,C); y (C), (A,B). De esta forma, empleando los algoritmos anteriores para una clasificación binaria, se crean clasificadores que distingan entre ambos grupos. Se realizan las predicciones para cada uno de los tres enfrentamientos, de manera que el que mejores resultados arroje para la clase individual, es a ésta a la que pertenece dicho dato de entrada.

Otra forma de proceder puede ser la siguiente. Una vez creado el clasificador para esas dos nuevas clases binarias, si se obtiene que el elemento se clasifica como del grupo de la clase única, ya está, dicho dato pertenece a esa clase. Por otro lado, si se clasifica como perteneciente al grupo de dos clases, se emplea un nuevo clasificador binario para estas dos clases. Finalmente, una vez obtenido a cuál de las dos clases de este subgrupo pertenece este último dato, ya se ha clasificado dicho elemento en una de las tres clases originales.

Por otro lado, los algoritmos *One-vs-One (OvO)* funcionan de forma similar a los ya explicados *One-vs-Rest (OvR)/One-vs-All (OvA)*, en el sentido de que también basan su funcionamiento en la agrupación de clases para la creación de clasificadores binarios. Sin embargo, en este caso se van enfrentando los grupos de clases uno a uno, es decir, se crean modelos de clasificación para cada pareja de clases posibles dentro de los datos a analizar. Siguiendo con el ejemplo de tres clases puesto antes (A), (B) y (C), empleando la técnica de *One-vs-One (OvO)* dichas parejas quedarían: (A), (B); (A), (C) y (B), (C). Una ventaja clara de este tipo de técnicas frente a las explicadas antes es que en esta ocasión el número de clasificadores a entrenar para  $N$  clases es  $\frac{N(N-1)}{2}$ , frente a los  $N$  modelos requeridos para *One-vs-Rest (OvR)/One-vs-All (OvA)*.

Para la implementación de los algoritmos SVM en este trabajo se empleará el programa *MatLab* y la biblioteca *Statistics and Machine Learning Toolbox*, donde se encuentran las herramientas para crear y probar un clasificador SVM.

### 3.3. Histograma de gradientes orientados (HOG)

Las técnicas de reconocimiento de imágenes consisten en esencia en tomar como entrada una imagen y dar a la salida una descripción de su contenido. Con tal propósito son de vital importancia los descriptores de características introducidos en 3.2. Estos descriptores no son útiles para la visualización de las imágenes como tal, pero son de suma importancia para que el ordenador entienda lo que está "viendo" en esa imagen con mayor facilidad, debido a que eliminan la información superflua y sin importancia [20].

Un tipo de estos descriptores son los *histogramas de gradientes orientados*. La técnica fue introducida por primera vez en 2005 por Navneet Dalal y Bill Triggs para la detección

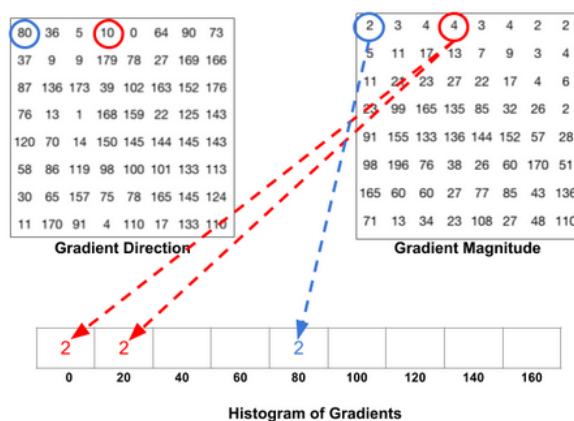
de peatones en imágenes [21] y se basa en la idea de que la apariencia de puede describir de forma efectiva mediante la distribución (histograma) de las direcciones de los bordes (gradientes orientados). Un resumen de los pasos a seguir en el proceso de cálculo del descriptor HOG de una imagen sería el siguiente:

1. **Cálculo de los gradientes verticales y horizontales de la imagen.** Se calculan los gradientes en las direcciones cartesianas  $x$  e  $y$ , obteniéndose los valores  $g_x$  y  $g_y$  respectivamente. Posteriormente se obtienen la magnitud y la dirección del gradiente para cada punto mediante las siguientes relaciones.

$$g = \sqrt{g_x^2 + g_y^2} \quad (3)$$

$$\Theta = \arctan\left(\frac{g_y}{g_x}\right) \quad (4)$$

2. **Cálculo del histograma de gradientes en una celda de  $n \times n$ .** Se divide la imagen en celdas de  $n \times n$  píxeles. La importancia de este paso reside en la representación de forma compacta de la información dicho grupo de píxeles. Una vez calculados los gradientes (magnitud y dirección) se crea un histograma de dichos valores dentro de la zona de  $n \times n$ . Este histograma consta de 9 casillas, correspondiente a direcciones de  $0, 20, 40, \dots, 160^\circ$  y en cada una de estas se va introduciendo las magnitudes correspondientes atendiendo a una serie de criterios. Si la dirección coincide directamente, se introduce en el grupo correspondiente. Si no, se reparte el valor de la magnitud de forma proporcional a la distancia que separa el valor de la dirección de los dos bloques entre los que se encuentra. En la Figura 3.6 se muestra un ejemplo.



**Figura 3.6:** Ejemplo de la creación del histograma de una zona de  $8 \times 8$  píxeles. El caso azul corresponde con la coincidencia directa del valor; mientras que el rojo, al estar  $10^\circ$  a mitad de camino entre  $0^\circ$  y  $20^\circ$ , se divide entre 2 la magnitud que va a cada casilla [20].

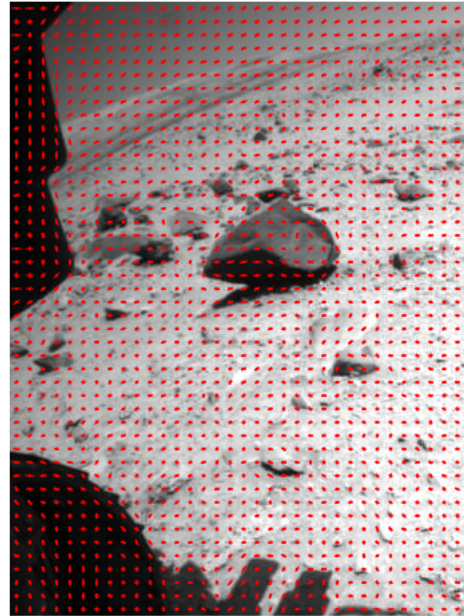
3. **Normalización de los bloques.** Debido a que los gradientes calculados son sensibles a la iluminación general de la imagen, con el objetivo de hacer que el descriptor sea independiente de estos efectos, se normalizan los valores. Esto se realiza con zonas de  $2n \times 2n$  píxeles, ya que se comprueba que de esta manera se obtienen mejores resultados. Estas zonas tienen cuatro histogramas que se concatenan en un vector de  $36 \times 1$  que es normalizado.



4. **Cálculo del vector de características del HOG.** Se vuelven a concatenar los resultados de la normalizaciones del paso anterior, de forma que se obtiene un vector para todas las zonas de la imagen. Un ejemplo del resultado de este proceso se muestra en la Figura 3.7.



(a) Imagen de Hazcam RAW.



(b) Imagen de Hazcam con HOG.

**Figura 3.7:** Ejemplo de imagen con histograma de gradientes orientados (HOG). Se muestra la zona C de la imagen con ID HAZ-CAM\_2FLB\_443568429EDR\_F0251070FHAZ00302M1. Se puede apreciar en (b) como los gradientes "rodean" los objetos (rocas) presentes en la imagen.

### 3.4. Herramientas de análisis del rendimiento

En cuanto a la medida del rendimiento de los distintos algoritmos y procesos de construcción de los modelos de predicción antes mencionados, se hará uso de los siguientes parámetros:

1. **Matriz de confusión ( $M_C$ ):** Como su nombre bien indica, se trata de una matriz cuadrada de dimensión  $n_{clase} \times n_{clase}$ , donde  $n_{clase}$  es el número de clases a identificar, y que se emplea para visualizar las distintos aciertos o confusiones de membresía del modelo a la hora de clasificar una entrada como perteneciente a una clase u otra. Ésta se estructura de manera que cada una de las filas representa cada clase real mientras que cada columna representa las clases predichas. Por tanto, para cada predicción hecha se le va sumando 1 a la posición correspondiente al par de valores  $(i_{real}, j_{predicha})$ , donde  $i_{real}$  es el número de la clase real a la que pertenece la entrada y  $j_{predicha}$  es el número de la clase predicha.

2. **Tasa de clasificación ( $T_{clas}$ ):** Se trata de una tasa en forma de porcentaje que representa el número de elementos sobre el total que se han clasificado correctamente. De forma analítica, ésta se puede definir como:

$$T_{clas} = \frac{\text{Tr}(M_C)}{N_{elementos}} \cdot 100 \quad (5)$$

donde  $M_C$  es la matriz de confusión,  $\text{Tr}(M_C)$  su traza y  $N_{elementos}$  es el número total de elementos clasificados.

3. **Tiempo de entrenamiento ( $t_{entr}$ ):** Es el tiempo medio de entrenamiento para cada uno de los modelos de predicción calculados. Por darle a dicho valor algo de contexto se incluyen las características técnicas del equipo con el que se han realizado los entrenamientos en la Tabla B.1 del Anexo B.

## 4. Detección de obstáculos en imágenes obtenidas con *Hazcam*

### 4.1. Descripción y creación de la base de datos

En primer lugar, para la obtención de las imágenes con las que se va a trabajar se accede a la base de datos de la misión MSL de la NASA y se obtienen el listado de las observaciones realizadas por el instrumento *Hazcam* del rover *Curiosity* desde el inicio de la misión. El conjunto de las bases de datos de las misiones de la NASA es conocida como *NASA Planetary Data System (PDS) Imaging Node*, que para la misión MSL está en la dirección [22]. Los datos correspondientes a *Hazcam* se pueden encontrar en [23], ordenadas por carpetas correspondientes a cada sol marciano.

Como se mencionó anteriormente 2.2, el instrumento MSL *Hazcam* está compuesto por 8 cámaras: 4 en la parte delantera del rover, a las que se denomina *Front Hazcams*; y otras 4 en la parte trasera, denominadas *Rear Hazcams*. A su vez, en cada uno de los grupos éstas se reparten en 2 en el lado izquierdo, *Left-A* y *Left-B*; y otras 2 en el lado derecho, *Right-A* y *Right-B*. Con ello, se tienen las siguientes IDs o identificadores para las cámaras: *FLA*, *FLB*, *FRA*, *FRB* para las frontales y *RLA*, *RLB*, *RRA*, *RRB* para las traseras. Estos IDs conforman las primeras 3 letras del *filename* de cada archivo.

Las imágenes con las que se va a trabajar son observaciones de tipo RAW, es decir, imágenes sin procesar, y de ahí el código *EDR* en el nombre de cada archivo. Dentro de los archivos *EDR* se encuentran dos tipos: *EDR\_T*, de *Thumbnail*, es decir, imágenes de muy baja resolución que solo se emplean para previsualización y ahorrar tiempo de transferencia; y *EDR\_F*, de *Full* y que incluyen la imagen a resolución completa. Asimismo, Entre el ID de la cámara y el *EDR\_F*, hay un número que da cuenta del valor del parámetro *Spacecraft Clock Time (SCLK)*, que es el tiempo en segundos transcurrido desde el inicio de la misión y que es identificativo del momento en el que se tomó la imagen. Las imágenes están en blanco y negro, y cuentan con una resolución de  $1200 \times 1200$  píxeles.

Con todo ello, se accede a [23] y se descargan el total de 6944 imágenes disponibles hasta el momento. Sin embargo, para la realización de la serie de actividades que se desarrollarán en el trabajo, es suficiente contar con unas 1000 observaciones, las cuales son elegidas siguiendo los siguientes criterios:

1. Deben ser imágenes en las que se vea la superficie que tiene delante/detrás el rover de forma clara, con el objetivo de que el análisis de la misma se refiera a dicho paisaje y no se vea obstruido por el brazo robótico del equipo, por ejemplo.
2. Deben ser imágenes lo más distintas posible entre ellas, de forma que se tenga una colección de situaciones lo más extensa posible.

Una vez elegidas las 1000 imágenes, debido a la naturaleza de las mismas y el propósito de la información que contienen, se dividen las distintas zonas presentes de la siguiente manera:

- (A). **Zona inferior del rover:** Debido a la distorsión de la imagen introducida en las capturas fruto de la lente de "ojo de pez", prácticamente la mitad inferior completa de la imagen se corresponde con la parte del terreno que se encuentra debajo del rover. Por tanto, la información aquí presente ya no es relevante.
- (B). **Paisaje lejano y cielo:** Al igual que en la parte inferior de la imagen, la parte superior, que engloba el paisaje lejano del rover y el cielo, aparece distorsionada. En cuanto a lo segundo, no nos interesa de cara al análisis de la peligrosidad del terreno, mientras que en el caso del paisaje lejano, la razón de su descarte es la baja resolución, lejanía y deformación de los elementos presentes en él.

La zona restante de la imagen será la que se estudie. Ésta a su vez se separa en 3 partes más pequeñas y de resolución  $340 \times 460$ , siendo éstas:

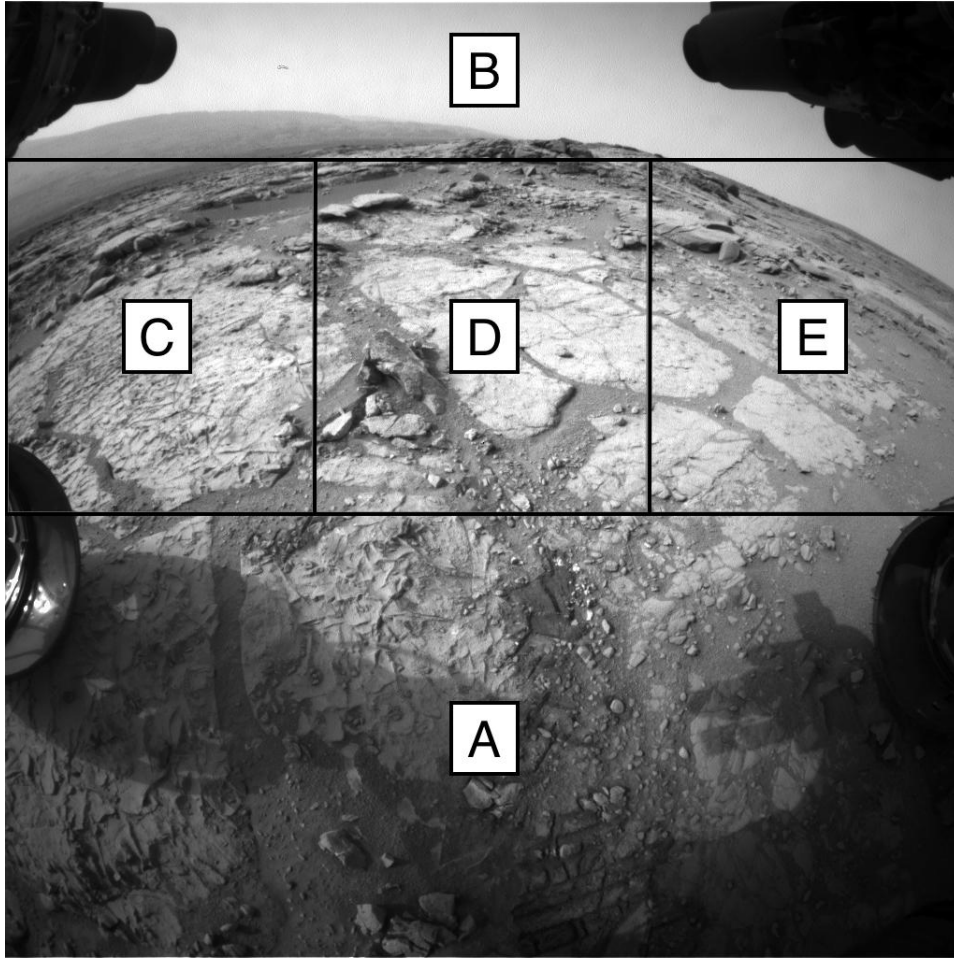
- (C) **Zona de estudio - Izquierda:** Parte de la imagen correspondiente a la *zona izquierda*.
- (D) **Zona de estudio - Centro:** Parte de la imagen correspondiente a la *zona central*.
- (E) **Zona de estudio - Derecha:** Parte de la imagen correspondiente a la *zona derecha*.

Un ejemplo del tipo de imágenes que se toman con el instrumento *Hazcam* se muestra en la Figura 4.1, donde en 4.1a se muestra la identificación dentro de la misma de las distintas zonas enumeradas anteriormente y en las Figuras 4.1b, 4.1c y 4.1d las imágenes resultantes con las que se trabajará.

Señalar que al recortar las imágenes se ha triplicado el tamaño de la base de datos, ya que de cada foto se acaban extrayendo 3 zonas. De esta manera, la base de datos con la que se trabajará tiene 3000 muestras. Una vez obtenidas las imágenes se procede al etiquetado de las mismas, es decir, a la asignación a cada una de las imágenes de un valor que sea característico de lo que representa esa imagen en base a lo que se quiere analizar, en este caso, el nivel de peligro debido a obstáculos en el paisaje. Por tanto, dicha labor consiste en ir visualizando manualmente una a una las 3000 muestras e ir asignándole a cada una un nivel de peligrosidad de entre los que sigue:

- (0) No hay peligro aparente a la vista.
- (1) Hay pequeñas rocas y obstáculos en el terreno que difieren del terreno llano pero a priori sin peligro aparente.
- (2) Hay rocas y obstáculo de pequeño y medio tamaño que requieren de cierta precaución.
- (3) Hay rocas y obstáculos de mediano tamaño que entrañan peligro.
- (4) Hay rocas y obstáculos de gran tamaño que entrañan peligro.
- (5) Hay obstáculo de gran tamaño que entrañan el nivel de peligro máximo.
- (10) Hay presencia de arena en la imagen, lo que entraña un gran nivel de alerta.

Se muestran ejemplo de una imagen correspondiente a cada uno de los niveles aquí expuestos en la Figura 4.2.



(a) Imagen capturada con el instrumento *Hazcam Front B* en la que se han identificado las distintas zonas dentro de la misma.



(b) Imagen resultante de la zona C de 4.1a.

