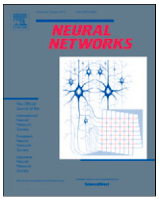
Neural Networks 152 (2022) 150–159

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Spatiotemporal CNN with Pyramid Bottleneck Blocks: Application to eye blinking detection

S.E. Bekhouche [b,c], I. Kajo [b], Y. Ruichek [b], F. Dornaika [a,c,d,*]

[a] *School of Computer and Information Engineering, Henan University, Kaifeng, China*
[b] *CIAD, University Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France*
[c] *University of the Basque Country UPV/EHU, San Sebastian, Spain*
[d] *IKERBASQUE, Basque Foundation for Science, Bilbao, Spain*

ARTICLE INFO

Article history:
Received 28 November 2021
Received in revised form 14 February 2022
Accepted 11 April 2022
Available online 18 April 2022

Keywords:
Eye blinking
Pyramid Bottleneck Blocks
Spatiotemporal CNN
Incremental SVD
Facial landmarks

ABSTRACT

Eye blink detection is a challenging problem that many researchers are working on because it has the potential to solve many facial analysis tasks, such as face anti-spoofing, driver drowsiness detection, and some health disorders. There have been few attempts to detect blinking in the wild scenario, while most of the work has been done under controlled conditions. Moreover, current learning approaches are designed to process sequences that contain only a single blink ignoring the case of the presence of multiple eye blinks. In this work, we propose a fast framework for eye blink detection and eye blink verification that can effectively extract multiple blinks from image sequences considering several challenges such as lighting changes, variety of poses, and change in appearance. The proposed framework employs fast landmarks detector to extract multiple facial key points including the ones that identify the eye regions. Then, an SVD-based method is proposed to extract the potential eye blinks in a moving time window that is updated with new images every second. Finally, the detected blink candidates are verified using a 2D Pyramidal Bottleneck Block Network (PBBN). We also propose an alternative approach that uses a sequence of frames instead of an image as input and employs a continuous 3D PBBN that follows most of the state-of-the-art approaches schemes. Experimental results show the better performance of the proposed approach compared to the state-of-the-art approaches.

1. Introduction

Eye blinking action is one of the significant vital signs that can indicate several human health issues such as red eye syndrome, fatigue, and drowsiness. Moreover, the signal of a sequence of eye blinks is extracted and employed in many applications such as disabled people communication, fake face detection, and face anti-spoofing. To appropriately detect an eye blink, several image and video processing techniques should be performed beforehand. The first phase of a traditional eye blinking scheme is to detect the face of the target person/s. There are a lot of face detection approaches that can be found in the literature where many of them (Chen, Huang, Peng, Zhou, & Zhang, 2020; Jiang & Learned-Miller, 2017; Kollreider, Fronthaler, Faraj, & Bigun, 2007; Li, Tang, Wu, Liu, & He, 2019; Viola & Jones, 2001) show good and robust performances in the presence of several challenges.

After the face being detected, the goal is to correctly detect the eyes for further processing. Researchers have proposed many techniques that detect the two eyes directly or indirectly by detecting their pupils and gazes. Such techniques can be divided into two classes: Infra-red based techniques and appearance-based techniques. The former category involves the techniques that make use of cameras equipped with infra-red sensors to obtain several eye location candidates based on their corneal reflections. Despite their good performance in providing accurate eye locations, the requirement of additional hardware remains an obvious downside that needs to be tackled. On the other hand, appearance-based techniques provide more practical frameworks that can be easily implemented in various real-world applications. Likewise, such a category of techniques can be divided based on their way of processing into two main subcategories: feature-based techniques and model-based techniques. The first subcategory consists of techniques that take advantage of the eye symmetry concept when measuring numerous detected local image features such as corner, edge, and gradient. Such a subcategory does not require any learning beforehand which makes its techniques more reliable to deal with untrained scenarios. However, such techniques are sensitive to noise and highly dependent

[*] Corresponding author at: University of the Basque Country UPV/EHU, San Sebastian, Spain.

E-mail addresses: sbekhouche001@ikasle.ehu.eus (S.E. Bekhouche), ibrahim.kajo@utbm.fr (I. Kajo), yassine.ruichek@utbm.fr (Y. Ruichek), fadi.dornaika@ehu.eus (F. Dornaika).

https://doi.org/10.1016/j.neunet.2022.04.010

on the accuracy of feature detection where falsely detected non-eye features lead to less stable performance. On the contrary, model-based techniques employ the global appearance of the eye or face images. Several machine and deep learning networks have been proposed to extract the accurate location of the eye region in addition to other facial features. These networks are usually trained on raw eye images or set of facial features and provide accurate and robust eye detection in most cases.

The last phase of an eye blink detection framework is eye state estimation where the state of the eye is identified whether it is closed or open. Dozens of eye state estimation techniques are proposed in the literature and they can be classified into three categories: template matching based, shape-based, and learning based techniques. Template-based techniques compare the detected eye images with templates that represent both eye states and the similarities among these images are measured to estimate the final eye state. On the other hand, shape-based techniques make use of different geometric characteristics of several shape features such as circular shape, curvature, and projection of pixel intensities along with both horizontal and vertical directions. The last category represents the techniques that use both machine and deep learning approaches to verify the state of detected eye images where their networks are basically trained on eye state sequences of (close–open–close) images. Despite their good and robust performances, the majority of these approaches are not designed to handle a sequence of eye blinks in the same video. An effective eye blink detection framework should be able to handle multiple blinks in one given video in addition to other challenges such as appearance changes and illumination variations.

In this paper, we aim to use a vision-based framework for automatic eye blinking detection where we propose two different approaches that improve performance in challenging scenarios. The first approach starts with constructing a feature-based matrix that contains temporal changes of the eyes, then uses SVD to extract the eye signal for eye blink detection. Finally, the detected blinks are verified using a 2D pyramidal bottleneck block network (PBBN). The second approach uses an end-to-end 3D PBBN to decide whether there is a blink in a specific image sequence or not. The main contributions of this work can be summarized in the followings:

- Introducing 2D and 3D light CNNs called Pyramidal Bottleneck Block Networks (PBBN) that contain Pyramid Bottleneck blocks.
- Proposing moving windowed-singular value decomposition (SVD) for eye blinks detection
- Proposing an end-to-end 3D PBBN to determine the existence of blink within an image sequence

The remaining of the paper is organized as follows: Section 2 presents related work on eye blinking. Then we introduce and describe the proposed approaches in Section 3. Section 4 presents the experiments and discusses the obtained results. Finally, Section 5 draws some conclusions and points for future directions.

2. Related work

Dozens of eye blink detection techniques have been proposed in the literature. These techniques can be classified into different classes according to their input data, way of processing, and the features used. There have been a variety of methods that proposed to detect eye blinks by estimating the eye states (open/close) using a single image only. Such estimation mainly starts by extracting different sets of features such as histogram of oriented gradients and local binary patterns which are fed into different machine learning techniques to learn the difference between both eye states. Recently, researchers started using different convolutional neural network architectures to enhance the accuracy of the state estimation results.

