Eye-LRCN: a Long-term Recurrent Convolutional Network for Eye Blink Completeness Detection

Gonzalo de la Cruz, Madalena Lira, Oscar Luaces, Beatriz Remeseiro

***Abstract*—Computer vision syndrome causes vision problems and discomfort mainly due to dry eye. Several studies show that dry eye in computer users is caused by a reduction in the blink rate and an increase in the prevalence of incomplete blinks. In this context, this manuscript introduces Eye-LRCN, a new eye blink detection method that also evaluates the completeness of the blink. The method is based on a long-term recurrent convolutional network, which combines a convolutional neural network (CNN) for feature extraction with a bidirectional recurrent neural network that performs sequence learning and classifies the blinks. A siamese architecture is used during CNN training to overcome the high-class imbalance present in blink detection and the limited amount of data available to train blink detection models. The method was evaluated on three different tasks: blink detection, blink completeness detection, and eye state detection. We report superior performance to the state-of-the-art methods in blink detection and blink completeness detection, and remarkable results in eye state detection.**

***Index Terms*—Computer vision syndrome, blink completeness detection, eye state detection, long-term recurrent convolutional networks, siamese neural networks.**

I. Introduction

COMPUTER vision syndrome (CVS) [1] is a temporary condition that causes eye and vision problems [2] by focusing the eyes on a computer screen for long, uninterrupted periods of time. Some of its symptoms are blurred vision, double vision, tired eyes, irritation, and redness. The main contributor to CVS is dry eye, caused by a reduced eye blink rate (EBR) and an increased prevalence of incomplete blinks when being exposed to screens for long periods of time [3].

Eye blinking is essential to keep the ocular surface healthy and hydrated. It keeps a stable tear film over the anterior ocular surface, cleaning it when it comes in contact with dust and dirt, and preventing the cornea from dryness. Blinking is a protective mechanism for the eye and it is vital for corneal health and optical performance [4]. In fact, the best way to avoid dry eye symptoms is to blink regularly [5]. The average EBR ranges from 10 to 22.4 blinks per minute. A lower EBR contributes to decrease the quality of the eye tear film and stresses the cornea, resulting in dry eye symptoms and sometimes inflammation.

Looking at screens during long periods of time may reduce the EBR by 60% [1]. Cardona et al. [6] studied the influence of the level of dynamism of different screen-related tasks on the EBR, blink amplitude, and tear film integrity. Their results show that the EBR can decrease to 1/3 or 1/2 of baseline levels depending on the dynamism and the cognitive demands of the task [3, 7]. A positive correlation between the percentage of incomplete blinks and the dynamism of the task is also reported.

Regarding the completeness of a blink, it is defined by whether the two eyelids touch or not during the blink. Portello et al. [8] studied the influence of blink completeness in CVS, concluding that the impact of a high prevalence of incomplete blinks may be as significant as a low EBR. Incomplete blinks are one of the causes of Meibomian gland dysfunction [9], which results in a disruption and instability of the tear film causing dry eye even with normal aqueous tear production. Meibomian gland dysfunction is now recognized to be the most common cause of evaporative dry eye [10]. Therefore, complete and incomplete blink classification is of utmost importance to identify the causes and worsening of dry eye.

This manuscript presents Eye-LRCN, a new eye blink detection method that also takes into account blink completeness. Our approach is based on a long-term recurrent convolutional network (LRCN) [11] that uses a siamese architecture to overcome the high imbalance present in blink detection problems. Balanced mini-batches were used to train the siamese network, which have proven to be very effective on unbalanced problems [12]. The siamese network is combined with a bidirectional long short-term memory (LSTM) network [13] that performs sequence learning based on the temporal context of past and future inputs. It has been proven that bidirectional networks are substantially better than their unidirectional counterparts in many fields such as speech recognition [14] or traffic prediction [15] but, to the best of our knowledge, their impact on blink detection problems has not yet been studied.

The rest of this manuscript is structured as follows. Section II provides a brief review of the state-of-the-art in eye blink detection. Section III presents Eye-LRCN. Section IV defines the experimental framework to evaluate the proposed method in three different tasks: blink detection, blink completeness detection, and eye state detection. Section V reports the results achieved by the proposed method and establishes a comparison with those obtained by some representative approaches presented in the review of the state-of-the-art. Finally, Section VI closes the manuscript with the final conclusions obtained from this research.