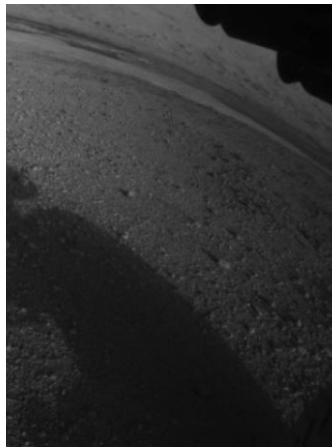


(c) Imagen resultante de la zona D de 4.1a.

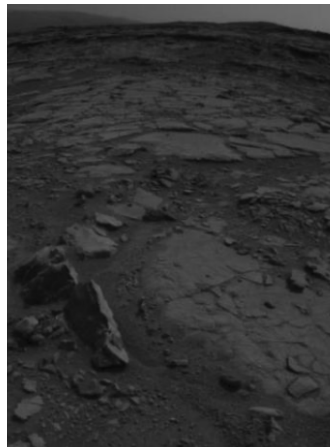


(d) Imagen resultante de la zona E de 4.1a.

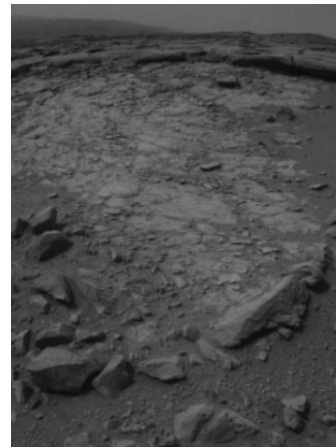
**Figura 4.1:** Ejemplo de las distintas zonas identificadas dentro de una imagen capturada por *Hazcam* con ID FRB\_421716121EDR\_F0060068FHAZ00323M1 [23].



(a) Nivel (0) de peligro.



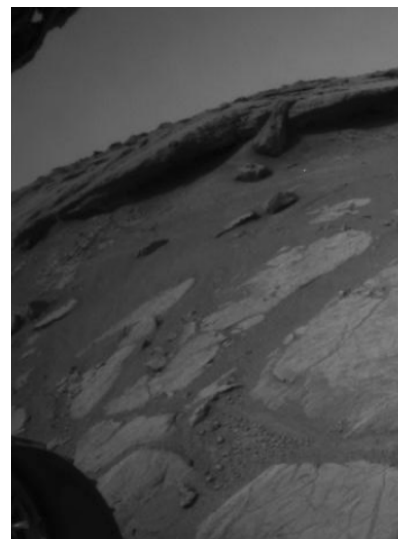
(b) Nivel (1) de peligro.



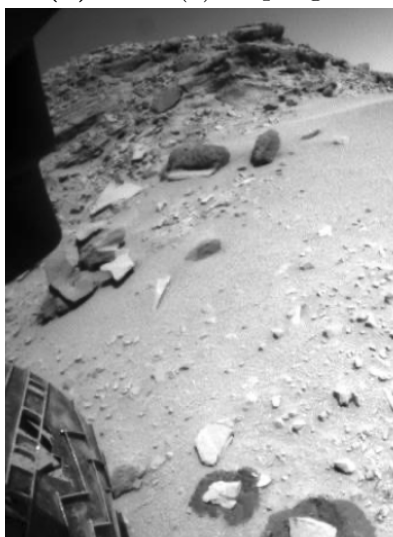
(c) Nivel (2) de peligro.



(d) Nivel (3) de peligro.



(e) Nivel (4) de peligro.



(f) Nivel (5) de peligro.



(g) Nivel (10) de peligro.

**Figura 4.2:** Ejemplos de imágenes para cada nivel de peligro [23].

## 4.2. Análisis con redes neuronales artificiales

### 4.2.1. Características basadas en los valores de los píxeles

La primera técnica que se empleará para la obtención de un modelo de predicción del nivel de peligro serán las redes neuronales artificiales (ANN) descritas en 3.1.

En la primera capa de estas redes neuronales el número de neuronas se corresponde con el número de datos de entrada a la red. En este caso, las entradas a la red serán los valores de brillo de cada uno de los píxeles de la imagen. De esta forma, el número de entradas será:  $340 \times 460 = 156400$  píxeles por foto. Dentro de las posibilidades técnicas con las que se cuenta es altamente inviable alojar y entrenar redes con tal número de neuronas en la capa de entrada y cantidades de las mismas acordes a dicho valor en las siguientes capas internas, por lo que se procede a bajar la resolución de las imágenes. Para ello, se establece un tamaño de *superpíxel* y se calcula la media del valor de los píxeles en el interior de dicha región. Este *tamaño del superpíxel o del grano* será ahora por tanto un nuevo parámetro de ajuste de nuestro modelo de cara a optimizar su rendimiento y coste computacional.

Se elige un tamaño de *superpíxel* que cumpla los siguientes requisitos:

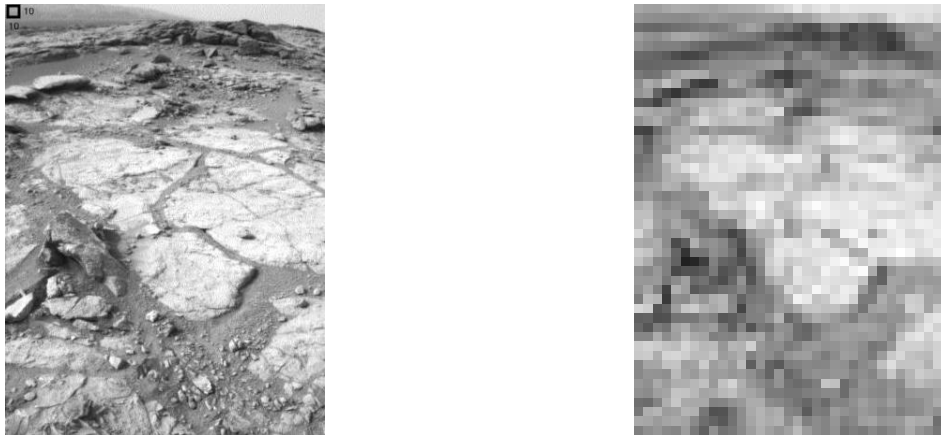
1. Que la pérdida de información que conlleve la reducción de la resolución sea la mínima posible.
2. Que el número de *superpíxeles* resultante sea una cantidad manejable por la red y equipos que se van a emplear.

Como primer valor del tamaño del *superpíxel* se elige unos valores de  $10 \times 10$ , de forma que en cada píxel hay  $10 \times 10 = 100$  píxeles, mientras que el valor total de *superpíxeles* en la imagen se reduce a  $\frac{156400}{100} = 1564$  *superpíxeles* por imagen, quedando éstas una resolución de  $34 \times 46$ . Un ejemplo de dicha reducción de resolución se muestra en la Figura 4.3.

A continuación, debido al carácter empírico de la creación de modelos de predicción empleando redes neuronales artificiales, se procede a realizar distintas pruebas de entrenamientos con distintas configuraciones o topologías de red. Durante éstos, se van variando parámetros como el número de capas ocultas y la cantidad de neuronas presentes en estas. Pese a no existir normas concretas sobre cómo debe ser esta topología en función de la cantidad de datos de entrada, una buena regla suele ser la presencia en la primera capa oculta de un número de neuronas mayor que  $\frac{2}{3}$  del tamaño de la capa de entrada pero menor que el doble del número de dichos datos. Sin embargo, cabe enfatizar en que dicha regla no es una norma, por lo que conviene ver también cuál es el rendimiento de la red probando otras configuraciones [24].

### Experimento 1A - *Superpíxeles* de $10 \times 10$

En primer lugar, se prueba la red introduciéndole a la misma los valores de los píxeles de las imágenes con un tamaño de grano de  $10 \times 10$ . Se realizan diferentes entrenamientos



(a) Imagen original recortada de resolución  $340 \times 460$ .

(b) Imagen una vez tomados los *superpíxeles* de  $10 \times 10$ .

**Figura 4.3:** Imagen antes y después de la reducción de resolución empleando *superpíxeles*. Se toma como ejemplo la zona D de la Figura 4.1 donde se muestra el tamaño elegido en primera instancia para el *superpíxel* de tamaño  $10 \times 10$  píxeles y el resultado.

empleando distintas topologías de red. Para el cálculo de las mismas se siguen las reglas enunciadas antes, de forma que para  $34 \times 46 = 1564$  entradas:

- Límite inferior:  $N_{min} = 1564 \cdot \frac{2}{3} \approx 1042 \sim 1000$
- Límite superior:  $N_{máx} = 1564 \cdot 2 = 3128 \sim 3000$

Sin embargo, como se ha mencionado antes se realizan entrenamientos con valores dentro y fuera de dicho rango.

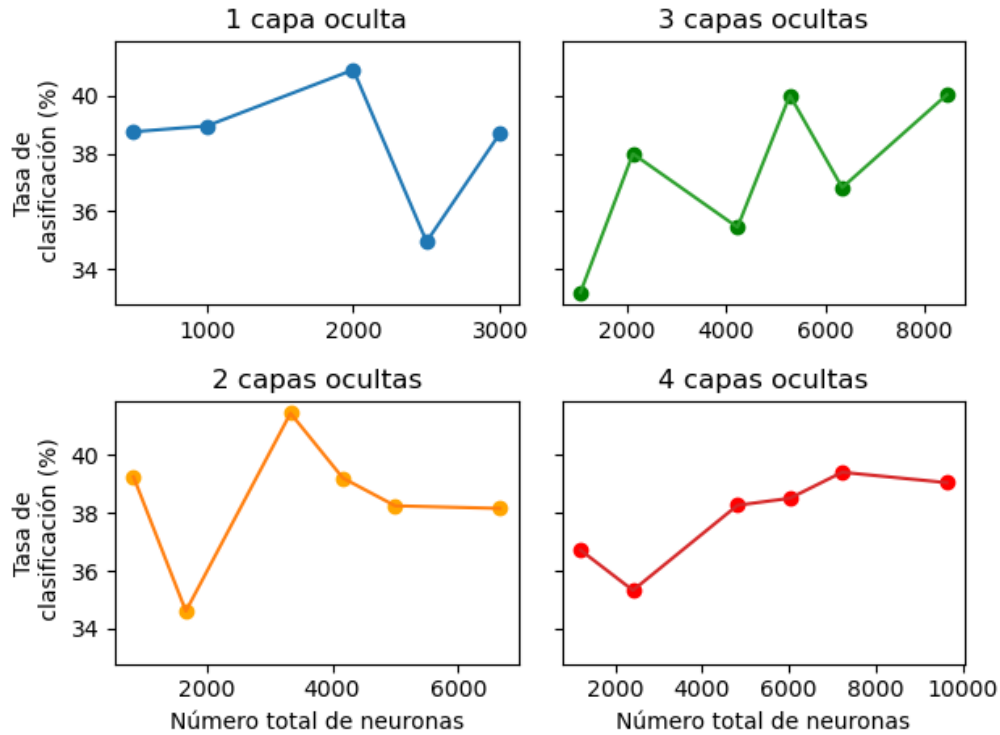
Como **entradas** a la red se emplearán valores de píxeles con imágenes de resolución de  $34 \times 46$  (1564 píxeles en total), fruto de la construcción de *superpíxeles* de tamaño  $10 \times 10$ .

Los **resultados** se recogen en las Tablas A.1 - A.4 del Anexo A. Además, se resumen éstos en la Figura 4.4. En la Figura 4.4a se representa la tasa de clasificación para cada topología en función del número total de neuronas empleado, utilizando un código de color para redes de una, dos, tres o cuatro capas ocultas. Del mismo modo, se representa en la Figura 4.4b el tiempo de entrenamiento para cada una de las configuraciones.

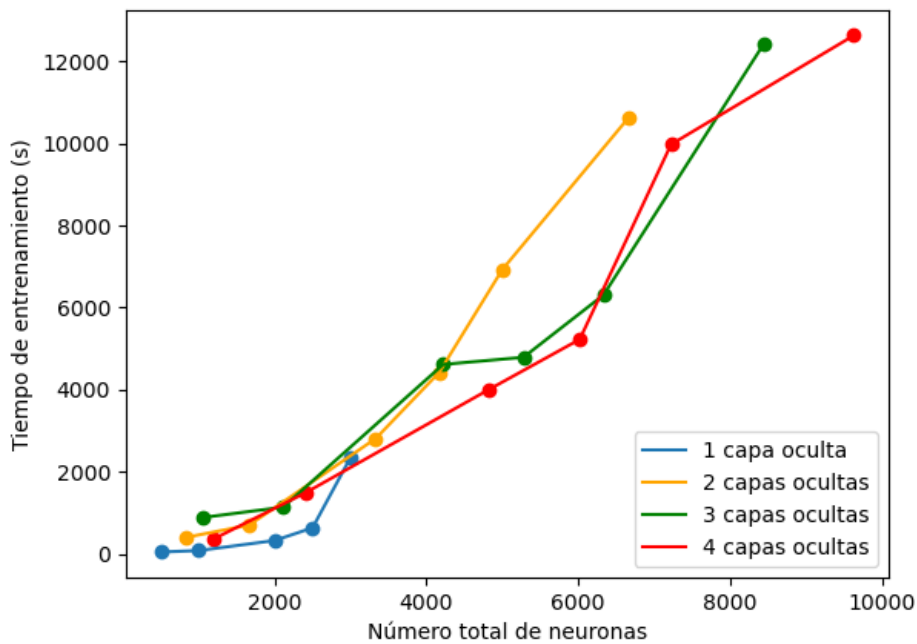
A la vista de los resultados obtenidos, se puede observar cómo la red tiene un rendimiento muy bajo, en el mejor de los casos del 41,45%. Este valor no sufre demasiados cambios al aumentar tanto las capas con las que cuenta la red, así como introduciendo un número mayor de neuronas en dichas capas internas. Además, analizando las matrices de confusión recogidas en el Anexo A se puede ver cómo la red confunde mucho las imágenes que representan peligros de los tres primeros niveles (0), (1) y (2).

Asimismo, se comprueba cómo añadir capas extra más allá de 2 capas ocultas no contribuye a una mejora de los resultados y únicamente conlleva un aumento sustancial del tiempo de entrenamiento. Finalmente, se comprueba cómo los mejores resultados se obtienen para topologías dentro de los márgenes marcados por las reglas de diseño enunciadas en 4.2.





(a) Tasas de clasificación.



(b) Tiempos de entrenamiento.

**Figura 4.4:** Resultados de las pruebas realizadas con las imágenes de *Hazcam* con diferentes topologías de red neuronal para un tamaño de grano de  $10 \times 10$ , tanto de tasas de clasificación (a) como de tiempos de entrenamiento (b). En 4.4a se emplea el mismo eje  $y$  en las cuatro figuras.

### Experimento 1B - Reducción tamaño del grano a *superpíxeles* de $5 \times 5$

Con el objetivo de mejorar los resultados anteriores, se realizan nuevos experimentos modificando nuevos parámetros, siendo el primero de éstos la reducción del tamaño de grano de la imagen. Estas nuevas pruebas se realizan ante la hipótesis de que con la reducción de resolución que conlleva el grano de  $10 \times 10$  empleado anteriormente se haya perdido información relevante. De esta forma, se reduce el tamaño del grano a  $5 \times 5$ , dando lugar a imágenes de resolución de  $68 \times 92$  y con  $68 \times 92 = 6256$  píxeles en total. El número de neuronas de las redes a emplear según las reglas enunciadas antes queda ahora, para dicho número de entradas:

- Límite inferior:  $N_{min} = 6256 \cdot \frac{2}{3} \approx 4170 \sim 4000$
- Límite superior:  $N_{máx} = 6256 \cdot 2 = 12512 \sim 12500$

Sin embargo, debido al alto coste de crear y entrenar redes con neuronas dentro de dichos rangos, se emplearán en su mayoría únicamente valores fuera de dicho rango.

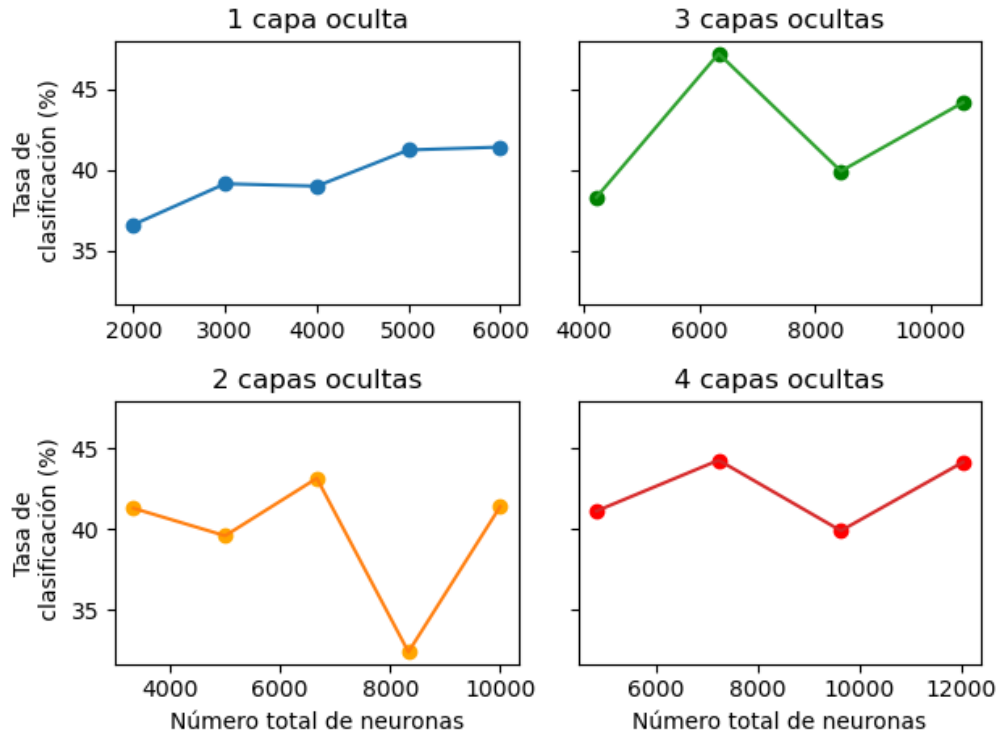
En este caso, como datos de **entrada** se emplean los valores de los píxeles con imágenes de resolución de  $68 \times 92$  (6256 píxeles en total), fruto de la construcción de *superpíxeles* de tamaño  $5 \times 5$ .

Los **resultados** están recogidos en las Tablas A.5 - A.8 del Anexo A. Además, se resumen éstos en la Figura 4.5, donde en la Figura 4.5a se representa la tasa de clasificación de la topología en función del número total de neuronal empleado, utilizando un código de color para redes de una, dos, tres o cuatro capas ocultas, mientras que en la Figura 4.5b se representan los tiempos de entrenamiento.