Zhao, Wang, Zhang, Qi, and Wang (2018) proposed a framework based on deep learning for eye blink classification composed of two deep networks. First, they detect the face from a frame using Viola–Jones, then the eye regions are cropped using facial landmarks provided by a deformable face alignment system. The image is fed to their first network which consists of convolution layers, pooling layers and fully connected layers. The last fully connected layer is followed by a Softmax function. In the second network, a flatten vector representing the image's pixels is fed to multiple fully connected layers followed by a Softmax function. In the end, they obtain three outputs: (i) fusion of Softmax functions from the two networks, (ii) the output of the first network and (iii) the output of the second network. These three outputs are passed to cross-entropy loss function which calculate the model error to train the whole system. This framework could run in real-time, however it is vulnerable to outdoor scenarios.

Li, Chang, and Lyu (2018) introduced a method of human eye blinking detection to expose the fake faces in videos generated by deep networks. They detected the face using Dlib library (King, 2009), then extracted the facial landmarks via Kazemi algorithm (Kazemi & Sullivan, 2014). These landmarks were used to align the face and crop the eye region. The cropped eye region of each frame was fed to their proposed Long-term Recurrent Convolutional Networks (LRCN) which can memorize the dynamic information from the input sequence. Generally, LRCN (Donahue et al., 2015) is composed of a visual feature extractor using CNN and sequence learning using a stack of recurrent neural networks (RNNs). They used the first fully connected layer of the VGG-16 model (Simonyan & Zisserman, 2014) to extract the features. Similar to Hu et al. (2020) and Li et al. (2018) proposed to use a RNN to capture temporal information of eye blinking in unconstrained scenarios. Instead of using deep features extracted by CNNs, they extracted the eye features using a lightweight uniform LBP descriptor (Ahonen, Hadid, & Pietikainen, 2006).

The second category of methods represents the techniques that process the whole video rather than a single image where the changes with respect to eye appearance features or eye motion are tracked and analyzed to construct a signal that represents the blinking events over time. Lalonde, Byrns, Gagnon, Teasdale, and Laurendeau (2007) introduced a multi-sensor approach that detects eye blinks in low contrast under near-infrared images. Initial eyes locations are calculated by finding the minimum of the large valleys in the extracted face profiles (row-wise projection). These initial eye locations are used to identify two eye regions of interest (ROI) in which SIFT feature points are extracted and tracked over time using Kalman filter to maintain the position alignment among successive frames. Then, motion detection followed by a thresholding procedure are performed in the tracked ROIs to identify the best eye blobs based on several geometry metrics such as area, position, angle, and ratios. Finally, the optical flows in the selected blob regions are computed to determine the dominant direction where vertical downward motion vectors are used to indicate the existence of an eye blink. This approach shows high detection rate and it performs in near real time. However, the usage of infrared illumination may cause harm to eyes especially at close distances.

Lee, Lee, and Park (2010) proposed a technique that detects both face and eye regions using Adaboost algorithm followed by illumination, binarization, and morphological operations. They introduced two features to detect the eye blink properly. The first feature is extracted by computing the height to width ratio

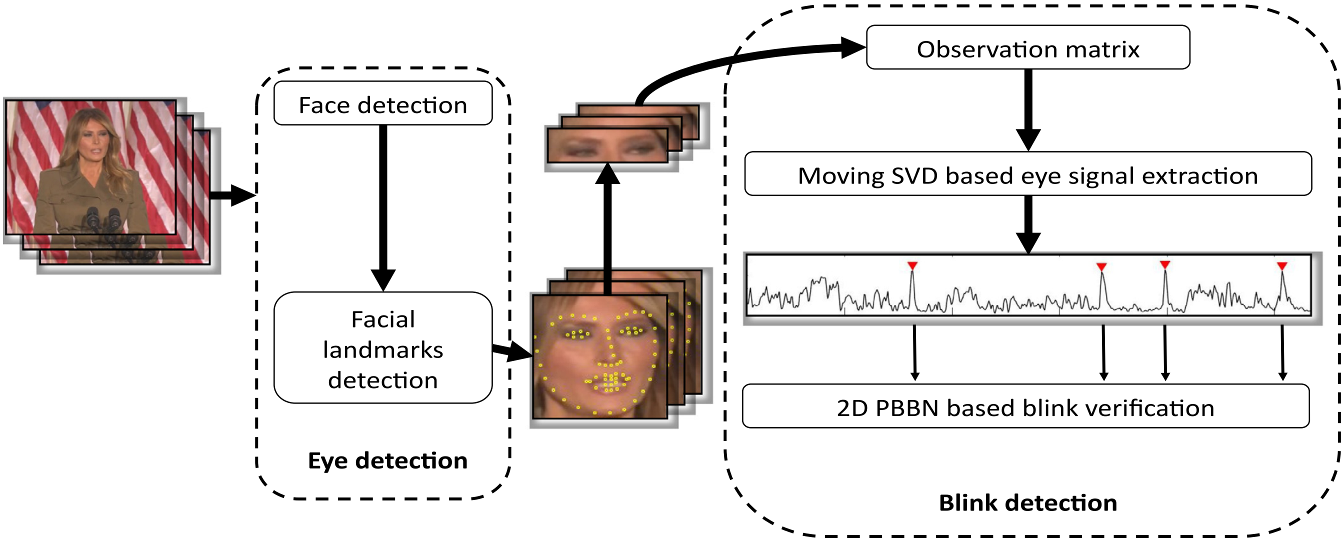

Face detection
Facial landmarks detection
Eye detection
Observation matrix
Moving SVD based eye signal extraction
2D PBBN based blink verification
Blink detection

Fig. 1. Eye system overview.

of each eye region. The second is obtained by computing the cumulative difference of the number of black pixels over time based on the assumption that this difference corresponds to the changes in the eye state. For better detection accuracy, the two extracted features are fed to a support vector machine (SVM) which is adaptively selected based on view angle of the target face. This approach shows robustness to different facial poses and different lighting conditions. On the downside, the eye blink detection misdetects many eye blinks due to the sensitivity of the proposed cumulative difference procedure to camera location, eye size, and initial eye mask.

Drutarovsky and Fogelton (2014) presented a motion based eye blink detection method that tracks the initial eye regions over time using a flock of KLT trackers. The tracked eye regions are split into 6 blocks where the dominant motion vector in each block is extracted by averaging the local vertical motion components located in the processed block. Then, a simple state machine is fed with the variance of the extracted average motion vectors to determine the eye state and detect the eye blink accordingly. Recently, the same authors enhanced their work by using Gunnar-Farneback tracker which provides less outliers than KLT tracker and better distribution of motion vectors. Subsequently, all vertical components of extracted motion vectors are normalized by the intraocular distance and averaged to construct a waveform that shows changes in its magnitudes while eye blinks. More recently, the same authors proposed another eye blink detection scheme that uses optical flow for the motion detection phase and LSTM for the eye state estimation phase. Their proposed approaches achieved high accuracy when they are tested on the existed datasets in addition to their proposed one. However, the tested datasets are limited to indoor videos and involves limited number of persons.