This work was supported in part by the Portuguese Foundation for Science and Technology (FCT) in the framework of the Strategic Funding UIDB/04650/2020. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan XP GPU used in this research.

G. de la Cruz, O. Luaces and B. Remeseiro are with the Artificial Intelligence Center, Universidad de Oviedo, 33204 Gijón, Spain, e-mails: UO244583@uniovi.es, oluaces@uniovi.es, bremeseiro@uniovi.es.

M. Lira is with the Physics Center of Minho and Porto Universities – CF-UM-UP, School of Sciences, University of Minho, 4710-057 Braga, Portugal, email: mlira@fisica.uminho.pt.

II. Related work

A blink is a rapid action in which the eyelids progressively close and reopen again. It is an action over time that takes around 572 ± 25 ms [16] during which the eye is fully closed for about 50 ms [17]. An ordinary webcam recording at 30 frames per second (fps) is enough to capture the fully-closed eye and, therefore, to differentiate between complete and incomplete blinks. Visual blink detection methods use either consumer-level webcams [18, 19, 20, 21] or more specialized sensors like eye trackers [22, 23] to acquire their input data. Blink detection methods usually separate the task in two parts: 1) face and eye detection, and 2) blink detection. For face and eye detection there are powerful and efficient state-of-the-art methods such as the Viola-Jones algorithm [24]. For this reason, most of blink detection models delegate face and eye detection to these methods and focus only on the blink detection procedure itself.

There is not a clear definition of what a blink is in the state-of-the-art. Some works use the term *blink* to refer to a single eye picture in which the eye is fully closed, while others use it to refer to a sequence of frames in which the eye closes and reopens again. In this context, it is important to differentiate between *blink detection* and *eye state detection*. In blink detection, a blink is defined as a sequence of frames in which the eyelids close and reopen again, although in some cases the eye does not close completely. In eye state detection, individual eye images are classified as open or closed eyes. It can be defined either as a binary problem or as a regression problem in which the percentage of eye closure is calculated. Note that blink detection methods can be based on eye state detection of individual frames.

Next, we include a brief review of some representative blink detection and eye state detection methods found in the literature. At the end of the section, we present the rationale of our approach compared to these methods.

A. Blink detection methods

These methods commonly use videos as input data, and are grouped into those that analyze individual frames and those that use sequences of frames. Following the first approach, Soukupová and Cech [25] proposed a real-time algorithm that uses a support vector machine (SVM) to detect blinks using the eye aspect ratios in a short temporal window or a hidden Markov model followed by a state machine to recognize blinks using the eye closure lengths.

Concerning works that analyze sequences of frames, Fogelton and Benesova [19] used a tracker to obtain the motion vectors in the eye region. Blink detection is performed by means of a state machine fed with the average motion vectors, normalized with standard deviation and time constraint to achieve invariance of the eye region size. They also introduced Researcher's Night, a large real-world dataset with more than 1800 annotated eye blinks. Later, the same authors presented the first eye blink detection method capable of evaluating blink completeness [26]. They extracted motion vectors from the eye region, which were then fed to a unidirectional recurrent neural network (RNN).

Hu et al. [27] introduced the HUST-LEBW dataset, the first eye-blink in the wild dataset that involves spatial-temporal sequence information. The authors formulated eye-blink detection as a binary spatial-temporal pattern recognition problem. They used kernelized correlation filters for eye tracking and a modified LSTM model to predict eye blinks. A comparative study on the HUST-LEBW dataset demonstrates the suitability of their approach for eye blink detection in the wild, showing superior performance than other evaluated methods.

Lamba et al. [28] proposed an eye blink detection method using feature level fusion. They introduced the eye-eyebrow facet ratio, which is formed by fusing the eye facet ratio and the eyebrow-to-nose facet ratio. Their method outperformed other eye blink recognition systems in the ZJU dataset [29].