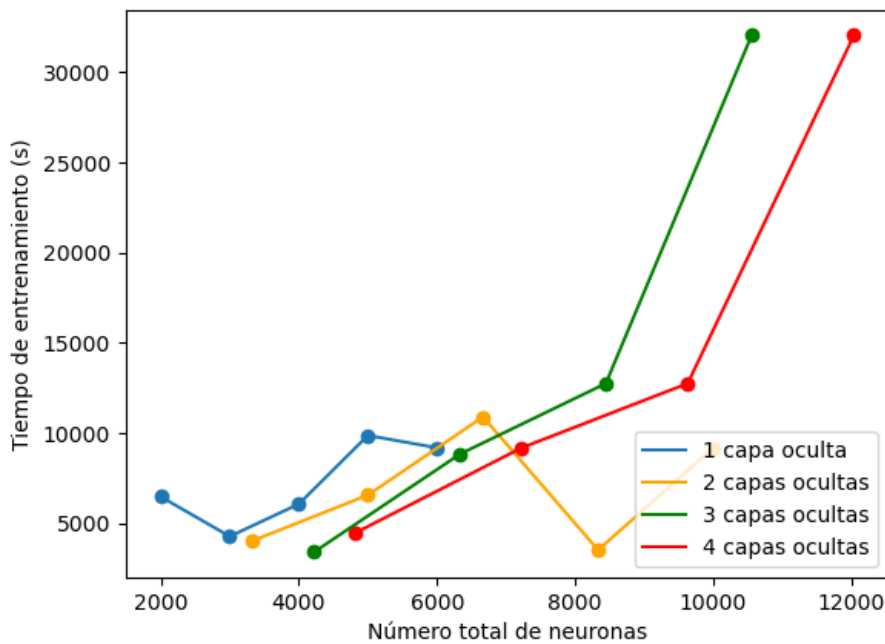
Se puede observar cómo la red sigue teniendo un rendimiento muy bajo, siendo la tasa de clasificación en el mejor de los casos del 40 %, mientras que los tiempos de entrenamiento han aumentado de forma significativa. Además dicho valor apenas cambia al aumentar tanto el número de capas internas como el número de neuronas en dichas capas. De esta forma, se concluye que el tamaño del grano de la imagen no era uno de los principales problemas, por lo que se descarta el cambio.

### Experimento 1C - Cambio en la etiqueta de la clase (0)

De nuevo, para mejorar el rendimiento de la red se realiza un nuevo experimento modificando el valor que se la atribuye a la clase de no peligro aparente, nivel (0) en 4.1, pasando éste a ser (-1). A la vista de los niveles de confusión entre los primeros niveles observados en los anteriores experimentos, este cambio se realiza con el objetivo de identificar la clase de menos peligro posible de forma mucho más clara y distinguible del resto. Se realizan las pruebas con la topología de red que daba los mejores resultados anteriormente, debido a que únicamente se desea verificar si este cambio implica algún tipo de mejora sustancial en el rendimiento. En ese caso, la red era de dos capas internas de 2000 y 1333 neuronas, respectivamente.



(a) Tasas de clasificación.



(b) Tiempos de entrenamiento.

**Figura 4.5:** Resultados de las pruebas realizadas con las imágenes de *Hazcam* con diferentes topologías de red neuronal para un tamaño de grano de  $5 \times 5$ , tanto de tasas de clasificación (a) como de tiempos de entrenamiento (b). En 4.5a se utiliza el mismo eje  $y$  en las cuatro figuras.

En esta ocasión se emplearán como datos de **entrada** los valores de los píxeles con imágenes de resolución de  $34 \times 46$  con *superpíxeles* de tamaño  $10 \times 10$ . Se modifica el valor de la clase de no peligro, pasando éste de (0) a (-1).

Los **resultados** se recogen en la Tabla 4.1. Se presentan la matriz de confusión ( $M_C$ ), la tasa de clasificación ( $T_{clas}$ ) y el tiempo de entrenamiento ( $t_{entr}$ ).

**Tabla 4.1:** Resultados para el experimento 1C, donde se ha etiquetado el nivel (0) como (-1). Se emplea una red con una configuración de dos capas internas con 2000 y 1333 neuronas, respectivamente.

$M_C$	$T_{clas}$ (%)	$t_{entr}$ (hh:mm:ss)
$\begin{pmatrix} 379 & 251 & 100 & 34 & 14 & 8 & 4 \\ 395 & 394 & 228 & 76 & 28 & 14 & 4 \\ 113 & 135 & 119 & 60 & 14 & 7 & 2 \\ 14 & 23 & 28 & 25 & 10 & 5 & 1 \\ 7 & 9 & 9 & 10 & 11 & 9 & 2 \\ 1 & 1 & 1 & 4 & 4 & 6 & 1 \\ 7 & 10 & 14 & 16 & 23 & 75 & 296 \end{pmatrix}$	41,03	00:56:16

Se puede observar como la red sigue teniendo un rendimiento muy bajo, ya que la tasa de clasificación es de nuevo cercana al 40%. Por tanto, se puede concluir que la propuesta de solución de reetiquetado de la clase (0) no conlleva una mejora de los resultados. Por ello, se descarta el cambio y se evalúan otras nuevas modificaciones.

### Experimento 1D - Agrupación de las clases (5) y (10)

A continuación se prueba a realizar un nuevo experimento agrupando las clases de obstáculo de gran tamaño y presencia de arena en la imagen, niveles (5) y (10) de 4.1, respectivamente. De esta manera, el nivel de máximo peligro, tanto por presencia de arena como de grandes rocas, se representa con una sola etiqueta. Al igual que en las pruebas anteriores, se realizan los experimentos con topologías de red que daban los mejores resultados, debido a que únicamente se desea buscar si este cambio implica algún tipo de mejora sustancial en el rendimiento.

Como **entrada** a la red se utilizan los valores de los píxeles con imágenes de resolución de  $34 \times 46$  con *superpíxeles* de tamaño  $10 \times 10$ . Se modifica el valor de la clase de presencia de arena, pasando éste de (10) a (5). Los **resultados** están recogidos en la Tabla 4.2. Se presentan la matriz de confusión ( $M_C$ ), la tasa de clasificación ( $T_{clas}$ ) y el tiempo de entrenamiento ( $t_{entr}$ ).

De nuevo, se ve cómo la red sigue teniendo un rendimiento muy bajo, siendo la tasa de clasificación de nuevo cercana al 40%. Por tanto, se vuelve a concluir que la propuesta de solución de reetiquetado de la clase (10) no conlleva una mejora de los resultados. Se descarta de nuevo el cambio realizado y se evalúan otras nuevas modificaciones.

**Tabla 4.2:** Resultados para el experimento 1D, donde se ha etiquetado el nivel (10) como (5). Se emplea una red con una configuración de dos capas internas con 2000 y 1333 neuronas, respectivamente.

$M_C$	$T_{clas}$ (%)	$t_{entr}$ (hh:mm:ss)
$\begin{pmatrix} 140 & 527 & 101 & 17 & 4 & 0 \\ 112 & 772 & 225 & 27 & 4 & 0 \\ 24 & 200 & 193 & 29 & 3 & 0 \\ 2 & 26 & 39 & 34 & 5 & 1 \\ 1 & 8 & 13 & 16 & 16 & 2 \\ 2 & 25 & 60 & 106 & 186 & 79 \end{pmatrix}$	41,14	00:46:42

### Experimento 1E - Agrupación de clases

A raíz de los malos resultados obtenidos y las mínimas variaciones que provocan los cambios introducidos en los experimentos anteriores 1C y 1D, y a la vista de las matrices de confusión de cada uno de los entrenamientos, sobre todo entre los primeros niveles, se concluye que el etiquetado que se ha realizado de las imágenes no es correcto. De hecho, la conclusión es que la red no es capaz de distinguir entre los niveles de peligro expuestos en 4.1 debido a las finas y no bien descritas fronteras entre ellos. Idealmente, dicha clasificación de niveles podría funcionar estableciendo unos criterios claros y *objetivos* para cada uno, así como la inclusión de más parámetros que dieran cuenta del contenido de la imagen.

Por tanto, con el objetivo de mejorar el rendimiento de la red y viendo que ésta no es capaz de distinguir entre los 7 niveles expuestos, se prueba a agrupar estos en un número menor de niveles. Se realizan distintos entrenamientos con configuraciones de red cercanas a la topología que antes mejores resultados daba.

En este caso como valores de **entrada** se emplean los valores de los píxeles con imágenes de resolución de  $34 \times 46$  con *superpíxeles* de tamaño  $10 \times 10$ . Se agrupan las clases de distintas formas, expuestas éstas en la Tabla 4.3. Los **resultados** se recogen en la Tabla 4.3. Se presentan las distintas agrupaciones de clases empleadas, la matriz de confusión ( $M_C$ ), la tasa de clasificación ( $T_{clas}$ ) y el tiempo de entrenamiento ( $t_{entro}$ ) para cada agrupación.

**Tabla 4.3:** Resultados experimento 1E con 2 capas ocultas de 2000 y 1333 neuronas, respectivamente.

Grupos de clases	$M_C$	$T_{clas}$ (%)	$t_{entr}$ (hh:mm:ss)
(0) (1,2) (3,4) (5,10)	$\begin{pmatrix} 226 & 538 & 24 & 1 \\ 127 & 1363 & 100 & 1 \\ 5 & 74 & 83 & 2 \\ 4 & 75 & 239 & 140 \end{pmatrix}$	60,37	00:48:58
(0) (1,2,3) (4,5) (10)	$\begin{pmatrix} 379 & 397 & 12 & 1 \\ 492 & 1137 & 66 & 1 \\ 18 & 25 & 30 & 1 \\ 91 & 57 & 193 & 100 \end{pmatrix}$	54,89	00:32:34

**Tabla 4.3:** Resultados experimento 1E con 2 capas ocultas de 2000 y 1333 neuronas, respectivamente.

Grupos de clases	$M_C$	$T_{clas}$ (%)	$t_{entr}$ (hh:mm:ss)
(0) (1,2,3) (4,5,10)	$\begin{pmatrix} 309 & 476 & 4 \\ 111 & 1565 & 20 \\ 4 & 272 & 239 \end{pmatrix}$	70,44	00:41:53
(0,1,2) (3,4,5,10)	$\begin{pmatrix} 2350 & 29 \\ 269 & 352 \end{pmatrix}$	90,05	00:53:14

A la vista de los resultados, se observa cómo la red sigue sin ser capaz de ofrecer un rendimiento aceptable para los grupos de clases de 3 niveles o más. Por ello, se concluye que la red únicamente es capaz de ofrecer un buen rendimiento con la clasificación binaria. De esta manera, con el objetivo de seguir manteniendo en cierta manera el significado físico de las clases, se elige esta clasificación binaria como: (0,1,2) *no peligro* y (3,4,5,10) *peligro*.

Aunque se ha logrado con esta última modificación aumentar sustancialmente el rendimiento de la red y llegar a una tasa de clasificación del 90,05 %, se continuarán realizando distintas pruebas mediante ajustes más precisos de la topología e implementado nuevas técnicas con el objetivo de ver qué rendimiento se puede alcanzar.

#### 4.2.2. Características basadas en el cálculo de histogramas de gradientes orientados (HOG)

A continuación se decide emplear una nueva técnica para la descripción de las imágenes, en vez de simplemente los brillos de los píxeles. Por ello, se emplea la técnica de *histogramas de gradientes orientados (HOG)* descrita en 3.3. En este proceso se obtienen los histogramas de gradientes de zonas delimitadas de la imagen, siendo el primer paso seleccionar el tamaño de las celdas sobre las que calcular dicho gradiente.

En primer lugar se debe tener en cuenta que del tamaño de la celda dependerá el tamaño de la capa de entrada de la red, de manera que la topología de la misma se deberá de ajustar al tamaño de ésta. Además, cuanto mayor es el tamaño de las celdas seleccionadas, más información del detalle de la imagen se pierde. De esta manera, el número de neuronas a emplear según las reglas empleadas hasta ahora, sobre imágenes a resolución completa de  $340 \times 460$ , queda entre los siguientes límites:

- Celdas de  $80 \times 80$  que dan lugar a entradas de 432 elementos.
  - Límite inferior:  $N_{min} = 432 \cdot \frac{2}{3} = 288$
  - Límite superior:  $N_{max} = 432 \cdot 2 = 864$
- Celdas de  $40 \times 40$  que dan lugar a entradas de 2520 elementos.
  - Límite inferior:  $N_{min} = 2520 \cdot \frac{2}{3} = 1680$
  - Límite superior:  $N_{max} = 2520 \cdot 2 = 5040$

Además, se prueba a utilizar como base para calcular los HOG las imágenes con *superpíxeles* de  $10 \times 10$  y  $5 \times 5$ . En el caso de las primeras se tienen imágenes de resolución  $34 \times 46$ , de forma que el número de neuronas queda:

- Celdas de  $8 \times 8$  que dan lugar a entradas de 432 elementos.

- Límite inferior:  $N_{min} = 432 \cdot \frac{2}{3} = 288$
- Límite superior:  $N_{máx} = 432 \cdot 2 = 864$

- Celdas de  $4 \times 4$  que dan lugar a entradas de 2520 elementos.

- Límite inferior:  $N_{min} = 2520 \cdot \frac{2}{3} = 1680$
- Límite superior:  $N_{máx} = 2520 \cdot 2 = 5040$

En el caso de la celdas de  $5 \times 5$ , para imágenes de resolución  $68 \times 92$  se tiene:

- Celdas de  $16 \times 16$  que dan lugar a entradas de 391 elementos.

- Límite inferior:  $N_{min} = 432 \cdot \frac{2}{3} = 288$
- Límite superior:  $N_{máx} = 432 \cdot 2 = 864$

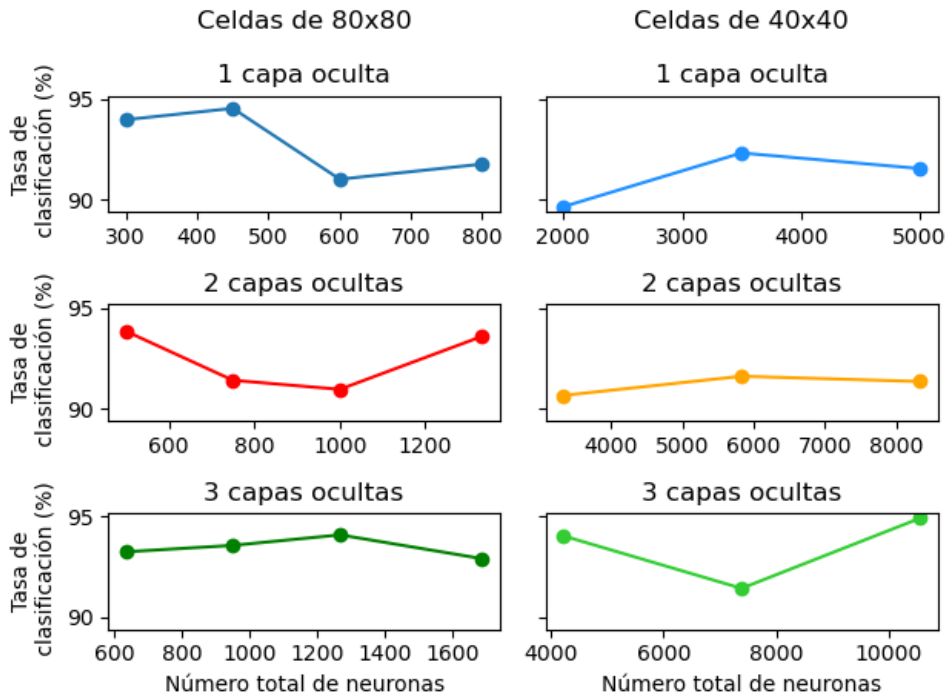
- Celdas de  $8 \times 8$  que dan lugar a entradas de 1564 elementos.

- Límite inferior:  $N_{min} = 2520 \cdot \frac{2}{3} = 1680$
- Límite superior:  $N_{máx} = 2520 \cdot 2 = 5040$

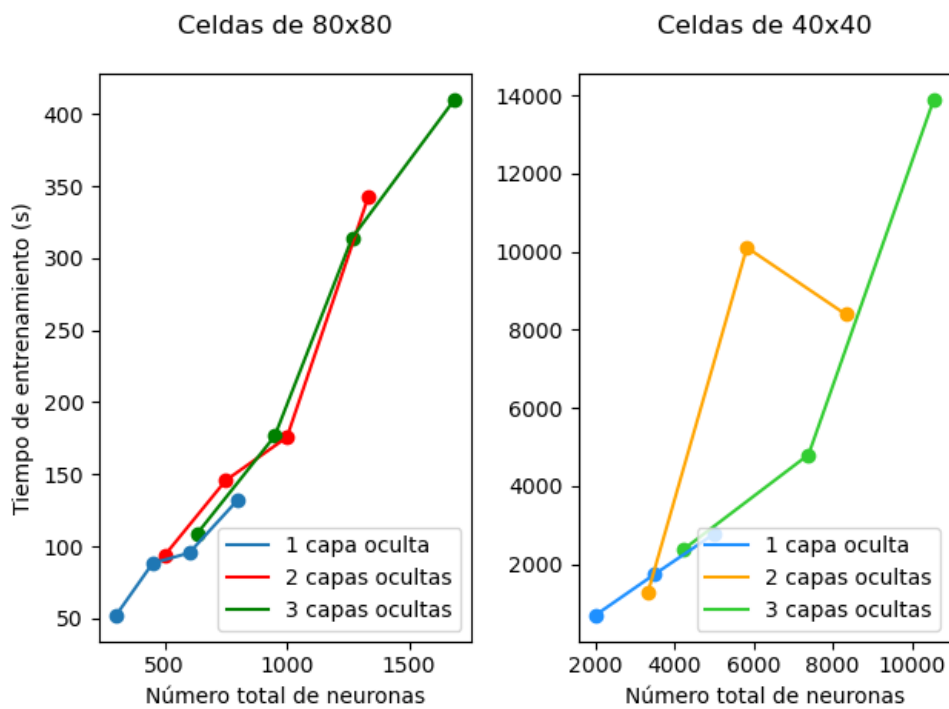
## Experimento 1F - Variación del tamaño de la celda HOG y de la imagen base

En este experimento se utilizan como **entrada** a la red los valores de HOG obtenidos a partir de imágenes originales de resolución de  $340 \times 460$ , imágenes con *superpíxeles* de  $10 \times 10$  y de  $5 \times 5$ . Los **resultados** están recogidos en las Tablas A.9 - A.26 del Anexo A. Además, se resumen éstos en las Figuras 4.6, 4.7 y 4.8. En los tres casos se muestran las tasas de clasificación y los tiempos de entrenamiento.