Chen, Wu, and Chien (2015) proposed a set of schemes for eye blink detection and gaze estimation without taking the advantage of infrared illumination. After the eyes being detected, several image preprocessing procedures are performed to eliminate the noise caused by the changes in normal-light conditions and reflections. To tackle the challenges presented while detecting eye parts under visible lighting conditions, they modified Starburst algorithm to make it more robust to such challenges. Using the adaptive Starburst extraction algorithm, their proposed technique correctly identifies both the iris and limbus features. Afterwards, the aspect ratio of the bounding box that contains the iris mask is computed over time where large values indicate eye-close states while small values indicate eye-open states.

Daza, Morales, Fierrez, and Tolosana (2020) proposed an eye blink classification approach using a modified VGG16 architecture. They also presented a dataset for eye blink classification under controlled conditions using three different sensors, namely 2 cameras (RGB and NIR) and electroencephalography (EEG) to detect the blink. Ryan et al. (2021) focused on blink detection using event cameras by proposing a fully convolutional gated recurrent YOLO network to detect eyes and then track them. Then, a fixed time window is used to analyze the presence/absence of eye blink.

3. Proposed approach

In this section, we aim to provide a detailed explanation of the proposed approach that tackles several challenges in the field of eye blinking detection. Our approach is divided into two main phases. The first one involves several preprocessing procedures such as face detection and eye detection while the second phase involves two processing stages: eye blinks detection via moving-windowed SVD and eye blink verification via 2D PBBN that verifies the existence of eye blink in each sub-sequence candidate extracted in the first stage. The general workflow of the proposed approach is shown in Fig. 1.

3.1. Face/facial landmark detection and eye region extraction

The first task of most facial analysis approaches is face detection. For this reason, the best face detection approach that is suitable to our eye blink detection approach is selected. The face detector we opted for is based on a Single Shot Detector (SSD) framework (Liu et al., 2016) using a ResNet model. After the face being detected, the eyes should be accurately localized to avoid the negative impact that false eye localization has on the eye blinking system. Therefore, our work is proposed to effectively tackle the eye localization challenges such as robustness in uncontrolled conditions, computation time and sensitivity to the illumination changes.

For an efficient eyes localization process, we propose to use Kazemi algorithm (Kazemi & Sullivan, 2014), which detects 68 facial points with specific coordinates that surround certain parts of the face including the eyes and nose (see 2) which can be

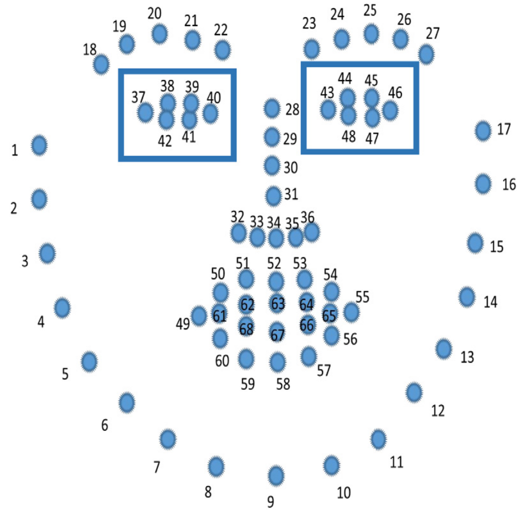


Fig. 2. Positions of 68 facial landmarks.

computed in about 1 millisecond. After detecting the facial landmarks in the input facial image as shown in Fig. 2, the face pose in the 2D image is rectified based on the eyes center similar to Bekhouche, Ouafi, Dornaika, Taleb-Ahmed, and Hadid (2017). Then, the landmarks from 37 to 42 and from 43 to 48 are used to crop the right-eye image and the left-eye image respectively. The method of cropping the eyes depends on padding the region that surrounds the landmarks of the intended eye by 25% in all directions. Finally, the cropped left and right eyes are resized to 96 × 96 pixels and placed according to their timestamp into two image sets representing both left and right eye sequences respectively.

3.2. Moving-windowed SVD

Suppose we have an image sequence $A = \{I_1, I_2, \dots, I_k\} \in \mathbb{R}^{m\times n\times k}$ that contains the cropped left/right eye images where m refers to the image height, n refers to the image width, and k indicates the number of images. Thus, a feature-based matrix that contains temporal changes of the pixels in eye regions is constructed as follows. After properly tracking and segmenting the eye regions, the segmented eye regions are divided into d blocks. Then, the pixel energy (sum of the square pixels' intensities) in each block is computed to construct a one-dimensional vector that contains the energy values of all blocks in a single cropped frame. Subsequently, the extracted vectors are employed to construct a $k \times d$ matrix $\mathbf{B} = \{e_1; e_2; \dots; e_k\}$, where each row e_t is an energy observation d-dimensional vector. To extract the eye change signal that best represents the eye blinking event, the singular value decomposition of matrix $\mathbf{B}$ is computed as follows:

$$\mathbf{U}^{\mathrm{T}}\mathbf{B}\mathbf{V} = \mathbf{S} = diag(s_1, \dots, s_p) \in \mathbb{R}^{k\times d} \tag{1}$$

where $p = min\{k, d\}$ and $s_1 \geq s_2 \geq \cdots \geq s_p \geq 0$. The matrices $\mathbf{U} \in \mathbb{R}^{k\times k}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ are the left and right singular vectors, respectively. Practically, a reduced-size SVD is utilized in this paper where the number of rows in the matrix $\mathbf{U} \in \mathbb{R}^{k\times k}$ is reduced to d where $\mathbf{U}$ become $\mathbf{U} \in \mathbb{R}^{k\times d}$. As discussed in Kajo, Kamel, Ruichek, and Malik (2018), the matrix $\mathbf{U}$ contains the same temporal information as the corresponding original matrix $\mathbf{B}$. Given this fact, the structures of the left singular vectors of matrix $\mathbf{U}$ should be further investigated. From a signal processing point of view, the projection of matrix $\mathbf{B}$ onto the first left singular vector u_1 subspace reveals the low-rank information embedded in $\mathbf{B}$. On the other hand, the projections of $\mathbf{B}$ onto the remaining left singular vectors' $\boldsymbol{u}_2, \boldsymbol{u}_3 \dots \boldsymbol{u}_d$ subspaces reveal the sparse information that represents the temporal changes in $\mathbf{B}$. Therefore, the vector that best represents the eye change signal is expected to be one of these vectors. The left singular vectors contain both negative and positive entries with values ranging between −1 and 1. For better representation and analysis of the estimated eye signal, the entries of the vectors of interest are scaled to fall on the interval of [0 1]. The scaled vectors are temporally processed using a moving average filter to reduce the outliers and remove the noise. The best vector that represents the eye change signal is determined based on its frequency information. To achieve this, a frequency estimation via fast Fourier transform is performed on each vector of interest and the vector having its principle frequency within a predefined interval and has the largest amplitude, is extracted. The entries that correspond to the frames when the eye is closed are expected to have large values while the entries that correspond to the frames when the eye is open are expected to have small values. Based on this fact, a coarse peak analysis is performed on the extracted vector-signal to obtain a set $C = \{A_1, A_2, \dots, A_l\} \in \mathbb{R}^{96\times 96\times k\times l}$ that contains l sub-sequence candidates of size 96 × 96 × k which are expected to show the eye blink events. Due to the fact that the verification stage is designed to deal with sequences with single potential eye blink, the obtained candidates are fed individually into the proposed 2D PBBN to verify the presence/absence of an eye blink. Fig. 3 illustrates an example of eye change signal extraction from a given video.

To achieve the real-time requirements, a moving window mechanism is used where the initial SVD components are extracted using the first k frames and the first frame in the sequence is removed when a new frame is arrived. The eye change signal is updated every second and the new added part is analyzed to detect new eye blink/s.

3.3. Pyramidal convolution neural network

Deep learning algorithms have significantly improved the performance in many computer vision tasks where deep learning models can learn more robust features compared to classic methods. Starting with LeNet (LeCun et al., 1995) then Alexnet (Krizhevsky, Sutskever, & Hinton, 2012), more generalized deep architectures have been merged like VGG (Simonyan & Zisserman, 2014), ResNet (He, Zhang, Ren, & Sun, 2016) and Inception (Szegedy et al., 2015). Inspired by residual block and bottleneck residual block (He et al., 2016), we propose a simple block named Pyramid Bottleneck which can be applied to both 2D and 3D inputs. The idea behind the Pyramid Bottleneck (PB) block is to reduce the total number of blocks in an architecture which leads to reduce the number of the parameters. The importance of a smaller number of parameters is mainly to shorten the inference time of the model and to fit the size of the model to the size of the training set, because the eye blink dataset does not contain many samples to train a model with a large number of parameters.

The proposed PBBN is composed of a starting block that contains a convolutional layer that filters the 96 × 96 × 3 input image with 64 kernels of size 3 × 3 without stride, a batch normalization layer which normalizes each input channel, ReLU layer which performs a threshold operation for the negative values to be set as 0 and a max-pooling layer of 3 × 3 with a stride of 1 × 1. Then, multiple PB blocks started with the convolution of 64 kernels and it doubles the kernels after each PB block. Finally, the networks end with a global average pooling which is connected with a fully connected layer that has the size of the number of classes or number of labels of the intended task.

The PB block is a bunch of branches shaped like a pyramid so that each branch contains multiple layers, the number of

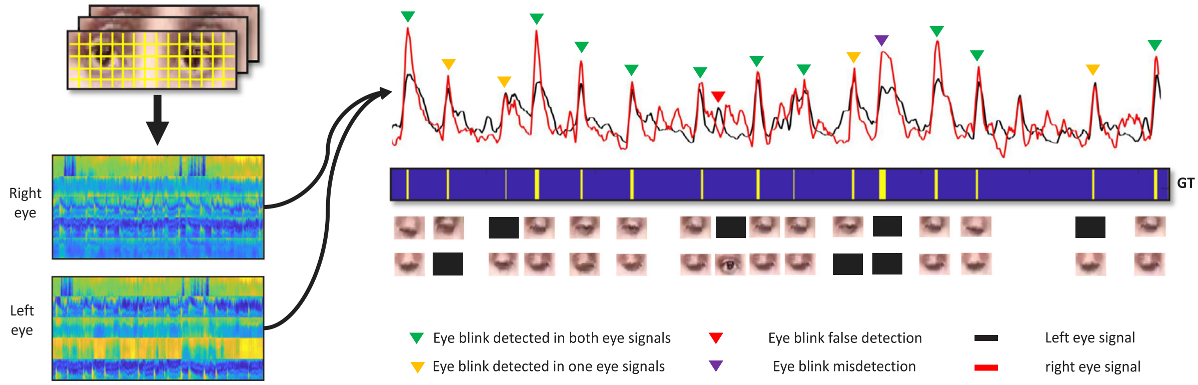

Right eye
Left eye
GT
▼ Eye blink detected in both eye signals
▼ Eye blink detected in one eye signals
▼ Eye blink false detection
▼ Eye blink misdetection
Left eye signal
right eye signal

Fig. 3. Eye signal extraction from image sequence of eyes.

Table 1
Architecture of an example of 3D PBBN that contains one pyramid with two branches.

Block		Layer	Filters number	Filter size	Stride size	Output
Input		3D Conv	64	3 × 3 × 3 × 3	1 × 1 × 2	96 × 96 × 7 × 64
		BN	–	–	–	96 × 96 × 7 × 64
		ReLU	–	–	–	96 × 96 × 7 × 64
		MaxPool	1	3 × 3 × 3	1 × 1 × 2	96 × 96 × 4 × 64
P1	B1	3D Conv	64	1 × 1 × 3 × 64	2 × 2 × 1	48 × 48 × 4 × 64
		BN	–	–	–	48 × 48 × 4 × 64
	B2	3D Conv	64	3 × 3 × 3 × 64	1 × 1 × 1	96 × 96 × 4 × 64
		BN	–	–	–	96 × 96 × 4 × 64
		ReLU	–	–	–	96 × 96 × 4 × 64
		3D Conv	64	1 × 1 × 3 × 64	2 × 2 × 1	48 × 48 × 4 × 64
		BN	–	–	–	48 × 48 × 4 × 64
	Add	ADD	–	–	–	48 × 48 × 4 × 64
		ReLU	–	–	–	48 × 48 × 4 × 64
Output		AvgPool	–	–	–	1 × 1 × 1 × 64
		FC	–	–	–	2

layers changes according to the number of PB. Let say we have l branches, the first branch has one convolution layer, the second branch has two convolution layers, and so on. Each convolution layer is followed by a batch normalization layer and ReLU layer except the last convolution layer of each branch where it is followed only by batch normalization. The last convolutions have filters of size 1 × 1 with a stride of 2 × 2. After each PB block, the channels dimension increases by double, and the spatial dimensions (i.e., $h \times w$) are reduced to half, and each PB branch starts with convolution layers that have a filter size of $2l - 1 \times 2l - 1$, and it keeps reducing the filter size by 2 of the next convolution of the same branch where l is the number of branches. The PB block could be explained mathematically as a given input $x \in \mathbb{R}^{c \times h \times w}$, where c is the number of channels; h, w are the height and width, respectively. The new feature map F' is computed as:

$$F'(x) = \sum_{b=1}^{l} F_l(x) \tag{2}$$

where l is the number of the branches inside the Pyramid Bottleneck and F_l is series of l convolutions. The outputs of branches are added element-wise together, hence the convolution layers have zero-padding except the last convolution layer of each branch, Fig. 4 illustrates an example of PB block with 4 branches.