A la vista de los resultados se puede concluir que los mejores resultados se obtienen empleando HOG calculado sobre la imagen a resolución completa, empleando una celda de  $40 \times 40$  y una configuración de red con tres capas internas con 5000, 3333 y 2222 neuronas. En dicho caso la tasa de clasificación se sitúa en 94,91 %. No obstante, esta configuración acarrea un importante coste computacional, ya que el tiempo de entrenamiento medio es de 03:51:50 en el equipo empleado. Es por esto por lo que puede ser conveniente explorar otras configuraciones, como por ejemplo es el caso de HOG calculado sobre las imágenes a resolución completa, con celdas de  $80 \times 80$  y una topología de una sola capa oculta con 450 neuronas. En este caso la tasa de clasificación es del 94,54 % y el tiempo medio de entrenamiento de 00:01:28, siendo este último muy inferior al anterior.



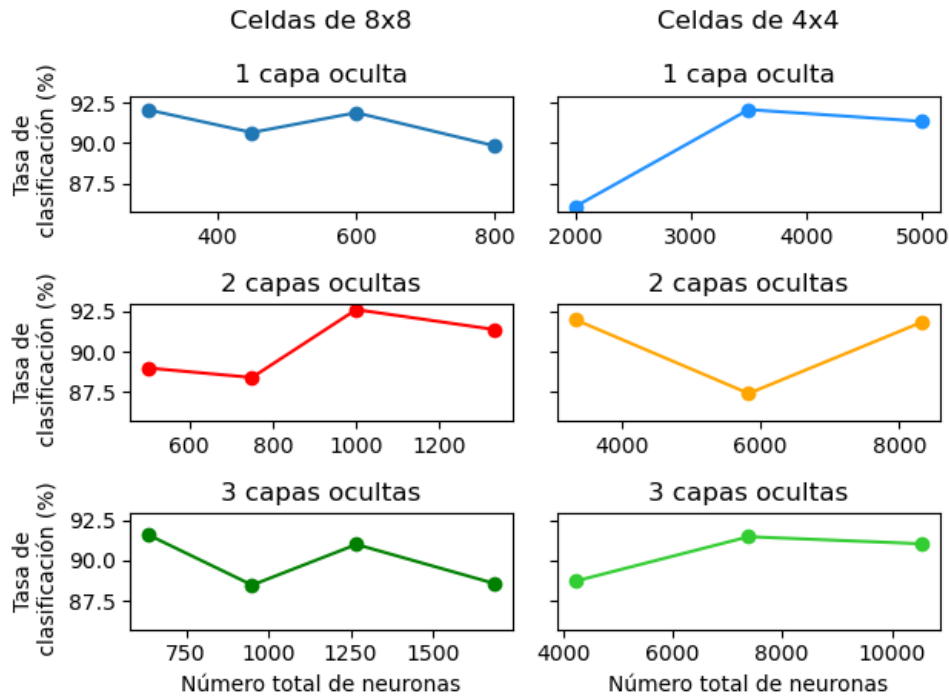
(a) Tasas de clasificación.



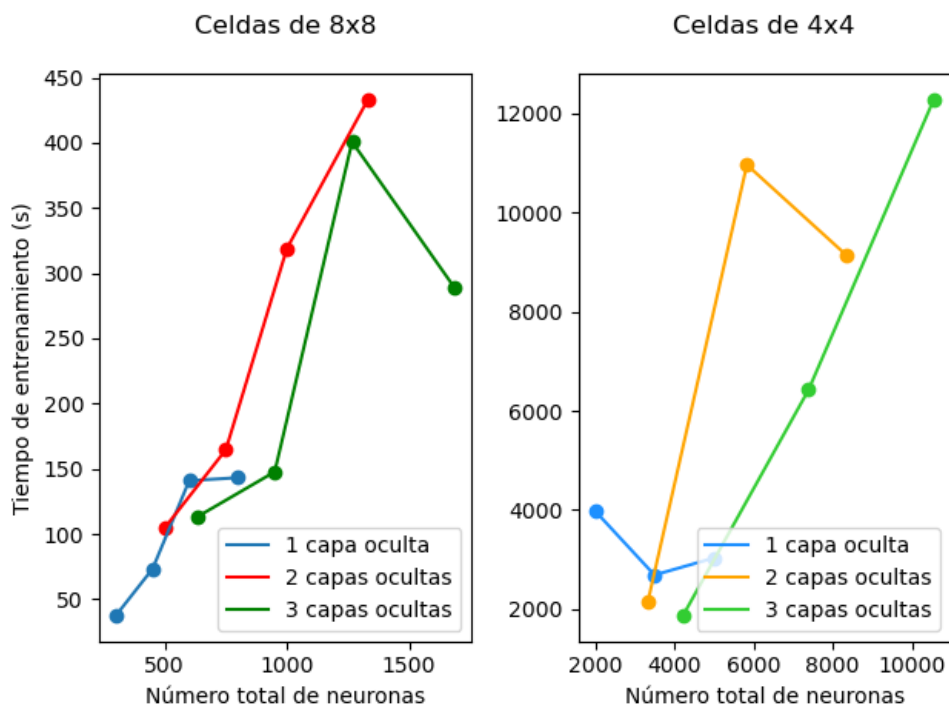
(b) Tiempos de entrenamiento.

**Figura 4.6:** Resultados de las pruebas realizadas con las imágenes de *Hazcam* con diferentes topologías de red neuronal calculando los HOG con las capturas a resolución completa. Se analizan tamaños de celda para el cálculo del HOG de  $80 \times 80$  y  $40 \times 40$ . Se representan tanto las tasas de clasificación (a) como los tiempos de entrenamiento (b).



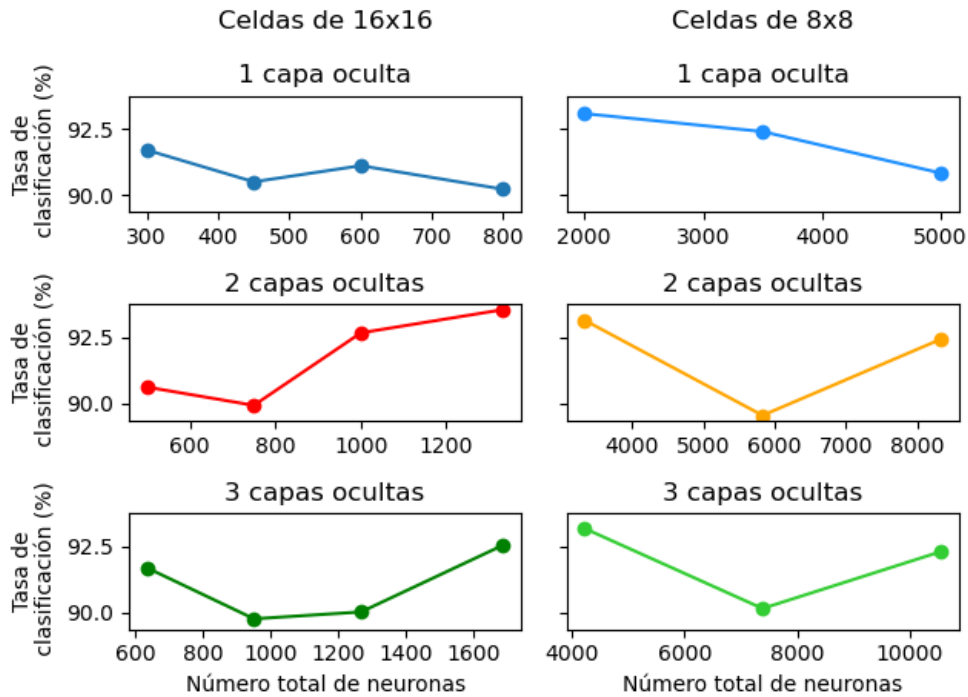


(a) Tasas de clasificación.

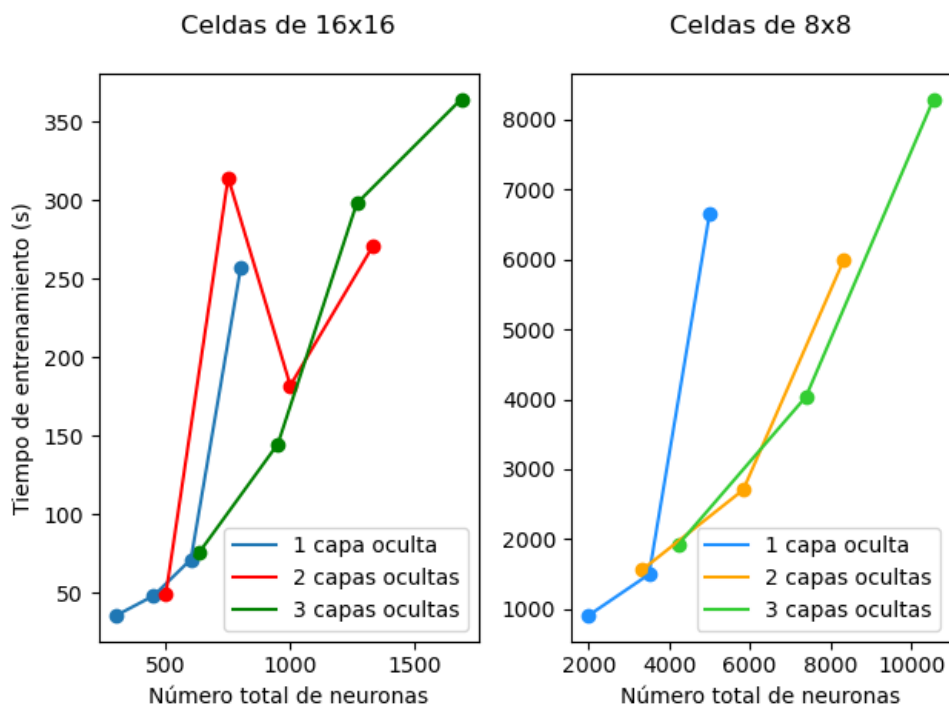


(b) Tiempos de entrenamiento.

**Figura 4.7:** Resultados de las pruebas realizadas con las imágenes de *Hazcam* con diferentes topologías de red neuronal calculando los HOG con las capturas a con *superpíxeles* de  $10 \times 10$ . Se analizan tamaños de celda para el cálculo del HOG de  $8 \times 8$  y  $4 \times 4$ . Se representan tanto las tasas de clasificación (a) como los tiempos de entrenamiento (b).



(a) Tasas de clasificación.



(b) Tiempos de entrenamiento.

**Figura 4.8:** Resultados de las pruebas realizadas con las imágenes de *Hazcam* con diferentes topologías de red neuronal calculando los HOG con las capturas a con *superpíxeles* de  $5 \times 5$ . Se analizan tamaños de celda para el cálculo del HOG de  $16 \times 16$  y  $8 \times 8$ . Se representan tanto las tasas de clasificación (a) como los tiempos de entrenamiento (b).

## 5. Detección de púlsares

### 5.1. Descripción de la base de datos

Como se mencionó en la introducción, los púlsares son un tipo raro de estrellas de neutrones que producen una emisión de radio detectable aquí en la Tierra. En la actualidad se utilizan herramientas de *machine learning* para etiquetar automáticamente los púlsares candidatos como problemas de clasificación binaria. Actualmente no se dispone de etiquetas multiclase, dados los costes asociados a la recopilación y análisis de dichos datos.

La base de datos con la que se va a trabajar [9, 25] contiene datos correspondientes a 17898 señales, de las cuales 16259 son señales que no pertenecen a púlsares y posiblemente han sido causadas por interferencias RFI y ruido, mientras que 1639 ejemplos son de púlsares reales. Cabe destacar que todos estos ejemplos han sido revisados por especialistas del campo. Los datos se presentan en formato CSV, de forma que cada fila enumera primero las variables, siendo la etiqueta de clase la última entrada. Las etiquetas de clase utilizadas son 0 para señales que no provienen de un púlsar (negativas) y 1 para las que sí (positivas).

Asimismo, los datos aquí recogidos no contienen información de posición ni otros detalles astronómicos. Se trata simplemente de datos de características extraídos de archivos candidatos utilizando la herramienta *PulsarFeatureLab* [25]. Cada una de dichas señales candidatas es descrita por 8 parámetros continuos y otro que indica la clase a la que pertenece. Estos parámetros son:

1. Media del perfil integrado.
2. Desviación estándar del perfil integrado.
3. Exceso de curtosis del perfil integrado.
4. Asimetría del perfil integrado.
5. Media de la curva DM-SNR.
6. Desviación estándar de la curva DM-SNR.
7. Exceso de curtosis de la curva DM-SNR.
8. Asimetría de la curva DM.

Los cuatro primeros son simples parámetros estadísticos obtenidos a partir del perfil de pulso integrado de la señal candidata, mientras que las cuatro variables restantes se obtienen de forma similar a partir de la curva DM-SNR. Esta curva relaciona la medida de dispersión (DM), la cual refleja el desplazamiento de los tiempos de llegada de las señales transmitidas según su frecuencia, con la relación señal ruido S/N de la señal (SNR).

## 5.2. Detección con redes neuronales

En primer lugar, se emplearán redes neuronales artificiales (ANN) descritas en 3.1 como técnica para la obtención de un modelo de predicción. En ellas los datos de la primera capa, la capa de entrada, serán los 8 parámetros que caracterizan cada señal descritos antes. Por tanto, a partir de la reglas establecidas en 4.1 para el diseño de la topología de las redes, los límites superior e inferior quedan para 8 entradas:

- Límite inferior:  $N_{min} = 8 \cdot \frac{2}{3} \approx 5$
- Límite superior:  $N_{máx} = 8 \cdot 2 = 16$

Al igual que con la anterior base de datos, se usarán topologías tanto dentro como fuera de dichos límites, así como los mismos métodos para el análisis del rendimiento y los valores de la configuración de la herramienta de entrenamiento.

### Experimento 2A - Red neuronal artificial

Como valores de **entrada** a la red se emplean los valores de los 8 parámetros que caracterizan la señal candidata.

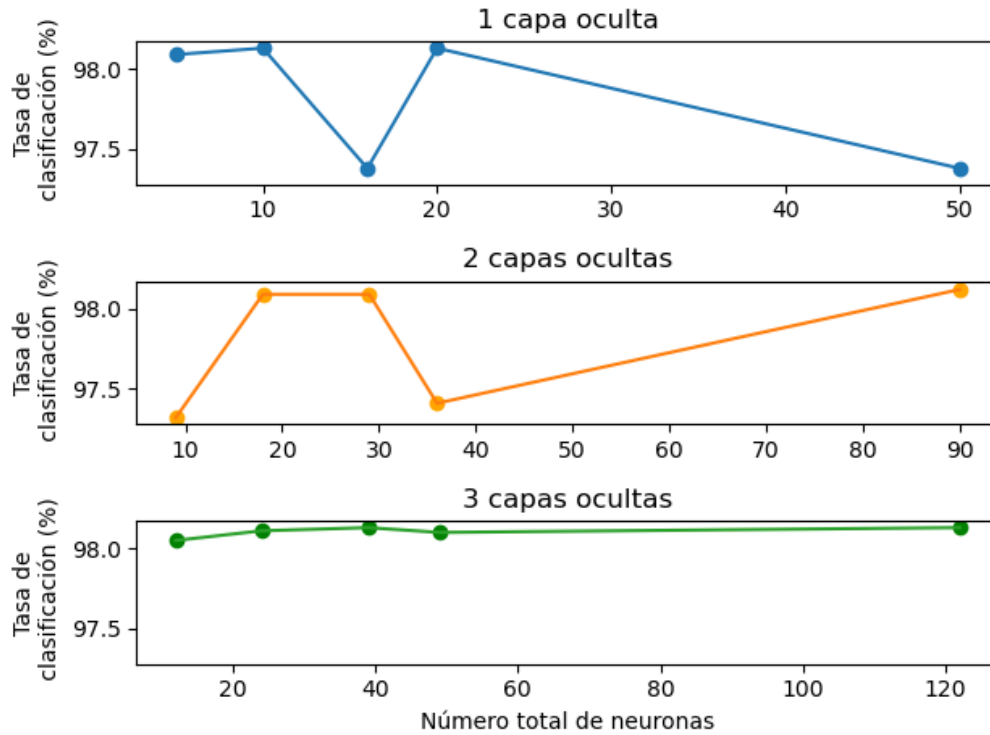
Los **resultados** se recogen en las Tablas A.27 - A.29 del Anexo A. Además, éstos se resumen en la Figura 5.1. En la Figura 5.1a se representa la tasa de clasificación de la topología en función del número total de neuronal empleado, empleando un código de color para redes de una, dos o tres capas ocultas. Del mismo modo, se representa en la Figura 5.1b el tiempo de entrenamiento para cada una de ellas.

A la vista de los resultados de puede ver cómo para todas las topologías de red las tasas de clasificación superan el 97%, llegando en el mejor de los casos del 98,13%, para topologías de tres capas (16,13,10) y (50,40,32). De estas dos se elige la primera como la más óptima para este problema debido a su menor coste computacional, 00:00:39 frente a 00:01:36.