Regarding the 3D Pyramid Bottleneck Block, it has the same characteristics as 2D one except for the input that contains depth information and they also differs in the spatial size of the filters where their size in 2D is $l \times l$ as for 3D it is $l \times l \times 3$, this is due to the fact that the depth has small dimension compared to the spatial information. Fig. 5 illustrates an example of 3D PB block with 3 branches. Similar to 2D PBBN, 3D PBBN is composed of an opening block that contains a 3D convolutional layer that filters the 96 × 96 × 13 × 3 input sequence with 64 kernels of size 3 × 3 × 3 with stride 1 × 1 × 2 for downsampling the temporal dimension where 13 is the number of frames, a batch normalization layer, ReLU layer and a 3D max-pooling layer of 3 × 3 × 3 that also downsamples the temporal dimension to half and maintains the spatial and channels dimensions. Then, come multiple 3D PB blocks like 2D CNN that down-sample the spatial dimension by half and double channels dimension after each 3D Block. The 3D PBBN ends with global average pooling, a fully connected layer that has two outputs and a softmax layer (case of eye blinking classification). Table 1 illustrates the architecture of an example of a 3D PBBN that contains one pyramid with two branches. As the latter table shows, the downsampling of the temporal dimension is done only in the first 3D convolution layer and the max-pooling layer. On the other hand, the downsampling of the spatial dimension is done after each PB block where the output will be downsized to half. Finally, the global average pooling layer will downsample both spatial and temporal dimensions to generate one feature map that is connected to a fully connected layer with an output corresponding to the specific task of the model.

4. Experiments

We consider two baseline tasks to evaluate the proposed work. The first task relates to eye blink classification where the objective is to determine the presence or absence of an eye blinking in a short sequence of images that contain only single eye blinks. As for the second task, named eye blink detection, the objective is to determine the time and duration of the detected eye blink.

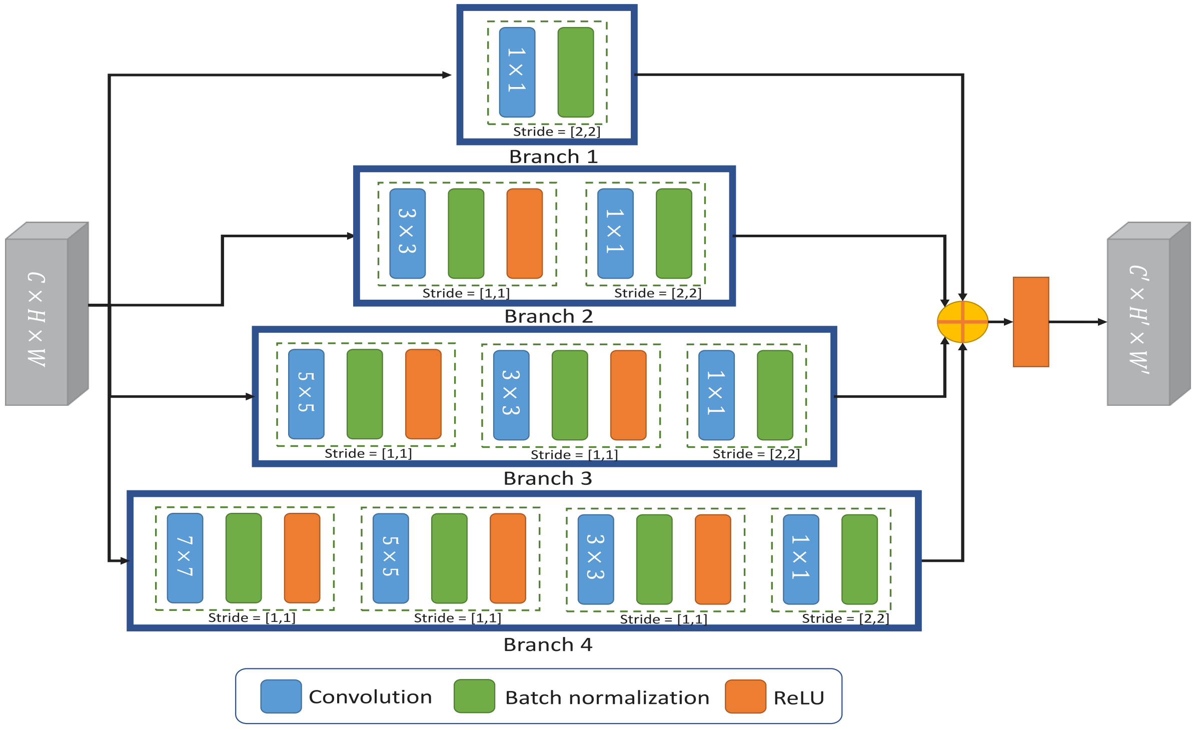


Fig. 4. Example of 2D Pyramid Bottleneck Block with 4 branches.

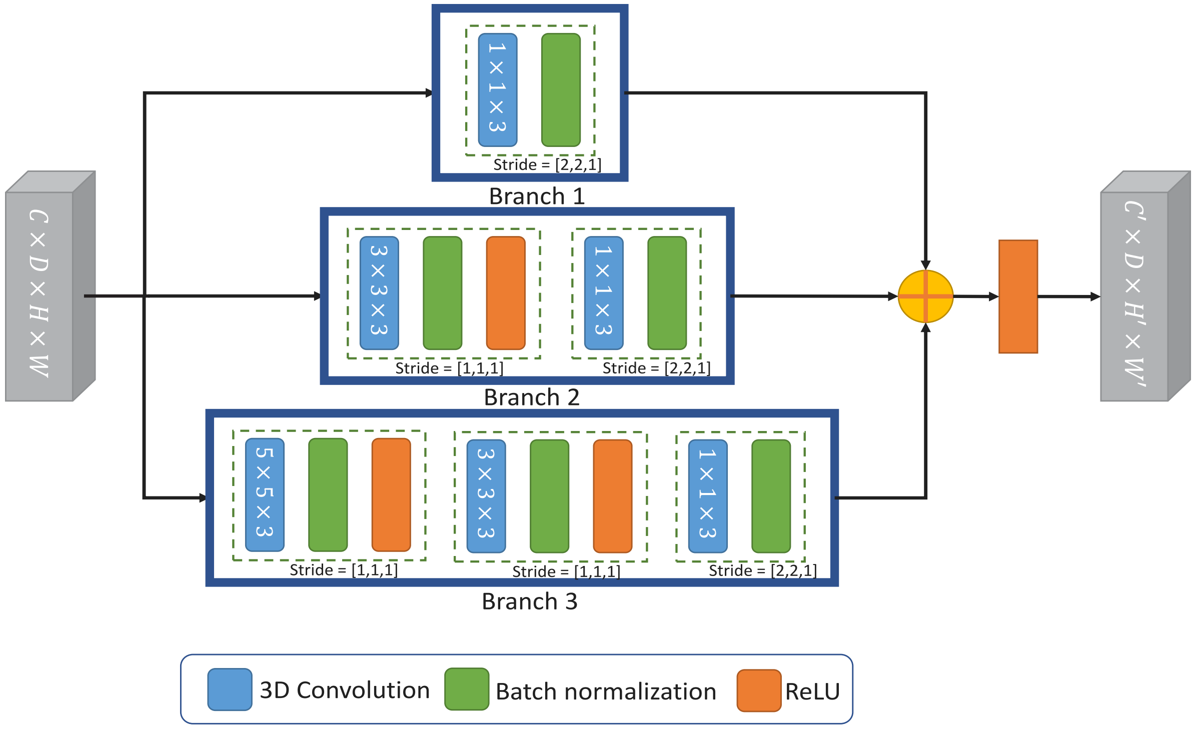


Fig. 5. Example of 3D Pyramid Bottleneck Block with 3 branches.

4.1. Eye blinking classification

4.1.1. Datasets

In this work, we focused on real scenarios of eye blinking classification, therefore we chose HUST-LEBW dataset as a suitable dataset for eye blinking classification in the wild. This dataset was created using clips from 20 movies and TV series such as The Matrix, A Chinese Fairy Tale and Game of Thrones. These clips were split into a training set and testing set and each clip is divided into multiple sub-clips videos, so the total number of videos reaches 90. Each video is either with a resolution of 1280 × 720 or 1456 × 600 and the actors in the videos appear in different poses and under different viewpoints. Through all the videos, 1314 samples were extracted and each sample is annotated with either a presence or absence of eye blink. The

Table 2
Distribution of sequences in HUST-LEBW dataset.

Eye	Blinking	Train	Test
Right	Yes	256	126
	No	190	98
Left	Yes	243	122
	No	181	98

Table 3
Performance of the different variants of the proposed 3d PBBN in HUST-LEBW dataset.

Network	Parameters	F_1	Recall	Precision
3D PBBN P2B2	**437 184**	0.8463	0.8548	0.8379
3D PBBN P2B3	1 974 912	0.8640	0.8710	0.8571
3D PBBN P2B4	5 933 952	0.8265	0.8548	0.8000
3D PBBN P3B2	1 286 784	0.8509	0.8629	0.8392
3D PBBN P3B3	5 775 360	**0.9105**	**0.8871**	**0.8730**
3D PBBN P3B4	17 220 096	0.8245	0.8145	0.8245

details are shown in Table 2. Each sample has a time span of 13 frames.

4.1.2. Evaluation

To address the problem of eye blinking classification, we performed some experiments using different combinations of the proposed 3D PBBN on HUST-LEBW dataset. As a classification problem, the evaluation of the performance of these experiments is done using *Recall*, *Precision* and F_1 metrics which are computed as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{5}$$

Unlike the proposed 2D PBBN, the proposed 3D PBBN is applied on the aggregate successive frames to learn the spatiotemporal information as we mentioned in Section 3.3. For a fair comparison with other works, we have used a span time of 13 frames as depth for the input video sequence that is fed to the proposed 3D PBBN. The training of the network is similar to the training of 2D PBBN, however, we reduced the mini-batch to 16 samples. Also, the loss function differs, we used cross-entropy function with mutually exclusive classes (blink/no-blink). We have chosen to evaluate the performance of different combinations of the proposed 3D PBBN by changing the number of pyramids from 2 to 3 and the number of branches of each pyramid from 2 to 4, which gives rise 6 combinations. Table 3 illustrates the results of each 3D PBBN combination on the eye blinking classification problem.

To show the generalization ability and the stability of the proposed model, we conducted more experiments besides the latter one. Specifically, we conducted three groups of experiments. The results of these experiments are shown in Table 4.

In the first group, we try to reduce the randomness of the training of the deep network. To this end, training and testing were repeated five times with the same training/test split. The average and standard deviation of recall, precision and F_1 score were reported. In the second set of experiments, we aim to reduce the randomness introduced by the selection of the training set. For this purpose, we perform 5 different random splits (70% for the training and 30% for the tests) and report the corresponding average and standard deviation of the evaluation metrics. In the last set of experiments, we use the classical scheme of five fold cross-validation. As can be seen, the obtained standard deviations are relatively small for all types of runs, indicating that the solution proposed by our scheme is stable both in terms of the training process of the network and in terms of the selected training images or videos.

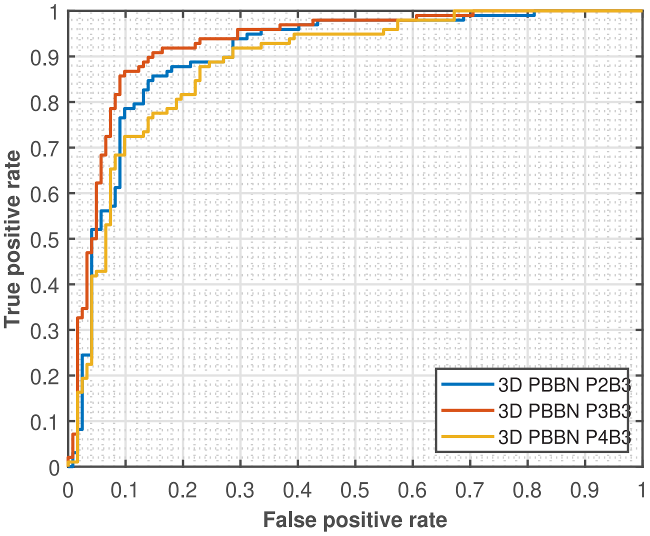


Fig. 6. ROC curves of the eye blinking classification results of the 3 branches 3D PBBN combinations.

From this table, we can observe that 3D PBBN P3B3 (3 pyramids with 3 branches in each pyramid) has the best results among the other variants. Also, we can notice that the best results are obtained from 3 branches pyramids (rows 2 and 5 of Table 7). Fig. 6 illustrates the ROC curves of the eye blinking classification results of the three 3D PBBN combinations, where it shows the promising potential performance of the three combinations of the proposed 3D PBBN with three branches. For a comprehensive evaluation, Table 5 provides comparison between the best combination (3D PBBN P3P3) and the state-of-the-art approaches where it reveals that the proposed approach is significantly better compared to the rest approaches in the recent benchmark.