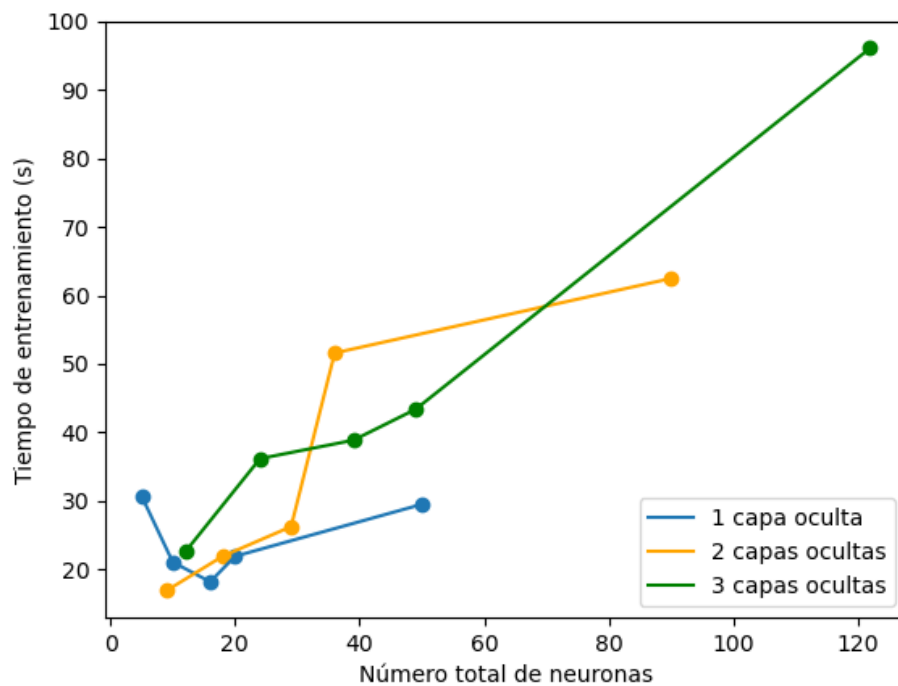
## 5.3. Máquinas de vectores de soporte (SVM)

La segunda forma de obtener un modelo de predicción para la detección de pulsares será empleando las *máquinas de vectores de soporte (SVM)* explicadas en 3.2. Para ello se emplearán las funciones de *MatLab* presentes dentro de la biblioteca *Statistics and Machine Learning Toolbox*. Del mismo modo, se vuelven a emplear las herramientas de medición del rendimiento del modelo utilizadas hasta el momento. Destacar que en este caso el tiempo de entrenamiento ( $t_{entr}$ ) se refiere al tiempo en el que se ha construido el modelo.

En cuanto al proceso de entrenamiento en sí mismo, se utilizarán  $\frac{2}{3}$  del total de 17898 muestras, es decir, 11932 muestras para entrenar el modelo; mientras que el  $\frac{1}{3}$  restante,



(a) Tasas de clasificación.



(b) Tiempos de entrenamiento.

**Figura 5.1:** Resultados de las pruebas realizadas con las señales candidatas a ser púlsares con diferentes topologías de red neuronal, tanto de tasas de clasificación (a) como de tiempos de entrenamiento (b).

5966 muestras, se empleará para probar el modelo obtenido. Dichas muestras se elegirán de forma aleatoria de entre el total.

Asimismo, se procede al entrenamiento del clasificador empleado la función *fitcsvm*, cuyos parámetros se establecen como sigue: *KernelFunction* en *rbf* (*radial basis function*), ya que es la función recomendada en este tipo de clasificadores, y *OptimizeHyperparameters* en *auto* para la realización del mejor ajuste de los parámetros del modelo por parte de la herramienta de entrenamiento.

En este experimento se emplean de nuevo como valores de **entrada** los valores de los 8 parámetros que caracterizan la señal candidata. Los **resultados** se recogen en la Tabla 6.1. Se representa la tasa de clasificación del modelo de predicción junto a la matriz de confusión de la que se ha extraído y el tiempo de entrenamiento.

**Tabla 5.1:** Resultados para el experimento 2C, donde se ha empleado un modelo de predicción creado con algoritmos SVM.

$\mathbf{M}_C$	$\mathbf{T}_{\text{clas}}$	$\mathbf{t}_{\text{entr}}$
$\begin{pmatrix} 5404 & 26 \\ 93 & 443 \end{pmatrix}$	98,01 %	00:20:52

A la vista de los resultados se puede concluir que para esta base de datos el modelo de predicción creado a partir de un clasificador SVM ofrece unos resultados muy buenos, con una tasa de acierto del 98,01 %. Se puede comprobar que ésta es algo inferior a la dada por las redes neuronales en el mejor de los casos. Por tanto, se concluye que el mejor modelo de predicción para la detección de pulsares a partir de datos de señales de RF detectadas es el que se obtiene empleando redes neuronales, debido a que pese a que se obtienen tasas de clasificación muy similares, en el caso de las ANN el tiempo de entrenamiento es mucho más bajo que para la obtención del clasificador SVM, 00:00:39 frente a 00:20:52.

## 6. Clasificación de objetos estelares

### 6.1. Descripción de la base de datos

En la línea de lo mencionado en capítulos anteriores, las técnicas de *machine learning* actuales también permiten realizar una clasificación de los diferentes objetos estelares que vemos en el cosmos en función de sus características espectrales. De este modo, atendiendo este criterio se puede distinguir entre elementos como estrellas, galaxias y cuásares, una de las clasificaciones más fundamentales de la astronomía.

La base de datos con la que se va a trabajar [26] consta de 100000 observaciones del espacio tomadas por el *Sloan Digital Sky Survey (SDSS)* [27]. Cada una de estas observaciones se describe mediante 17 parámetros de características y la clase a la que pertenece el objeto observado, es decir, si se trata de una estrella, galaxia o cuásar. Los parámetros empleados para la descripción de los objetos son:

1. **Obj\_ID:** Identificador del objeto y valor único que lo identifica en el catálogo de imágenes utilizado.
2. **Alpha:** Ángulo de ascensión recta en la época J2000.
3. **Delta:** Ángulo de declinación en la época J2000.
4. **u:** Filtro ultravioleta en el sistema fotométrico.
5. **g:** Filtro verde en el sistema fotométrico.
6. **r:** Filtro rojo en el sistema fotométrico.
7. **i:** Filtro del infrarrojo cercano en el sistema fotométrico.
8. **z:** Filtro infrarrojo en el sistema fotométrico.
9. **Run\_ID:** Número de ejecución utilizado para identificar la exploración.
10. **Rerun\_ID:** Número de repetición de la ejecución para especificar cómo se ha procesado la imagen.
11. **Cam\_col:** Columna de la cámara para identificar la línea de escaneo dentro de la ejecución.
12. **Field\_ID:** Número de campo empleado para identificar cada campo.
13. **Spec\_obj\_ID:** ID único utilizado para los objetos espectroscópicos ópticos. Esto significa que 2 observaciones diferentes con el mismo spec\_obj\_ID deben compartir la clase de salida.
14. **Redshift:** Valor de *redshift* (desplazamiento al rojo) basado en el aumento de la longitud de onda.
15. **Plate:** Identifica cada placa en SDSS.

16. **MJD**: Fecha juliana modificada, utilizada para indicar cuándo se tomó un dato determinado del SDSS.
17. **Fiber\_ID**: Identifica la fibra que apuntó la luz al plano focal en cada observación.

Al igual que en el caso de las anteriores bases de datos, se emplearán tanto redes neuronales artificiales como algoritmos SVM. El interés del análisis de esta nueva base de datos reside en que, a diferencia del caso de detección de púlsares, esta vez se trata de una clasificación multiclase. Para ello se realizan pruebas y ajustes de los modelos siguiendo los mismos pasos que en los casos anteriores.

## 6.2. Análisis con redes neuronales

El primer método que se emplea para la obtención del modelo de predicción es el de las redes neuronales artificiales. En este caso, la primera capa, la capa de entrada, contará con 17 neuronas, cada una de ellas correspondiente a uno de los parámetros descritos antes. Del mismo modo, se utilizarán las mismas herramientas para el análisis del rendimiento de los distintos algoritmos y modelos, así como las reglas establecidas en 4.1 para el diseño de la topología de las redes. De esta manera, los límites superior e inferior quedan para 17 entradas:

- Límite inferior:  $N_{min} = 17 \cdot \frac{2}{3} \approx 12$
- Límite superior:  $N_{máx} = 17 \cdot 2 = 34$

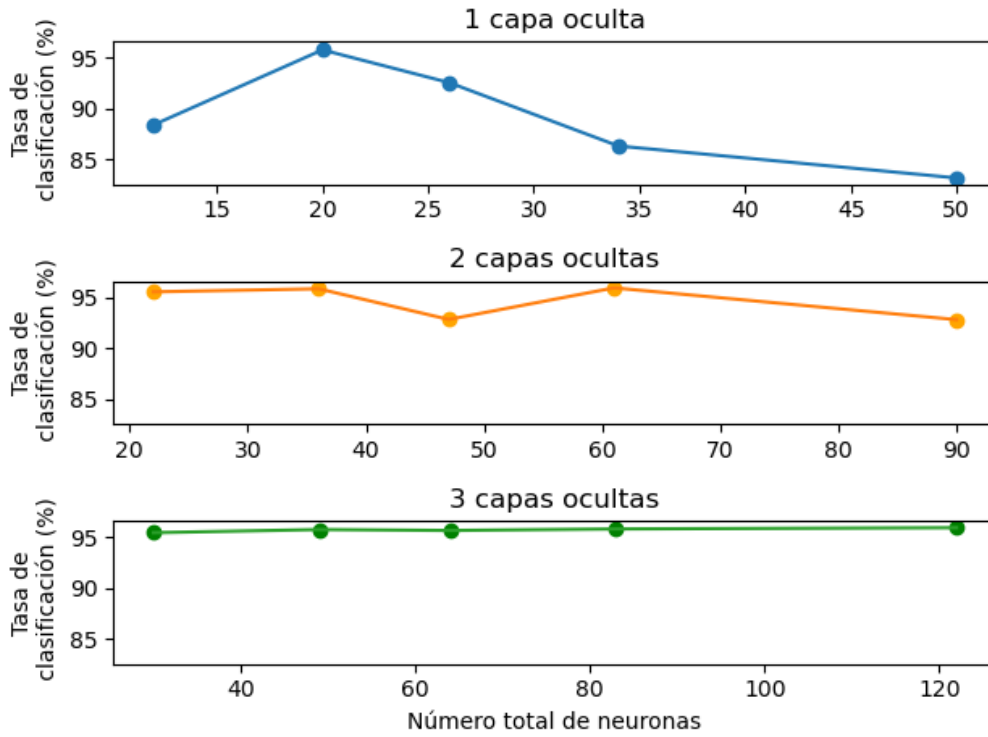
### Experimento 3A - Red neuronal artificial

Como datos de **entrada** a la red se utilizan los valores de los 17 parámetros que caracterizan cada una de las observaciones.

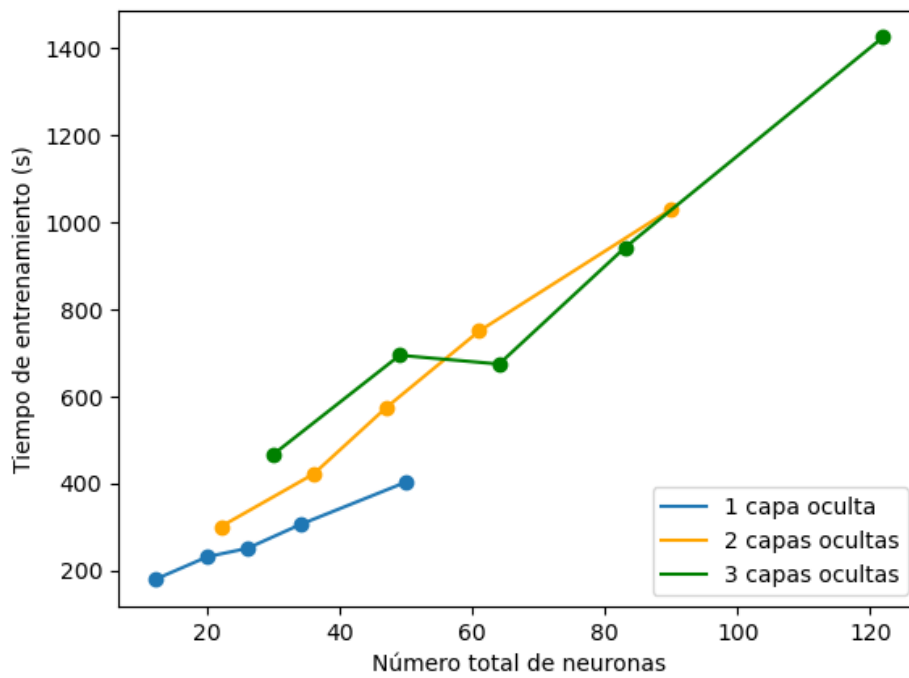
Los **resultados** se recogen en las Tablas A.30 - A.32 del Anexo A. Además, éstos se resumen en la Figura 6.1, donde en la Figura 6.1a se representa la tasa de clasificación de la topología en función del número total de neuronas empleado, utilizando un código de color para redes de una, dos o tres capas ocultas. Del mismo modo, se representa en la Figura 6.1b el tiempo de entrenamiento para cada configuración.

A la vista de los resultados se observa que la mejor tasa de clasificación se obtiene para una configuración de dos capas con 34 y 27 neuronas, con la que dicha tasa es del 95,97%. Con 3 capas ocultas se obtienen también buenos resultados pero estos conllevan un mayor tiempo de entrenamiento.





(a) Tasas de clasificación.



(b) Tiempos de entrenamiento.

**Figura 6.1:** Resultados de las pruebas realizadas con las señales de los objetos estelares con diferentes topologías de red neuronal, tanto de tasas de clasificación (a) como de tiempo de entrenamiento (b).

### 6.3. Análisis con máquinas de vectores de soporte (SVM)

Al igual que en el caso de la detección de púlsares, para la clasificación de objetos estelares también se va a probar un modelo de predicción obtenido a partir de algoritmos SVM. En este caso, al tratarse de una clasificación de tres clases (estrella, galaxia o cuásar) se emplearán las técnicas SVM multiclase explicadas en 3.2.

En este caso se utilizará  $\frac{1}{6}$  del total de las 100000 muestras para entrenar el modelo, es decir, 16667 muestras; mientras que los  $\frac{5}{6}$  restantes, 83333 muestras, se emplearán para probar el clasificador obtenido. Dichas muestras se elegirán de forma aleatoria de entre el total. Se emplean las mismas funciones y configuraciones de las herramientas que en el caso de la detección de púlsares para la construcción del modelo.

Como datos de **entrada** a la red se utilizan los valores de los 17 parámetros que caracterizan cada una de las observaciones. Los **resultados** están recogidos en la Tabla 6.1. Se representa la tasa de clasificación del modelo de predicción junto a la matriz de confusión de la que se ha extraído y el tiempo de entrenamiento.

**Tabla 6.1:** Resultados para el experimento 2C, donde se ha empleado un modelo de predicción creado con algoritmos SVM.

$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
$\begin{pmatrix} 18055 & 0 & 0 \\ 15746 & 0 & 0 \\ 49532 & 0 & 0 \end{pmatrix}$	21,67 %	00:03:27

A la vista de los resultados se puede concluir que no se logran unos resultados aceptables. Se realizan varias pruebas, tanto empleando las distintas funciones y métodos de las librerías empleadas como *fitcecoc* que realizan entrenamientos de clasificadores SVM en *MatLab*, como la programación manual de las técnicas expuestas en 3.2. Asimismo, se prueba a normalizar los datos de entrada en función del máximo y mínimo valor de cada tipo de dato, además de aumentar y disminuir el número de muestras empleado para entrenar el modelo y probarlo. Tras un gran número de experimentos se acaba comprobando cómo con ninguna de dichas modificaciones se consiguen crear modelos que sean capaces de clasificar los datos de entrada de forma correcta.

De esta manera, se concluye que el mejor modelo obtención para la clasificación de objetos estelares es el creado empleando redes neuronales artificiales descrito en 6.2.

## 7. Resumen y conclusiones

### 7.1. Resumen

Los campos de la investigación espacial, astronomía y astrofísica son ámbitos en los que la aplicación de técnicas de *machine learning* es de un gran interés, debido al enorme volumen de los datos que se deben analizar en muchas investigaciones. Con el objetivo de analizar la viabilidad de la implementación de este tipo de técnicas se han obtenido modelos empleando dos de ellas: redes neuronales artificiales y algoritmos SVM. Junto con ello, se han utilizado tres bases de datos, dos con los datos ya tratados y organizados, para la detección de púlsares y la clasificación de objetos estelares; y otra, empleando imágenes obtenidas por las *Hazcams* del rover *Curiosity*, en la que ha sido necesario realizar un tratamiento de datos, mediante el etiquetado y posprocesado de las imágenes. Además, entre las mejoras introducidas para el refinamiento del modelo se ha implementado la técnica del cálculo de *histogmas de gradientes orientados (HOG)*.

Debido al carácter empírico del diseño y aplicación de los citados modelos de predicción, para cada una de las bases de datos y cada tipo de herramienta empleada se ha realizado una variación de los parámetros de diseño. Con ello, se han analizado los resultados empleando técnicas como la matriz de confusión, la tasa de clasificación o el tiempo de entrenamiento, mediante el análisis de las cuales se han ido realizando ajustes y variaciones a los modelos con el objetivo de obtener el mayor rendimiento posible.

### 7.2. Conclusiones

El objetivo principal del este trabajo es evaluar las posibilidades de resolución de distintos problemas relacionados con la astronomía y la investigación espacial empleando distintas técnicas de *machine learning*. Se ha comprobado cómo empleando distintas técnicas, como son las redes neuronales o los algoritmos SVM, se pueden obtener modelos de predicción bastante precisos para la clasificación de distintos datos.

En el caso de la base de datos creada a partir de las imágenes de *Hazcam*, se ha observado cómo el etiquetado del nivel de peligro que representaban los paisajes de las capturas que se ha realizado en un principio no ha sido adecuado. Dicha conclusión nace de que, en el caso de la redes neuronales, se ha visto cómo no había ningún tipo de mejora al modificar ciertos parámetros del modelo, como son el etiquetado de ciertos niveles de forma más clara, la modificación del número de entradas de la red, proveniente de la creación de *superpíxeles* en la imágenes originales, o la variación de la topología de la red. Por tanto, se establece cómo la única forma de obtener unos resultados aceptables el la agrupación de las etiquetas en dos grupos, unas señalando el terreno como peligroso y otras como sin peligro. A partir de aquí se emplea la técnica de HOG con el objetivo de ver si aporta una cierta mejora. Se ha visto como se consigue aumentar la tasa de acierto del 90,90 % al 94,91 % empleando este tipo de técnicas, por lo que se concluye que es óptimo su empleo como entrada a la red neuronal para esta base de datos.

En cuanto a las otras dos bases de datos ya tratadas, las correspondientes a la detección de púlsares y la clasificación de objetos estelares, se comprueba cómo los modelos creados empleando redes neuronales dan unos resultados muy satisfactorios, con unas tasas de clasificación en el mejor de los casos del 98,13 % y 95,97 %, respectivamente. Además, la construcción de dichos modelos tiene un coste computacionales relativamente bajo. En cuanto a los algoritmos SVM, en el caso de la detección de púlsares estos ofrecen unos resultados similares en cuanto a tasas de clasificación a los obtenidos con redes neuronales en el mejor de los casos, pero con un coste computacional mayor. Por ello, se eligen los primeros como más óptimos en este caso. Por otro lado, en el caso de la clasificación de los objetos estelares, tras la realización de un gran número de pruebas no se ha conseguido obtener un modelo de clasificación capaz de distinguir entre los objetos pertenecientes a las tres clases. De esta manera, se concluye que el mejor modelo obtenido es el expuesto anteriormente empleando redes neuronales artificiales.