4.2. Eye blinking detection

4.2.1. Datasets

The Epan-EyeBlink dataset[1] was collected from Youtube videos running at frame rates of 30fps. We collected and trimmed 18 videos with a variation of subjects, poses, no glasses/glasses, expressions and illumination. Unlike other publicly available datasets, the videos in the proposed dataset have multiple blinks in their sequences which allows studying the time and width of each blink. The average time of the videos is 26 s, and the average number of blinks is 15. Fig. 7 shows some samples from our Epan-EyeBlink dataset.

4.2.2. Evaluation

We use the same metrics as for eye blinking classification. However, the TP, FP, and FN are defined in a different way where TP means there is intersection in time dimension between the predicted blink and ground truth. FN and FP are the numbers of missed blinks and the number of false detected blinks.

In this section, we performed three evaluations for eye blinking detection. The first one is concerned with the proposed moving windowed-SVD approach, the second one about the proposed

[1] https://github.com/Bekhouche/Epan-EyeBlink.

Fig. 7. Samples from Epan-EyeBlink dataset.

Table 4
Performance of the proposed 3D PBBN using different training strategies in HUST-LEBW dataset.

Strategy	Eye idx	Recall	Precision	F1 score
5-repetition	Left Right	0.9049 $\mp$ 0.0093 0.8984 $\mp$ 0.0067	0.8805 $\mp$ 0.0086 0.8844 $\mp$ 0.0063	0.8925 $\mp$ 0.0135 0.8913 $\mp$ 0.0060
5-random split	Left Right	0.9032 $\mp$ 0.0103 0.8952 $\mp$ 0.0181	0.8770 $\mp$ 0.0191 0.8691 $\mp$ 0.0138	0.8898 $\mp$ 0.0116 0.8819 $\mp$ 0.0142
5-fold cross validation	Left Right	0.8968 $\mp$ 0.0177 0.8901 $\mp$ 0.0188	0.8642 $\mp$ 0.0199 0.8593 $\mp$ 0.0203	0.8801 $\mp$ 0.0135 0.8744 $\mp$ 0.0180

Table 5
Performance comparison among the different eyeblink verification methods on HUST-LEBW dataset.

Method	Eye idx	Recall	Precision	F1 score
Morris (ver.) (Morris, Blenkhorn, & Zaidi, 2002) (2002)	Left Right	0.5246 0.5635	0.4741 0.5064	0.4981 0.5334
Morris (hor.) (Morris et al., 2002) (2002)	Left Right	0.6393 0.5476	0.5342 0.5107	0.5821 0.5285
Morris (flow) (Morris et al., 2002) (2002)	Left Right	0.4918 0.4286	0.4918 0.4741	0.4918 0.4502
Chau (Chau & Betke, 2005) (2005)	Left Right	0.1721 0.2302	**1.0000** **0.9656**	0.2937 0.3718
Drutarovsky (Drutarovsky & Fogelton, 2014) (2014)	Left Right	0.1190 0.0952	0.4757 0.2860	0.1904 0.1428
Daza (Daza et al., 2020) (2020)	Left Right	**0.9603** 0.7950	0.6080 0.7348	0.7446 0.7637
Hu (Hu et al., 2020) (2020)	Left Right	0.7805 0.8333	0.7385 0.7778	0.7589 0.8046
Proposed (3D PBBN)	Left Right	0.9161 **0.9048**	0.8812 0.8507	**0.8983** **0.8769**

Table 6
Results of the proposed SVD approach on Epan-EyeBlink dataset.

Video	Recall	Precision	F_1
1	0.8205	0.8205	0.8205
2	1.0000	0.2405	0.3878
3	0.9048	0.4750	0.6230
4	0.9467	0.4863	0.6426
5	0.9200	0.4035	0.5610
6	1.0000	0.5778	0.7324
7	1.0000	0.0909	0.1667
8	0.8537	0.5512	0.6699
9	0.8148	0.4074	0.5432
10	1.0000	0.8636	0.9268
11	0.9245	0.6164	0.7396
12	1.0000	0.3483	0.5167
13	1.0000	0.1374	0.2416
14	1.0000	0.3226	0.4878
15	1.0000	0.6419	0.7819
16	0.7609	0.6604	0.7071
17	0.9635	0.6839	0.8000
18	1.0000	0.2815	0.4393
Average	0.9394	0.4783	0.5993

2D PBBN, and the last one evaluated the combination of the proposed moving windowed-SVD method and the proposed 2D PBBN. Herein, we evaluate the proposed moving windowed SVD on the Epan-EyeBlink dataset, the detailed results are presented in Table 6. The results show poor precision and high recall owing to the fact that the proposed moving windowed-SVD gives a lot of false detected blinks. On the other hand, it detects most of the blinks.

In the case of the combination-based approach, the objective of the 2D PBBN is to verify the existence or absence of eye blink to enhance the performance of the SVD approach by filtering most of the false detected blinks. Therefore, we first trained a light 2D PBBN (2 pyramids and each pyramid has 2 branches) using some images of the HUST-LEBW dataset. To make the database suitable for training the proposed light 2D PBBN, we took three images from each sequence and labels them similar to their sequence label, so that we have 2610 images for training and 1332 images for validation. The best-achieved result on the validation subset was 91.14% recall.

The results of the 2D PBBN on the Epan-EyeBlink dataset are given in Table 7. Then, the trained network is applied on

Table 7
Results of the proposed PBBN approach on Epan-EyeBlink dataset.

Video	Recall	Precision	F_1
1	0.8478	1.0000	0.9176
2	1.0000	1.0000	1.0000
3	0.9545	0.9545	0.9545
4	0.9074	0.9608	0.9333
5	0.8846	0.8846	0.8846
6	1.0000	0.9808	0.9903
7	1.0000	0.6364	0.7778
8	0.7917	0.9500	0.8636
9	0.9107	0.9623	0.9358
10	1.0000	1.0000	1.0000
11	0.9381	0.9464	0.9422
12	1.0000	0.8750	0.9333
13	1.0000	0.6667	0.8000
14	1.0000	0.9268	0.9620
15	1.0000	0.9517	0.9752
16	0.7794	1.0000	0.8760
17	0.9638	0.9568	0.9603
18	1.0000	0.9143	0.9552
Average	0.9432	0.9204	0.9257

Table 8
Results of the proposed SVD+2D PBBN approach on Epan-EyeBlink dataset.