### 7.3. Futuras actividades derivadas del TFG

En cuanto a las continuación de las tareas realizadas en este trabajo, el muchas veces expresado carácter empírico del diseño de todos estos modelos de predicción hace que una clara área de expansión del trabajo sea el continuar con las pruebas realizadas para cada base de datos. De esta manera, a más pruebas con modificaciones de parámetros de los distintos modelos, mayor ajuste y refinamiento de los mismos, lo cual conlleva idealmente a mejores resultados.

Por otro lado, a la vista de los resultados, un tema claro en el que seguir desarrollando el trabajo es seguir realizando pruebas con el algoritmo de clasificación SVM multiclase empleado en 6.3. En dichos experimentos no se ha conseguido crear un buen modelo de clasificación, ni empleando las herramientas y funciones dentro de las bibliotecas utilizadas, ni mediante la programación manual de las mismas. Es por ello por lo que, con más tiempo, se podría seguir investigando la causa de dicho bajo rendimiento del modelo de predicción.

Asimismo, en el trabajo únicamente se han empleado redes neuronales artificiales y algoritmos SVM, por lo que en otro aspecto en el que se podría trabajar sería utilizar otro tipo de técnicas de *machine learning*, como podrían ser las redes neuronales convoluciones, algoritmos de *deep learning*, árboles de decisión, *k-means*, etc.

Además, también se podría trabajar en el procesamiento de datos. En el caso de la base de datos de las imágenes tomadas con *Hazcam* se ha visto como el etiquetado realizado no era el adecuado. Con el objetivo de obtener un modelo capaz de distinguir entre más de un nivel de peligro es necesario etiquetar las imágenes de forma correcta. De igual manera, en los tres casos se podrían implementar técnicas relacionadas con el preprocesados de los datos, como la eliminación de *outliers* o *data augmentation*, entre otras. Se ha comprobado como su empleo normalmente conlleva una mejora de los resultados.

# Referencias

- [1] E. Mining, *Machine Learning: 4 Books in 1: Basic Concepts + Artificial Intelligence + Python Programming + Python Machine Learning. A Comprehensive Guide to Build Intelligent Systems Using Python Libraries*. Independently published, 2020.
- [2] E. Alpaydin, *Introduction to Machine Learning, fourth edition (Adaptive Computation and Machine Learning series)*, fourth edition ed. The MIT Press, 2020.
- [3] Harvard, “Machine Learning in Astronomical Data Analysis,” 01 2017. [Online]. Available: <https://hea-www.harvard.edu/AstroStat/aas233/special.html>
- [4] A. C. Strickland, “Mars exploration: Why are we so fascinated with the red planet?” 04 2021. [Online]. Available: <https://edition.cnn.com/2021/04/15/world/mars-exploration-humankind-science/index.html>
- [5] R. M. Haberle, J. Holton, J. Curry, and J. Pyle, *Planetary atmospheres: Mars*, 1st ed. Academic Press, 2003, pp. 1745–1755.
- [6] NASA, “Curiosity Self-Portrait at ‘Big Sky’ Drilling Site,” 08 2017. [Online]. Available: <https://www.nasa.gov/image-feature/jpl/pia19920/curiosity-self-portrait-at-big-sky-drilling-site/>
- [7] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, R. Gellert, J. B. Gilbert, M. Golombek, J. Gómez-Elvira, D. M. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. C. Malin, I. Mitrofanov, J. J. Simmonds, D. Vaniman, R. V. Welch, and R. C. Wiens, “Mars Science Laboratory Mission and Science Investigation,” *Space Science Reviews*, vol. 170, no. 1-4, pp. 5–56, 2012.
- [8] C. Cofield, “What Are Pulsars?” 04 2016. [Online]. Available: <https://www.space.com/32661-pulsars.html>
- [9] R. J. Lyon, “HTRU2 Data Set,” 2017. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/HTRU2>
- [10] ESA, “How many stars are there in the Universe?” [Online]. Available: [https://www.esa.int/Science\\_Exploration/Space\\_Science/Herschel/How\\_many\\_stars\\_are\\_there\\_in\\_the\\_Universe](https://www.esa.int/Science_Exploration/Space_Science/Herschel/How_many_stars_are_there_in_the_Universe)
- [11] J. S. Allen, “The Classification of Stellar Spectra.” [Online]. Available: [http://www.star.ucl.ac.uk/%7Eepac/spectral\\_classification.html](http://www.star.ucl.ac.uk/%7Eepac/spectral_classification.html)
- [12] J. S. Almeida, J. A. L. Aguerri, C. Muñoz-Tuñón, and A. de Vicente, “AUTOMATIC UNSUPERVISED CLASSIFICATION OF ALL SLOAN DIGITAL SKY SURVEY DATA RELEASE 7 GALAXY SPECTRA,” *The Astrophysical Journal*, vol. 714, no. 1, pp. 487–504, 2010.
- [13] V. A. Cosmos, “Cuásares o Quásar: ¿Qué son?, Características y más,” 04 2022. [Online]. Available: <https://www.postposmo.com/cuasares/>

- [14] G. Palm, “Warren mcculloch and walter pitts: A logical calculus of the ideas immanent in nervous activity,” Berlin, Heidelberg, pp. 229–230, 1986.
- [15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*, illustrated ed. The MIT Press, 2016.
- [16] E. Mendez Garces and G. Mafla, “Conceptual model based on artificial neural networks oriented to the prediction of carbon dioxide (co2) as a pollution index,” 11 2019.
- [17] D. Brys, “Neural Networks – Crash course,” 04 2021. [Online]. Available: <https://saince.io/2020/10/07/neural-networks-crash-course/>
- [18] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [19] S. Mallick, “Image Recognition and Object Detection : Part 1 — LearnOpenCV ,” 05 2021. [Online]. Available: <https://learnopencv.com/image-recognition-and-object-detection-part1>
- [20] —, “Histogram of Oriented Gradients explained using OpenCV,” 11 2021. [Online]. Available: <https://learnopencv.com/histogram-of-oriented-gradients/>
- [21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [22] NASA, “PDS Imaging Node - MSL,” 12 2021. [Online]. Available: <https://pds-imaging.jpl.nasa.gov/data/msl/>
- [23] —, “PDS Imaging Node - MSL - Hazcam,” 12 2021. [Online]. Available: [https://pds-imaging.jpl.nasa.gov/data/msl/MSLHAZ\\_0XXX/EXTRAS/FULL/](https://pds-imaging.jpl.nasa.gov/data/msl/MSLHAZ_0XXX/EXTRAS/FULL/)
- [24] J. Heaton, *Introduction to Neural Networks with Java, 2nd Edition*, 2nd ed. Heaton Research, Incorporated, pp. 158–159.
- [25] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles, “Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach,” *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 1, pp. 1104–1123, 2016.
- [26] fedesoriano, “Stellar Classification Dataset - SDSS17,” 01 2022. [Online]. Available: <https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17>
- [27] Abdurro’uf, K. Accetta, C. Aerts, V. Silva Aguirre, R. Ahumada, N. Ajgaonkar, N. Filiz Ak, S. Alam, C. Allende Prieto, A. Almeida, F. Anders, S. F. Anderson, B. H. Andrews, B. Anguiano, E. Aquino-Ortíz, A. Aragón-Salamanca, M. Argudo-Fernández, M. Ata, M. Aubert, V. Avila-Reese, C. Badenes, R. H. Barbá, K. Barger, J. K. Barrera-Ballesteros, R. L. Beaton, T. C. Beers, F. Belfiore, C. F. Bender, M. Bernardi, M. A. Bershad, F. Beutler, C. M. Bidin, J. C. Bird, D. Bizyaev, G. A. Blanc, M. R. Blanton, N. F. Boardman, A. S. Bolton, M. Boquien, J. Borissova, J. Bovy, W. N. Brandt, J. Brown, J. R. Brownstein, M. Brusa, J. Buchner, K. Bundy, J. N.

Burchett, M. Bureau, A. Burgasser, T. K. Cabang, S. Campbell, M. Cappellari, J. K. Carlberg, F. C. Wanderley, R. Carrera, J. Cash, Y.-P. Chen, W.-H. Chen, B. Cherinka, C. Chiappini, P. D. Choi, S. D. Chojnowski, H. Chung, N. Clerc, R. E. Cohen, J. M. Comerford, J. Comparat, L. da Costa, K. Covey, J. D. Crane, I. Cruz-Gonzalez, C. Culhane, K. Cunha, Y. S. Dai, G. Damke, J. Darling, J. W. Davidson Jr., R. Davies, K. Dawson, N. De Lee, A. M. Diamond-Stanic, M. Cano-Díaz, H. D. Sánchez, J. Donor, C. Duckworth, T. Dwelly, D. J. Eisenstein, Y. P. Elsworth, E. Emselem, M. Eracleous, S. Escoffier, X. Fan, E. Farr, S. Feng, J. G. Fernández-Trincado, D. Feuillet, A. Filipp, S. P. Fillingham, P. M. Frinchaboy, S. Fromenteau, L. Galbany, R. A. García, D. A. García-Hernández, J. Ge, D. Geisler, J. Gelfand, T. Géron, B. J. Gibson, J. Goddy, D. Godoy-Rivera, K. Grabowski, P. J. Green, M. Greener, C. J. Grier, E. Griffith, H. Guo, J. Guy, M. Hadjara, P. Harding, S. Hasselquist, C. R. Hayes, F. Hearty, J. Hernández, L. Hill, D. W. Hogg, J. A. Holtzman, D. Horta, B.-C. Hsieh, C.-H. Hsu, Y.-H. Hsu, D. Huber, M. Huertas-Company, B. Hutchinson, H. S. Hwang, H. J. Ibarra-Medel, J. I. Chitham, G. S. Ilha, J. Imig, W. Jaekle, T. Jayasinghe, X. Ji, J. A. Johnson, A. Jones, H. Jönsson, I. Katkov, D. A. Khalatyan, K. Kinemuchi, S. Kisku, J. H. Knapen, J.-P. Kneib, J. A. Kollmeier, M. Kong, M. Kounkel, K. Kreckel, D. Krishnarao, I. Lacerna, R. R. Lane, R. Langgin, R. Laverder, D. R. Law, D. Lazarz, H. W. Leung, H.-H. Leung, H. M. Lewis, C. Li, R. Li, J. Lian, F.-H. Liang, L. Lin, Y.-T. Lin, S. Lin, C. Lintott, D. Long, P. Longa-Peña, C. López-Cobá, S. Lu, B. F. Lundgren, Y. Luo, J. T. Mackereth, A. de la Macorra, S. Mahadevan, S. R. Majewski, A. Manchado, T. Mandeville, C. Maraston, B. Margalef-Bentabol, T. Masseron, K. L. Masters, S. Mathur, R. M. McDermid, M. Mckay, A. Merloni, M. Merrifield, S. Meszaros, A. Miglio, F. Di Mille, D. Minniti, R. Minsley, A. Monachesi, J. Moon, B. Mosser, J. Mulchaey, D. Muna, R. R. Muñoz, A. D. Myers, N. Myers, S. Nadathur, P. Nair, K. Nandra, J. Neumann, J. A. Newman, D. L. Nidever, F. Nikakhtar, C. Nitschelm, J. E. O’Connell, L. Garma-Oehmichen, G. Luan Souza de Oliveira, R. Olney, D. Oravetz, M. Ortigoza-Urdaneta, Y. Osorio, J. Otter, Z. J. Pace, N. Padilla, K. Pan, H.-A. Pan, T. Parikh, J. Parker, S. Peirani, K. Peña Ramírez, S. Penny, W. J. Percival, I. Perez-Fournon, M. Pinsonneault, F. Poidevin, V. J. Poovelil, A. M. Price-Whelan, A. Bárbara de Andrade Queiroz, M. J. Raddick, A. Ray, S. B. Rembold, N. Riddle, R. A. Riffel, R. Riffel, H.-W. Rix, A. C. Robin, A. Rodríguez-Puebla, A. Roman-Lopes, C. Román-Zúñiga, B. Rose, A. J. Ross, G. Rossi, K. H. R. Rubin, M. Salvato, S. F. Sánchez, J. R. Sánchez-Gallego, R. Sanderson, F. A. Santana Rojas, E. Sarceno, R. Sarmiento, C. Sayres, E. Sazonova, A. L. Schaefer, R. Schiavon, D. J. Schlegel, D. P. Schneider, M. Schultheis, A. Schwöpe, A. Serenelli, J. Serna, Z. Shao, G. Shapiro, A. Sharma, Y. Shen, M. Shetrone, Y. Shu, J. D. Simon, M. F. Skrutskie, R. Smethurst, V. Smith, J. Sobeck, T. Spoo, D. Sprague, D. V. Stark, K. G. Stassun, M. Steinmetz, D. Stello, A. Stone-Martinez, T. Storchi-Bergmann, G. S. Stringfellow, A. Stutz, Y.-C. Su, M. Taghizadeh-Popp, M. S. Talbot, J. Tayar, E. Telles, J. Teske, A. Thakar, C. Theissen, A. Tkachenko, D. Thomas, R. Tojeiro, H. Hernandez Toledo, N. W. Troup, J. R. Trump, J. Trussler, J. Turner, S. Tuttle, E. Unda-Sanzana, J. A. Vázquez-Mata, M. Valentini, O. Valenzuela, J. Vargas-González, M. Vargas-Magaña, P. V. Alfaro, S. Villanova, F. Vincenzo, D. Wake, J. T. Warfield, J. D. Washington, B. A. Weaver, A.-M. Weijmans, D. H. Weinberg, A. Weiss, K. B. Westfall, V. Wild, M. C. Wilde, J. C. Wilson, R. F. Wilson, M. Wilson, J. Wolf, W. M. Wood-Vasey, R. Yan, O. Zamora, G. Zasowski, K. Zhang, C. Zhao, Z. Zheng, Z. Zheng, and K. Zhu, “The Seventeenth Data Re-

lease of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar, and APOGEE-2 Data,” *The Astrophysical Journal Supplement Series*, vol. 259, no. 2, p. 35, 2022.



## A. Resultados de los experimentos

### A.1. Detección de obstáculos en imágenes obtenidas con *Hazcam*

#### Experimento 1A

Tabla A.1: Resultados experimento 1A con 1 capa oculta.

Neuronas	$M_C$	$T_{clas}$ (%)	$t_{entr}$ (hh:mm:ss)
500	$\begin{pmatrix} 203 & 368 & 144 & 46 & 14 & 9 & 5 \\ 316 & 495 & 219 & 68 & 23 & 15 & 6 \\ 98 & 169 & 112 & 47 & 12 & 8 & 3 \\ 13 & 30 & 28 & 21 & 8 & 5 & 1 \\ 10 & 12 & 6 & 10 & 8 & 9 & 2 \\ 2 & 1 & 1 & 3 & 3 & 6 & 1 \\ 5 & 14 & 13 & 14 & 19 & 59 & 318 \end{pmatrix}$	38,75	00:01:27
1000	$\begin{pmatrix} 192 & 354 & 157 & 54 & 17 & 12 & 4 \\ 267 & 518 & 232 & 81 & 21 & 15 & 5 \\ 77 & 178 & 129 & 42 & 14 & 8 & 3 \\ 11 & 27 & 31 & 21 & 11 & 4 & 1 \\ 8 & 13 & 10 & 8 & 8 & 7 & 3 \\ 1 & 2 & 2 & 3 & 2 & 6 & 1 \\ 5 & 13 & 20 & 24 & 21 & 62 & 296 \end{pmatrix}$	38,95	00:10:57
2000	$\begin{pmatrix} 237 & 344 & 136 & 41 & 16 & 9 & 6 \\ 317 & 501 & 215 & 60 & 23 & 17 & 7 \\ 104 & 158 & 121 & 45 & 11 & 8 & 3 \\ 14 & 27 & 27 & 23 & 9 & 4 & 1 \\ 11 & 11 & 8 & 9 & 8 & 8 & 2 \\ 1 & 1 & 2 & 2 & 4 & 6 & 2 \\ 5 & 13 & 12 & 19 & 15 & 45 & 332 \end{pmatrix}$	40,88	00:23:44
2500	$\begin{pmatrix} 279 & 285 & 146 & 47 & 16 & 11 & 5 \\ 424 & 404 & 205 & 72 & 21 & 11 & 3 \\ 151 & 135 & 101 & 41 & 13 & 7 & 2 \\ 31 & 19 & 25 & 19 & 9 & 3 & 1 \\ 19 & 11 & 8 & 6 & 6 & 6 & 1 \\ 4 & 1 & 2 & 2 & 4 & 4 & 1 \\ 91 & 10 & 16 & 17 & 19 & 52 & 235 \end{pmatrix}$	34,95	00:28:54
3000	$\begin{pmatrix} 177 & 372 & 154 & 50 & 19 & 11 & 6 \\ 266 & 500 & 246 & 84 & 24 & 14 & 6 \\ 77 & 168 & 131 & 48 & 13 & 8 & 3 \\ 12 & 27 & 29 & 22 & 11 & 4 & 2 \\ 9 & 9 & 8 & 9 & 13 & 6 & 2 \\ 1 & 2 & 2 & 2 & 2 & 8 & 1 \\ 3 & 16 & 16 & 18 & 19 & 59 & 310 \end{pmatrix}$	38,69	00:39:07

**Tabla A.2:** Resultados experimento 1A con 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
500 333	$\begin{pmatrix} 163 & 393 & 151 & 49 & 19 & 10 & 4 \\ 255 & 531 & 239 & 70 & 26 & 14 & 5 \\ 78 & 169 & 134 & 47 & 12 & 8 & 3 \\ 11 & 27 & 27 & 29 & 8 & 3 & 2 \\ 9 & 12 & 8 & 10 & 10 & 5 & 2 \\ 1 & 1 & 1 & 3 & 4 & 6 & 0 \\ 3 & 15 & 15 & 22 & 24 & 57 & 305 \end{pmatrix}$	39,25	00:06:41
1000 667	$\begin{pmatrix} 268 & 309 & 138 & 44 & 16 & 11 & 3 \\ 413 & 422 & 208 & 60 & 19 & 14 & 4 \\ 148 & 146 & 99 & 35 & 10 & 8 & 4 \\ 28 & 27 & 25 & 13 & 9 & 4 & 1 \\ 18 & 12 & 9 & 7 & 6 & 3 & 2 \\ 4 & 1 & 2 & 3 & 2 & 5 & 0 \\ 90 & 13 & 17 & 21 & 18 & 55 & 227 \end{pmatrix}$	34,61	00:11:42
2000 1333	$\begin{pmatrix} 211 & 375 & 132 & 37 & 14 & 13 & 7 \\ 285 & 525 & 229 & 61 & 21 & 13 & 6 \\ 91 & 165 & 124 & 47 & 14 & 6 & 3 \\ 9 & 28 & 31 & 23 & 8 & 4 & 2 \\ 8 & 10 & 9 & 8 & 12 & 9 & 2 \\ 1 & 1 & 2 & 2 & 2 & 8 & 2 \\ 4 & 11 & 13 & 12 & 13 & 48 & 341 \end{pmatrix}$	41,45	00:46:52
2500 1667	$\begin{pmatrix} 181 & 366 & 158 & 53 & 19 & 8 & 3 \\ 264 & 519 & 236 & 73 & 30 & 13 & 4 \\ 71 & 181 & 127 & 45 & 18 & 5 & 2 \\ 9 & 28 & 29 & 23 & 12 & 3 & 2 \\ 6 & 13 & 9 & 10 & 12 & 7 & 0 \\ 1 & 2 & 2 & 2 & 3 & 5 & 2 \\ 2 & 12 & 15 & 21 & 23 & 60 & 309 \end{pmatrix}$	39,20	01:13:38
3000 2000	$\begin{pmatrix} 128 & 386 & 190 & 48 & 21 & 11 & 5 \\ 203 & 552 & 262 & 81 & 25 & 15 & 3 \\ 62 & 179 & 133 & 49 & 12 & 11 & 4 \\ 7 & 26 & 32 & 24 & 11 & 4 & 2 \\ 6 & 13 & 8 & 10 & 10 & 7 & 3 \\ 1 & 2 & 1 & 3 & 3 & 6 & 1 \\ 1 & 13 & 20 & 22 & 27 & 64 & 295 \end{pmatrix}$	38,25	01:55:29
4000 2667	$\begin{pmatrix} 134 & 405 & 166 & 51 & 19 & 10 & 4 \\ 214 & 536 & 264 & 84 & 26 & 14 & 3 \\ 64 & 183 & 131 & 46 & 13 & 9 & 3 \\ 7 & 30 & 30 & 25 & 9 & 3 & 1 \\ 7 & 15 & 8 & 9 & 11 & 6 & 2 \\ 1 & 1 & 3 & 2 & 3 & 6 & 1 \\ 1 & 14 & 18 & 19 & 23 & 64 & 302 \end{pmatrix}$	38,16	02:57:19

**Tabla A.3:** Resultados experimento 1A con 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
500 333 222	$\left( \begin{array}{ccccccc} 410 & 233 & 96 & 32 & 9 & 7 & 2 \\ 614 & 312 & 142 & 46 & 14 & 10 & 2 \\ 231 & 100 & 74 & 30 & 9 & 5 & 1 \\ 50 & 19 & 16 & 11 & 7 & 3 & 0 \\ 28 & 7 & 6 & 6 & 5 & 4 & 1 \\ 7 & 1 & 2 & 2 & 1 & 4 & 0 \\ 177 & 8 & 7 & 11 & 15 & 42 & 179 \end{array} \right)$	32,20	00:14:45
1000 667 445	$\left( \begin{array}{ccccccc} 174 & 388 & 151 & 47 & 15 & 11 & 3 \\ 282 & 510 & 236 & 68 & 26 & 12 & 6 \\ 94 & 174 & 114 & 45 & 12 & 8 & 3 \\ 12 & 30 & 28 & 21 & 8 & 5 & 1 \\ 11 & 14 & 9 & 7 & 9 & 6 & 2 \\ 1 & 2 & 3 & 2 & 3 & 6 & 1 \\ 3 & 13 & 12 & 21 & 23 & 62 & 307 \end{array} \right)$	37,99	00:18:58
2000 1333 889	$\left( \begin{array}{ccccccc} 269 & 307 & 149 & 40 & 12 & 8 & 4 \\ 401 & 430 & 215 & 64 & 16 & 11 & 4 \\ 142 & 144 & 103 & 37 & 13 & 8 & 3 \\ 30 & 22 & 26 & 18 & 9 & 2 & 1 \\ 18 & 10 & 8 & 9 & 6 & 5 & 1 \\ 4 & 1 & 1 & 2 & 4 & 4 & 1 \\ 91 & 11 & 14 & 20 & 20 & 52 & 234 \end{array} \right)$	35,45	01:16:56
2500 1667 1111	$\left( \begin{array}{ccccccc} 189 & 389 & 140 & 41 & 16 & 10 & 4 \\ 274 & 533 & 226 & 66 & 22 & 15 & 5 \\ 86 & 176 & 120 & 44 & 14 & 7 & 3 \\ 11 & 31 & 28 & 21 & 9 & 4 & 1 \\ 8 & 12 & 8 & 10 & 11 & 6 & 2 \\ 1 & 1 & 2 & 2 & 2 & 7 & 2 \\ 3 & 12 & 15 & 16 & 21 & 54 & 320 \end{array} \right)$	40,00	01:19:49
3000 2000 1333	$\left( \begin{array}{ccccccc} 133 & 378 & 181 & 65 & 18 & 10 & 5 \\ 212 & 517 & 276 & 88 & 31 & 13 & 3 \\ 59 & 181 & 129 & 54 & 16 & 9 & 2 \\ 7 & 27 & 34 & 21 & 10 & 4 & 2 \\ 7 & 12 & 10 & 8 & 10 & 9 & 1 \\ 1 & 2 & 1 & 3 & 4 & 5 & 1 \\ 1 & 15 & 20 & 23 & 24 & 69 & 290 \end{array} \right)$	36,82	01:45:10
4000 2667 1778	$\left( \begin{array}{ccccccc} 178 & 390 & 145 & 46 & 14 & 10 & 4 \\ 256 & 530 & 250 & 68 & 19 & 12 & 5 \\ 80 & 172 & 127 & 49 & 12 & 8 & 3 \\ 8 & 28 & 31 & 24 & 10 & 3 & 1 \\ 8 & 10 & 10 & 9 & 12 & 8 & 1 \\ 1 & 2 & 2 & 2 & 3 & 6 & 1 \\ 3 & 12 & 13 & 15 & 18 & 55 & 324 \end{array} \right)$	40,03	03:27:13

**Tabla A.4:** Resultados experimento 1A con 4 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
500 333 222 148	$\left( \begin{array}{cccccc} 144 & 393 & 167 & 51 & 18 & 12 & 4 \\ 233 & 538 & 239 & 85 & 24 & 15 & 6 \\ 79 & 185 & 117 & 42 & 13 & 11 & 3 \\ 10 & 34 & 29 & 18 & 10 & 3 & 1 \\ 9 & 17 & 9 & 6 & 7 & 6 & 2 \\ 1 & 2 & 2 & 3 & 3 & 5 & 1 \\ 3 & 15 & 26 & 29 & 27 & 68 & 274 \end{array} \right)$	36,72	00:06:06
1000 667 445 297	$\left( \begin{array}{cccccc} 300 & 309 & 114 & 39 & 15 & 8 & 4 \\ 442 & 412 & 192 & 59 & 18 & 14 & 4 \\ 161 & 138 & 85 & 44 & 12 & 7 & 4 \\ 32 & 26 & 21 & 16 & 6 & 5 & 1 \\ 18 & 10 & 8 & 7 & 6 & 5 & 2 \\ 4 & 2 & 2 & 2 & 3 & 4 & 1 \\ 91 & 11 & 13 & 19 & 19 & 51 & 238 \end{array} \right)$	35,33	00:24:49
2000 1333 889 593	$\left( \begin{array}{cccccc} 150 & 383 & 167 & 54 & 18 & 12 & 4 \\ 238 & 541 & 244 & 71 & 24 & 18 & 4 \\ 80 & 173 & 121 & 49 & 14 & 9 & 4 \\ 10 & 31 & 25 & 27 & 7 & 5 & 1 \\ 9 & 11 & 11 & 9 & 9 & 5 & 2 \\ 1 & 3 & 1 & 3 & 3 & 6 & 0 \\ 3 & 14 & 19 & 23 & 26 & 63 & 293 \end{array} \right)$	38,27	01:06:36
2500 1667 1111 741	$\left( \begin{array}{cccccc} 160 & 385 & 156 & 55 & 15 & 13 & 5 \\ 233 & 532 & 253 & 77 & 24 & 16 & 5 \\ 77 & 177 & 124 & 49 & 13 & 8 & 2 \\ 10 & 31 & 28 & 23 & 8 & 6 & 1 \\ 8 & 14 & 8 & 11 & 10 & 6 & 1 \\ 1 & 2 & 3 & 2 & 3 & 6 & 1 \\ 4 & 14 & 15 & 20 & 28 & 60 & 301 \end{array} \right)$	38,51	01:27:07
3000 2000 1333 889	$\left( \begin{array}{cccccc} 188 & 380 & 147 & 44 & 14 & 10 & 5 \\ 282 & 519 & 231 & 63 & 24 & 16 & 5 \\ 85 & 176 & 120 & 47 & 12 & 7 & 3 \\ 11 & 29 & 28 & 24 & 9 & 5 & 1 \\ 6 & 14 & 9 & 9 & 10 & 8 & 1 \\ 1 & 2 & 1 & 2 & 5 & 5 & 1 \\ 4 & 11 & 17 & 18 & 18 & 55 & 317 \end{array} \right)$	39,41	02:46:17
4000 2667 1778 1185	$\left( \begin{array}{cccccc} 164 & 392 & 152 & 49 & 16 & 13 & 3 \\ 245 & 522 & 254 & 76 & 25 & 14 & 4 \\ 75 & 172 & 133 & 43 & 15 & 10 & 3 \\ 8 & 31 & 29 & 24 & 9 & 4 & 1 \\ 6 & 12 & 12 & 7 & 11 & 8 & 2 \\ 0 & 1 & 2 & 2 & 3 & 7 & 2 \\ 2 & 13 & 17 & 21 & 22 & 54 & 312 \end{array} \right)$	39,05	03:30:34

## Experimento 1B

Tabla A.5: Resultados experimento 1B con 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
2000	$\left( \begin{array}{cccccccc} 288 & 309 & 125 & 38 & 15 & 8 & 5 & \\ 417 & 430 & 198 & 55 & 20 & 15 & 6 & \\ 150 & 143 & 102 & 32 & 12 & 9 & 2 & \\ 30 & 24 & 22 & 17 & 8 & 3 & 2 & \\ 17 & 9 & 7 & 9 & 8 & 5 & 1 & \\ 4 & 2 & 1 & 3 & 3 & 3 & 1 & \\ 88 & 11 & 13 & 18 & 18 & 43 & 250 & \end{array} \right)$	36,59	01:47:55
3000	$\left( \begin{array}{cccccccc} 133 & 399 & 177 & 48 & 15 & 14 & 4 & \\ 199 & 548 & 274 & 78 & 22 & 15 & 5 & \\ 57 & 170 & 144 & 53 & 14 & 8 & 4 & \\ 10 & 26 & 33 & 24 & 8 & 3 & 1 & \\ 8 & 13 & 8 & 10 & 11 & 6 & 1 & \\ 1 & 2 & 1 & 2 & 3 & 7 & 1 & \\ 3 & 13 & 16 & 18 & 22 & 62 & 306 & \end{array} \right)$	39,12	01:11:12
4000	$\left( \begin{array}{cccccccc} 145 & 396 & 163 & 50 & 19 & 12 & 5 & \\ 222 & 537 & 259 & 74 & 24 & 18 & 5 & \\ 66 & 172 & 135 & 51 & 15 & 8 & 4 & \\ 9 & 27 & 32 & 25 & 8 & 4 & 1 & \\ 7 & 11 & 8 & 14 & 8 & 8 & 1 & \\ 2 & 1 & 1 & 4 & 3 & 6 & 0 & \\ 2 & 13 & 15 & 15 & 21 & 63 & 312 & \end{array} \right)$	38,97	01:41:07
5000	$\left( \begin{array}{cccccccc} 160 & 408 & 147 & 45 & 15 & 9 & 6 & \\ 220 & 569 & 240 & 66 & 21 & 16 & 7 & \\ 63 & 167 & 152 & 44 & 15 & 7 & 3 & \\ 10 & 29 & 27 & 27 & 9 & 3 & 2 & \\ 8 & 12 & 8 & 10 & 13 & 5 & 2 & \\ 1 & 2 & 1 & 2 & 4 & 5 & 1 & \\ 3 & 14 & 18 & 18 & 22 & 57 & 311 & \end{array} \right)$	41,20	02:44:29
6000	$\left( \begin{array}{cccccccc} 174 & 380 & 149 & 48 & 19 & 12 & 7 & \\ 232 & 568 & 235 & 62 & 19 & 16 & 7 & \\ 64 & 178 & 135 & 47 & 13 & 9 & 3 & \\ 8 & 25 & 31 & 28 & 8 & 4 & 2 & \\ 6 & 12 & 9 & 9 & 7 & 11 & 2 & \\ 1 & 2 & 1 & 3 & 3 & 7 & 1 & \\ 3 & 13 & 13 & 16 & 20 & 55 & 321 & \end{array} \right)$	41,39	02:33:04

**Tabla A.6:** Resultados experimento 1B con 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
2000 1333	$\begin{pmatrix} 162 & 408 & 151 & 39 & 15 & 9 & 4 \\ 229 & 562 & 246 & 63 & 21 & 13 & 5 \\ 62 & 177 & 143 & 43 & 13 & 9 & 3 \\ 8 & 30 & 30 & 25 & 9 & 3 & 1 \\ 6 & 11 & 10 & 11 & 10 & 8 & 2 \\ 1 & 3 & 1 & 2 & 3 & 6 & 2 \\ 3 & 16 & 11 & 12 & 18 & 51 & 331 \end{pmatrix}$	41,29	01:07:13
3000 2000	$\begin{pmatrix} 131 & 405 & 170 & 54 & 14 & 10 & 5 \\ 196 & 562 & 273 & 73 & 20 & 12 & 3 \\ 57 & 178 & 149 & 44 & 12 & 8 & 2 \\ 8 & 30 & 33 & 24 & 9 & 2 & 1 \\ 10 & 12 & 10 & 9 & 9 & 6 & 2 \\ 2 & 1 & 1 & 2 & 3 & 7 & 0 \\ 3 & 14 & 16 & 17 & 23 & 60 & 307 \end{pmatrix}$	39,59	01:49:23
4000 2667	$\begin{pmatrix} 196 & 398 & 130 & 35 & 16 & 10 & 4 \\ 245 & 569 & 233 & 60 & 17 & 12 & 4 \\ 66 & 177 & 139 & 46 & 12 & 8 & 2 \\ 9 & 22 & 32 & 28 & 10 & 3 & 2 \\ 5 & 11 & 8 & 9 & 13 & 8 & 2 \\ 1 & 2 & 2 & 2 & 3 & 7 & 1 \\ 2 & 10 & 12 & 14 & 11 & 49 & 342 \end{pmatrix}$	43,15	03:01:37
5000 3333	$\begin{pmatrix} 538 & 166 & 59 & 17 & 4 & 3 & 2 \\ 777 & 227 & 101 & 21 & 7 & 6 & 2 \\ 296 & 70 & 57 & 17 & 6 & 3 & 1 \\ 68 & 11 & 12 & 10 & 3 & 1 & 1 \\ 37 & 6 & 2 & 4 & 5 & 3 & 1 \\ 11 & 0 & 1 & 1 & 2 & 2 & 1 \\ 266 & 6 & 5 & 6 & 6 & 21 & 132 \end{pmatrix}$	32,39	00:58:37
6000 4000	$\begin{pmatrix} 174 & 380 & 149 & 48 & 19 & 12 & 7 \\ 232 & 568 & 235 & 62 & 19 & 16 & 7 \\ 64 & 178 & 135 & 47 & 13 & 9 & 3 \\ 8 & 25 & 31 & 28 & 8 & 4 & 2 \\ 6 & 12 & 9 & 9 & 7 & 11 & 2 \\ 1 & 2 & 1 & 3 & 3 & 7 & 1 \\ 3 & 13 & 13 & 16 & 20 & 55 & 321 \end{pmatrix}$	41,39	02:33:04

**Tabla A.7:** Resultados experimento 1B con 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
2000 1333 889	$\begin{pmatrix} 310 & 309 & 111 & 34 & 12 & 9 & 5 \\ 442 & 435 & 183 & 47 & 17 & 12 & 5 \\ 147 & 145 & 102 & 37 & 11 & 6 & 3 \\ 29 & 24 & 22 & 21 & 7 & 2 & 1 \\ 18 & 7 & 8 & 8 & 9 & 5 & 2 \\ 4 & 1 & 2 & 1 & 4 & 4 & 0 \\ 90 & 10 & 9 & 13 & 14 & 37 & 268 \end{pmatrix}$	38,28	00:56:47
3000 2000 1333	$\begin{pmatrix} 267 & 382 & 90 & 25 & 13 & 8 & 5 \\ 290 & 583 & 184 & 44 & 18 & 13 & 7 \\ 76 & 159 & 154 & 36 & 10 & 10 & 5 \\ 11 & 22 & 31 & 32 & 7 & 2 & 1 \\ 6 & 8 & 6 & 10 & 16 & 9 & 2 \\ 1 & 1 & 1 & 1 & 4 & 7 & 1 \\ 4 & 9 & 9 & 11 & 14 & 35 & 358 \end{pmatrix}$	47,18	02:27:16
4000 2667 1778	$\begin{pmatrix} 122 & 421 & 166 & 53 & 15 & 9 & 3 \\ 187 & 575 & 265 & 73 & 21 & 13 & 5 \\ 59 & 176 & 140 & 51 & 12 & 9 & 2 \\ 6 & 30 & 33 & 25 & 8 & 3 & 1 \\ 7 & 12 & 9 & 12 & 10 & 6 & 1 \\ 1 & 2 & 2 & 1 & 3 & 7 & 1 \\ 1 & 11 & 17 & 15 & 18 & 60 & 318 \end{pmatrix}$	39,90	03:32:33
5000 3333 2222	$\begin{pmatrix} 177 & 413 & 133 & 39 & 13 & 10 & 5 \\ 221 & 594 & 228 & 60 & 17 & 16 & 5 \\ 61 & 168 & 159 & 41 & 13 & 6 & 4 \\ 8 & 23 & 30 & 33 & 9 & 3 & 0 \\ 6 & 7 & 7 & 12 & 19 & 5 & 1 \\ 1 & 2 & 1 & 2 & 4 & 7 & 0 \\ 3 & 11 & 10 & 17 & 16 & 47 & 336 \end{pmatrix}$	44,15	08:54:10