Video	Recall	Precision	F_1
1	0.8478	1.0000	0.9176
2	1.0000	1.0000	1.0000
3	0.9512	0.9750	0.9630
4	0.9699	0.8815	0.9236
5	0.9636	0.9298	0.9464
6	1.0000	0.9333	0.9655
7	1.0000	0.8485	0.9180
8	0.9130	0.9921	0.9509
9	0.9138	0.9815	0.9464
10	1.0000	0.8636	0.9268
11	0.9490	0.9371	0.9430
12	1.0000	0.8764	0.9341
13	1.0000	0.8550	0.9218
14	1.0000	0.8629	0.9264
15	1.0000	0.9256	0.9614
16	0.8281	1.0000	0.9060
17	0.9733	0.9430	0.9579
18	1.0000	0.9556	0.9773
Average	0.9617	0.9312	0.9437

all the detected blink candidates obtained by the proposed SVD based method, and the results are shown in Table 8. Due to the high recall of the proposed 2D PBBN based verification phase, the thresholding parameters used in the peak analysis procedure applied on the extracted eye signals, are set to be as low as possible. Such step guarantees the detection of the majority existed eye blinks in an eye signal which is clearly indicated by the high precision values in Table 8. On the other hand, lowering the thresholding parameters increases the likelihood of false detections which is resulted in low recall values as reported in Table 8.

The performance comparison of the proposed approach with the recent state-of-the-art approaches on the proposed Epan-EyeBlink dataset is provided in Table 9. We can observe that our proposed SVD based method has a better recall than the other state-of-the-art approaches however its precision is very low due to the multiple false detections. On the other hand, the proposed PBBN has good precision and recall. Wherefore, the combination of SVD + PBBN improves both precision and recall and it has the best results among the other works.

5. Conclusion

In this paper, we proposed different supervised and unsupervised learning approaches to provide an effective and robust eye blink detection framework. First, we proposed an efficient 3D model to determine the exists of an eye blink in eye sequence images as this model contains a small number of parameters compared to known CNN models. Second, we incorporated the unsupervised learning using SVD which is effectively employed to extract the eye motion signal that contains unique patterns which represent the eye blinks. Then, the supervised learning based on the 2D PBBN which is utilized to verify the detected eye blink candidates and enhance the detection performance in terms of recall values. Such fusion of supervised and unsupervised learning approaches provides a robust eye blink detection framework that is capable of handling several challenges such as different lighting conditions, variety of appearance, and multi-blink sequences. Moreover, available datasets within this research field were limited to sequences with only one eye blink per sequence which in turn prevents the evaluation of the performance of proposed techniques in the long-term and in the presence of the challenges that accompany these sequences. Therefore, we introduced a new dataset that involves several videos with multiple eye blinks in each sequence in addition to different challenges. The experimental results indicate the effectiveness and outperformance of the proposed framework compared to state-of-the-art methods.

Table 9
Performance comparison among the different eyeblink detection methods on Epan-EyeBlink dataset.

Method	Recall	Precision	F_1
Li (Li et al., 2018) (2018)	0.8507	0.8153	0.8326
Maior (Maior, das Chagas Moura, Santana, & Lins, 2020) (2020)	0.8976	0.6120	0.7278
Hu (Hu et al., 2020) (2020)	0.8712	0.8636	0.8674
Proposed (SVD)	0.9394	0.4783	0.5993
Proposed (PBBN)	0.9432	0.9204	0.9257
Proposed (SVD-PBBN)	**0.9617**	**0.9312**	**0.9437**

As future work, we envision the use of temporal transformers networks and the improved combinations of CNN-LSTM for eye blinking and other related applications such as yawning and drowsiness detection. One limitation of the state-of-the-art eye blinking methods is that they require frontal face. Thus, we envision trying to tackle and investigate this problem by using non-frontal faces.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041.

Bekhouche, S., Ouafi, A., Dornaika, F., Taleb-Ahmed, A., & Hadid, A. (2017). Pyramid multi-level features for facial demographic estimation. *Expert Systems with Applications*, *80*, 297–310.

Chau, M., & Betke, M. (2005). *Real time eye tracking and blink detection with usb cameras*: *Technical report*, Boston University Computer Science Department.

Chen, W., Huang, H., Peng, S., Zhou, C., & Zhang, C. (2020). YOLO-Face: a real-time face detector. *The Visual Computer*, 1–9.

Chen, B.-C., Wu, P.-C., & Chien, S.-Y. (2015). Real-time eye localization, blink detection, and gaze estimation system without infrared illumination. In *2015 IEEE international conference on image processing (ICIP)* (pp. 715–719). IEEE.

Daza, R., Morales, A., Fierrez, J., & Tolosana, R. (2020). mEBAL: A multimodal database for eye blink detection and attention level estimation.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., et al. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2625–2634).

Drutarovsky, T., & Fogelton, A. (2014). Eye blink detection using variance of motion vectors. In *European conference on computer vision* (pp. 436–448). Springer.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hu, G., Xiao, Y., Cao, Z., Meng, L., Fang, Z., Zhou, J. T., et al. (2020). Towards real-time eyeblink detection in the wild: Dataset, theory and practices. *IEEE Transactions on Information Forensics and Security*, *15*, 2194–2208.

Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)* (pp. 650–657). IEEE.

Kajo, I., Kamel, N., Ruichek, Y., & Malik, A. S. (2018). SVD-Based tensor-completion technique for background initialization. *IEEE Transactions on Image Processing*, *27*(6), 3114–3126.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1867–1874).

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.

Kollreider, K., Fronthaler, H., Faraj, M. I., & Bigun, J. (2007). Real-time face detection and motion analysis with application in "liveness" assessment. *IEEE Transactions on Information Forensics and Security*, *2*(3), 548–558.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N., & Laurendeau, D. (2007). Real-time eye blink detection with GPU-based SIFT tracking. In *Fourth Canadian conference on computer and robot vision (CRV'07)* (pp. 481–487). IEEE.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., et al. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks: The Statistical Mechanics Perspective*, *261*, 276.

Lee, W. O., Lee, E. C., & Park, K. R. (2010). Blink detection robust to various facial poses. *Journal of Neuroscience Methods*, *193*(2), 356–372.

Li, Y., Chang, M.-C., & Lyu, S. (2018). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE international workshop on information forensics and security (WIFS)* (pp. 1–7). IEEE.

Li, Z., Tang, X., Wu, X., Liu, J., & He, R. (2019). Progressively refined face detection through semantics-enriched representation learning. *IEEE Transactions on Information Forensics and Security*, *15*, 1394–1406.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37). Springer.

Maior, C. B. S., das Chagas Moura, M. J., Santana, J. M. M., & Lins, I. D. (2020). Real-time classification for autonomous drowsiness detection using eye aspect ratio. *Expert Systems with Applications*, Article 113505.

Morris, T., Blenkhorn, P., & Zaidi, F. (2002). Blink detection for real-time eye tracking. *Journal of Network and Computer Applications*, *25*(2), 129–143.

Ryan, C., O'Sullivan, B., Elrasad, A., Cahill, A., Lemley, J., Kielty, P., et al. (2021). Real-time face & eye tracking and blink detection using event cameras. *Neural Networks*, *141*, 87–97.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, Vol. 1* (p. I). IEEE.

Zhao, L., Wang, Z., Zhang, G., Qi, Y., & Wang, X. (2018). Eye state recognition based on deep integrated neural network and transfer learning. *Multimedia Tools and Applications*, *77*(15), 19415–19438.