**Tabla A.8:** Resultados experimento 1B con 4 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
2000 1333 889 593	$\begin{pmatrix} 191 & 380 & 136 & 51 & 16 & 11 & 5 \\ 257 & 549 & 231 & 58 & 23 & 16 & 6 \\ 74 & 175 & 131 & 46 & 12 & 10 & 4 \\ 12 & 27 & 29 & 26 & 7 & 4 & 2 \\ 8 & 11 & 9 & 9 & 12 & 6 & 3 \\ 1 & 2 & 2 & 3 & 3 & 6 & 1 \\ 2 & 12 & 16 & 16 & 22 & 53 & 320 \end{pmatrix}$	41,12	01:14:19
3000 2000 1333 889	$\begin{pmatrix} 245 & 377 & 111 & 28 & 11 & 12 & 6 \\ 308 & 561 & 194 & 43 & 18 & 10 & 6 \\ 77 & 186 & 131 & 37 & 11 & 6 & 3 \\ 11 & 27 & 29 & 25 & 8 & 5 & 1 \\ 8 & 9 & 8 & 13 & 10 & 6 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 1 \\ 4 & 9 & 9 & 12 & 15 & 40 & 351 \end{pmatrix}$	44,27	02:32:56
4000 2667 1778 1185	$\begin{pmatrix} 122 & 421 & 166 & 53 & 15 & 9 & 3 \\ 187 & 575 & 265 & 73 & 21 & 13 & 5 \\ 59 & 176 & 140 & 51 & 12 & 9 & 2 \\ 6 & 30 & 33 & 25 & 8 & 3 & 1 \\ 7 & 12 & 9 & 12 & 10 & 6 & 1 \\ 1 & 2 & 2 & 1 & 3 & 7 & 1 \\ 1 & 11 & 17 & 15 & 18 & 60 & 318 \end{pmatrix}$	39,90	03:32:33
5000 3333 2222 1481	$\begin{pmatrix} 177 & 413 & 133 & 39 & 13 & 10 & 5 \\ 221 & 594 & 228 & 60 & 17 & 16 & 5 \\ 61 & 168 & 159 & 41 & 13 & 6 & 4 \\ 8 & 23 & 30 & 33 & 9 & 3 & 0 \\ 6 & 7 & 7 & 12 & 19 & 5 & 1 \\ 1 & 2 & 1 & 2 & 4 & 7 & 0 \\ 3 & 11 & 10 & 17 & 16 & 47 & 336 \end{pmatrix}$	44,15	08:54:10

**Experimento 1F****Tabla A.9:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $80 \times 80$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
300	$\begin{pmatrix} 2341 & 38 \\ 142 & 479 \end{pmatrix}$	93,98	00:00:52
450	$\begin{pmatrix} 2338 & 41 \\ 123 & 498 \end{pmatrix}$	94,54	00:01:28
600	$\begin{pmatrix} 2341 & 38 \\ 232 & 389 \end{pmatrix}$	91,03	00:01:36
800	$\begin{pmatrix} 2351 & 28 \\ 218 & 403 \end{pmatrix}$	91,78	00:02:13



**Tabla A.10:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $80 \times 80$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300 200	$\begin{pmatrix} 2340 & 39 \\ 146 & 475 \end{pmatrix}$	93,84	00:01:34
450 300	$\begin{pmatrix} 2344 & 35 \\ 222 & 399 \end{pmatrix}$	91,41	00:02:26
600 400	$\begin{pmatrix} 2347 & 32 \\ 239 & 382 \end{pmatrix}$	90,96	00:02:56
800 533	$\begin{pmatrix} 2333 & 46 \\ 146 & 475 \end{pmatrix}$	93,59	00:05:42

**Tabla A.11:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $80 \times 80$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300 200 133	$\begin{pmatrix} 2339 & 40 \\ 162 & 459 \end{pmatrix}$	93,25	00:01:49
450 300 200	$\begin{pmatrix} 2334 & 45 \\ 147 & 474 \end{pmatrix}$	93,57	00:02:57
600 400 267	$\begin{pmatrix} 2343 & 36 \\ 141 & 480 \end{pmatrix}$	94,10	00:05:14
800 533 355	$\begin{pmatrix} 2342 & 37 \\ 176 & 445 \end{pmatrix}$	92,91	00:06:50

**Tabla A.12:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $40 \times 40$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000	$\begin{pmatrix} 2336 & 43 \\ 267 & 354 \end{pmatrix}$	89,67	00:11:26
3500	$\begin{pmatrix} 2350 & 29 \\ 201 & 420 \end{pmatrix}$	92,33	00:28:58
5000	$\begin{pmatrix} 2348 & 31 \\ 222 & 399 \end{pmatrix}$	91,56	00:46:11

**Tabla A.13:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $40 \times 40$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000 1333	$\begin{pmatrix} 2345 & 34 \\ 246 & 375 \end{pmatrix}$	90,67	00:21:07
3500 2333	$\begin{pmatrix} 2349 & 30 \\ 221 & 400 \end{pmatrix}$	91,61	02:48:26
5000 3333	$\begin{pmatrix} 2351 & 28 \\ 232 & 389 \end{pmatrix}$	91,35	02:19:51

**Tabla A.14:** Resultados experimento 1F para HOG sobre imagen a resolución completa para una celda de  $40 \times 40$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000 1333 889	$\begin{pmatrix} 2341 & 38 \\ 141 & 480 \end{pmatrix}$	94,04	00:39:28
3500 2333 1555	$\begin{pmatrix} 2349 & 30 \\ 226 & 395 \end{pmatrix}$	91,44	01:20:02
5000 3333 2222	$\begin{pmatrix} 2338 & 41 \\ 112 & 509 \end{pmatrix}$	94,91	03:51:50

**Tabla A.15:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $8 \times 8$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300	$\begin{pmatrix} 2328 & 51 \\ 187 & 434 \end{pmatrix}$	92,06	00:00:38
450	$\begin{pmatrix} 2319 & 60 \\ 221 & 400 \end{pmatrix}$	90,65	00:01:13
600	$\begin{pmatrix} 2317 & 62 \\ 182 & 439 \end{pmatrix}$	91,88	00:02:21
800	$\begin{pmatrix} 2309 & 70 \\ 235 & 386 \end{pmatrix}$	89,82	00:02:23

**Tabla A.16:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $8 \times 8$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300 200	$\begin{pmatrix} 2331 & 48 \\ 282 & 339 \end{pmatrix}$	88,99	00:01:45
450 300	$\begin{pmatrix} 2328 & 51 \\ 297 & 324 \end{pmatrix}$	88,40	00:02:45

**Tabla A.16:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $8 \times 8$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
600 400	$\begin{pmatrix} 2323 & 56 \\ 165 & 456 \end{pmatrix}$	92,61	00:05:19
800 533	$\begin{pmatrix} 2320 & 59 \\ 200 & 421 \end{pmatrix}$	91,37	00:07:14

**Tabla A.17:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $8 \times 8$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300 200 133	$\begin{pmatrix} 2316 & 63 \\ 189 & 432 \end{pmatrix}$	91,60	00:01:53
450 300 200	$\begin{pmatrix} 2327 & 52 \\ 294 & 327 \end{pmatrix}$	88,47	00:02:28
600 400 267	$\begin{pmatrix} 2319 & 60 \\ 211 & 410 \end{pmatrix}$	90,99	00:06:41
800 533 355	$\begin{pmatrix} 2327 & 52 \\ 292 & 329 \end{pmatrix}$	88,56	00:04:48

**Tabla A.18:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $4 \times 4$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000	$\begin{pmatrix} 2339 & 40 \\ 378 & 243 \end{pmatrix}$	86,05	01:06:22
3500	$\begin{pmatrix} 2325 & 54 \\ 184 & 437 \end{pmatrix}$	92,07	00:44:49
5000	$\begin{pmatrix} 2321 & 58 \\ 202 & 419 \end{pmatrix}$	91,34	00:50:32

**Tabla A.19:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $4 \times 4$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000 1333	$\begin{pmatrix} 2319 & 60 \\ 181 & 440 \end{pmatrix}$	91,99	00:35:49
3500 2333	$\begin{pmatrix} 2352 & 27 \\ 351 & 270 \end{pmatrix}$	87,41	03:02:42
5000 3333	$\begin{pmatrix} 2324 & 55 \\ 189 & 432 \end{pmatrix}$	91,84	02:32:27

**Tabla A.20:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $10 \times 10$  para una celda de  $4 \times 4$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
2000 1333 889	$\begin{pmatrix} 2333 & 46 \\ 293 & 328 \end{pmatrix}$	88,71	00:31:13
3500 2333 1555	$\begin{pmatrix} 2321 & 58 \\ 197 & 424 \end{pmatrix}$	91,47	01:47:01
5000 3333 2222	$\begin{pmatrix} 2315 & 64 \\ 205 & 416 \end{pmatrix}$	91,03	03:24:49

**Tabla A.21:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $16 \times 16$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
300	$\begin{pmatrix} 2326 & 53 \\ 197 & 424 \end{pmatrix}$	91,69	00:00:35
450	$\begin{pmatrix} 2339 & 40 \\ 245 & 376 \end{pmatrix}$	90,49	00:00:48
600	$\begin{pmatrix} 2315 & 64 \\ 203 & 418 \end{pmatrix}$	91,10	00:01:11
800	$\begin{pmatrix} 2337 & 42 \\ 252 & 369 \end{pmatrix}$	90,21	00:04:17

**Tabla A.22:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $16 \times 16$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
300 200	$\begin{pmatrix} 2338 & 41 \\ 240 & 381 \end{pmatrix}$	90,62	00:00:50
450 300	$\begin{pmatrix} 2337 & 42 \\ 260 & 361 \end{pmatrix}$	89,93	00:05:14
600 400	$\begin{pmatrix} 2329 & 50 \\ 170 & 451 \end{pmatrix}$	92,67	00:03:02
800 533	$\begin{pmatrix} 2332 & 47 \\ 147 & 474 \end{pmatrix}$	93,55	00:04:31

**Tabla A.23:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $16 \times 16$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
300 200 133	$\begin{pmatrix} 2324 & 55 \\ 195 & 426 \end{pmatrix}$	91,68	00:01:15
450 300 200	$\begin{pmatrix} 2336 & 43 \\ 264 & 357 \end{pmatrix}$	89,75	00:02:25
600 400 267	$\begin{pmatrix} 2334 & 45 \\ 255 & 366 \end{pmatrix}$	90,01	00:04:58
800 533 355	$\begin{pmatrix} 2328 & 51 \\ 172 & 449 \end{pmatrix}$	92,54	00:06:05

**Tabla A.24:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $8 \times 8$  y 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000	$\begin{pmatrix} 2330 & 49 \\ 159 & 462 \end{pmatrix}$	93,07	00:15:09
3500	$\begin{pmatrix} 2328 & 51 \\ 177 & 444 \end{pmatrix}$	92,39	00:24:57
5000	$\begin{pmatrix} 2341 & 38 \\ 237 & 384 \end{pmatrix}$	90,82	01:50:54

**Tabla A.25:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $8 \times 8$  y 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000 1333	$\begin{pmatrix} 2328 & 51 \\ 155 & 466 \end{pmatrix}$	93,15	00:26:04
3500 2333	$\begin{pmatrix} 2344 & 35 \\ 278 & 343 \end{pmatrix}$	89,55	00:45:05
5000 3333	$\begin{pmatrix} 2326 & 53 \\ 174 & 447 \end{pmatrix}$	92,43	01:40:02

**Tabla A.26:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $8 \times 8$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
2000 1333 889	$\begin{pmatrix} 2326 & 53 \\ 152 & 469 \end{pmatrix}$	93,17	00:31:59

**Tabla A.26:** Resultados experimento 1F para HOG sobre imagen con megapíxeles de  $5 \times 5$  para una celda de  $8 \times 8$  y 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
3500 2333 1555	$\begin{pmatrix} 2340 & 39 \\ 257 & 364 \end{pmatrix}$	90,14	01:07:17
5000 3333 2222	$\begin{pmatrix} 2327 & 52 \\ 179 & 442 \end{pmatrix}$	92,29	02:18:17

## A.2. Detección de pulsares

### Experimento 2A

**Tabla A.27:** Resultados experimento 2A con 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
5	$\begin{pmatrix} 16152 & 107 \\ 235 & 1404 \end{pmatrix}$	98,09	00:00:30
10	$\begin{pmatrix} 16157 & 102 \\ 233 & 1406 \end{pmatrix}$	98,13	00:00:21
16	$\begin{pmatrix} 16171 & 88 \\ 381 & 1258 \end{pmatrix}$	97,38	00:00:18
20	$\begin{pmatrix} 16158 & 101 \\ 234 & 1405 \end{pmatrix}$	98,13	00:00:22
50	$\begin{pmatrix} 16166 & 93 \\ 375 & 1264 \end{pmatrix}$	97,38	00:00:29

**Tabla A.28:** Resultados experimento 2A con 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
5 4	$\begin{pmatrix} 16160 & 99 \\ 380 & 1259 \end{pmatrix}$	97,32	00:00:17
10 8	$\begin{pmatrix} 16151 & 108 \\ 234 & 1405 \end{pmatrix}$	98,09	00:00:22
16 13	$\begin{pmatrix} 16147 & 112 \\ 229 & 1410 \end{pmatrix}$	98,10	00:00:26
20 16	$\begin{pmatrix} 16166 & 93 \\ 370 & 1269 \end{pmatrix}$	97,41	00:00:51
50 40	$\begin{pmatrix} 16152 & 107 \\ 230 & 1409 \end{pmatrix}$	98,10	00:00:52

**Tabla A.29:** Resultados experimento 2A con 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
5 4 3	$\begin{pmatrix} 16145 & 114 \\ 235 & 1404 \end{pmatrix}$	98,05	00:00:23
10 8 6	$\begin{pmatrix} 16149 & 111 \\ 228 & 1411 \end{pmatrix}$	98,11	00:00:37
16 13 10	$\begin{pmatrix} 16154 & 105 \\ 230 & 1409 \end{pmatrix}$	98,13	00:00:39
20 16 13	$\begin{pmatrix} 16150 & 109 \\ 232 & 1408 \end{pmatrix}$	98,10	00:00:43
50 40 32	$\begin{pmatrix} 16159 & 100 \\ 234 & 1405 \end{pmatrix}$	98,13	00:01:36

### A.3. Clasificación de objetos estelares

#### Experimento 3A

**Tabla A.30:** Resultados experimento 3A con 1 capa oculta.

Neuronas	$M_C$	$T_{\text{clas}}$ (%)	$t_{\text{entr}}$ (hh:mm:ss)
12	$\begin{pmatrix} 17133 & 142 & 4319 \\ 14 & 13155 & 5792 \\ 424 & 914 & 58107 \end{pmatrix}$	88,40	00:02:59
20	$\begin{pmatrix} 21424 & 168 & 1 \\ 16 & 16448 & 2497 \\ 512 & 1004 & 57930 \end{pmatrix}$	95,80	00:03:52
26	$\begin{pmatrix} 21416 & 177 & 1 \\ 18 & 13148 & 5795 \\ 515 & 921 & 58009 \end{pmatrix}$	92,57	00:04:11
34	$\begin{pmatrix} 21433 & 159 & 2 \\ 9 & 6584 & 12368 \\ 494 & 637 & 58315 \end{pmatrix}$	86,33	00:05:06
50	$\begin{pmatrix} 21412 & 181 & 2 \\ 9 & 3302 & 15650 \\ 489 & 488 & 58468 \end{pmatrix}$	83,18	00:06:44

**Tabla A.31:** Resultados experimento 3A con 2 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
12 10	$\begin{pmatrix} 21461 & 130 & 3 \\ 5 & 16199 & 2757 \\ 506 & 1024 & 57915 \end{pmatrix}$	95,58	00:05:01
20 16	$\begin{pmatrix} 21454 & 138 & 2 \\ 6 & 16558 & 2397 \\ 509 & 1065 & 57872 \end{pmatrix}$	95,88	00:07:02
26 21	$\begin{pmatrix} 21462 & 131 & 2 \\ 5 & 13337 & 5619 \\ 495 & 884 & 58067 \end{pmatrix}$	92,86	00:07:02
34 27	$\begin{pmatrix} 21445 & 147 & 2 \\ 6 & 16609 & 2347 \\ 493 & 1032 & 57920 \end{pmatrix}$	95,97	00:12:31
50 40	$\begin{pmatrix} 21423 & 168 & 3 \\ 8 & 13295 & 5658 \\ 482 & 836 & 58127 \end{pmatrix}$	92,84	00:17:11

**Tabla A.32:** Resultados experimento 3A con 3 capas ocultas.

Neuronas	$M_C$	$T_{\text{clas}} (\%)$	$t_{\text{entr}} (\text{hh:mm:ss})$
12 10 8	$\begin{pmatrix} 21473 & 115 & 5 \\ 4 & 16128 & 2828 \\ 508 & 1070 & 57867 \end{pmatrix}$	95,47	00:07:48
20 16 13	$\begin{pmatrix} 21470 & 121 & 3 \\ 4 & 16433 & 2523 \\ 498 & 1066 & 57881 \end{pmatrix}$	95,78	00:11:35
26 21 17	$\begin{pmatrix} 21441 & 149 & 4 \\ 4 & 16291 & 2666 \\ 485 & 999 & 57961 \end{pmatrix}$	95,69	00:11:15
34 27 22	$\begin{pmatrix} 21453 & 138 & 3 \\ 5 & 16399 & 2557 \\ 487 & 970 & 57988 \end{pmatrix}$	95,84	00:15:42
50 40 32	$\begin{pmatrix} 21446 & 144 & 4 \\ 6 & 16499 & 2456 \\ 479 & 954 & 58012 \end{pmatrix}$	95,96	00:23:46



## B. Especificaciones técnicas de los equipos empleados

**Tabla B.1:** Especificaciones técnicas del equipo en el que se han realizado los entrenamientos.

<b>Modelo</b>	MacBook Pro (13-inch, 2017)
<b>Procesador</b>	2,3 GHz Intel Core i5 de doble núcleo
<b>Memoria RAM</b>	8 GB 2133 MHz LPDDR3
<b>Gráficos</b>	Intel Iris Plus Graphics 640 1536 MB
<b>Almacenamiento</b>	128 Gb