Different computer vision applications can be addressed by solving the blink detection problem. For example, Han et al. [30] presented a driver drowsiness detection method based on eyelid movement, whilst Jordan et al. [31] proposed solving the same task through a CNN-based system embedded in connected glasses. Other interesting applications include fatigue recognition [32] and deep fake videos detection [33].

B. Eye state detection methods

The most recent eye state detection methods found in the literature can be mainly grouped into two categories: those that compute a feature vector from input images and then classify it into open/closed using classical machine learning methods, and those that solve the task by means of CNNs.

With respect to the former group, Song et al. [34] presented the Closed Eyes in the Wild dataset and proposed a method that combines features from a new descriptor based on histograms of oriented gradients, local ternary patterns, and Gabor wavelets. Next, they used these features to train different classifiers and evaluated them on still images from two datasets, ZJU and Closed Eyes in the Wild, achieving the best results with the SVM classifier. For their part, Remeseiro et al. [18] analyzed the low-level features of the eye region using uniform histograms and discrete wavelets, which were used to feed different classifiers. The proposed method was evaluated on their own dataset, obtaining the best results with a multilayer perceptron (MLP). More recently, Eddine et al. [20] presented EyeLSD, a new framework to localize the eyes and identify their states without the face detection step. In particular, they proposed two novel descriptors based on local binary patterns, which were used to train SVM and MLP classifiers. Their approaches were evaluated on the ZJU dataset, achieving a lower performance than Song et al. [34] but improving computational efficiency.

Regarding the use of CNNs, Anas et al. [35] proposed two CNN architectures based on the well-known LeNet [36]: one to address binary eye state detection (closed and open eyes) and the other for three-class eye state detection (closed, open, and partially open eyes). A more up-to-date CNN architecture was considered in [37]. In particular, the authors used a pre-trained ResNet50 [38] and fine-tuned it using a mix of the ZJU dataset and their own dataset. Finally, Cortacero et al. [21] presented a new open-source dataset for eye state detection

(RT-BENE) and proposed a set of baseline CNNs using standard backbone architectures. They also proposed a method that uses mask R-CNN [39] to perform semantic labeling of the eye region. The CNN-based models outperformed the R-CNN method, which was trained with few samples due to the time-consuming annotation process.

C. Rationale of the approach

Two main problems arise when training deep neural networks (DNNs) for blink detection. First, the available datasets for blink detection are relatively small compared to the datasets used for training deep learning models for other tasks such as object recognition [40] or image segmentation [41]. Cortacero et al. [21] analyzed the impact of dataset size in training DNNs for blink detection, showing that increasing the dataset size improves the performance of the trained models.

The second problem is that blink detection is a highly unbalanced problem. Blinks are fast actions that last less than half a second and, therefore, very few frames can be annotated as blinks during video recording. In the Researcher's Night dataset [19] about 5% of the frames are considered part of a blink and about 1.3% of the visible eye pictures are considered to be fully closed eyes; whilst the percentage of closed eyes pictures in the RT-BENE dataset [21] is around 4.3%.

Some of the blink detection methods based on DNNs use different techniques to reduce the effect of these problems. To mention a few, some research works use data augmentation to prevent the overfitting caused by training models with a small number of samples [35, 37], others use transfer learning to reduce the impact of the lack of data [37], and others applied oversampling techniques to reduce the class imbalance between open and closed eyes [35, 21].

However, to the best of our knowledge, there is no previous research focused on finding a solution to both problems. We present Eye-LRCN, a novel approach to blink detection and blink completeness detection that has been designed with these two issues in mind. Our approach is based on the one proposed by Li et al. [33], which also uses a LRCN for blink detection. The main novelties with respect to their approach is the use of a siamese architecture for CNN training and the use of a bidirectional LSTM [13] instead of a unidirectional LSTM. Note that siamese architectures have proven effective for other problems with high-class imbalance [42, 43, 44], being also a popular solution for one-shot and few-shot learning problems [45]. Bidirectional LSTM have proven to be considerably better than their unidirectional counterparts in many fields such as speech recognition [14] or traffic prediction [15]. We combine the siamese architecture with data augmentation and transfer learning, making our approach robust to class imbalance and having a relatively small number of training samples. Furthermore, our approach, which is mainly intended for blink detection and blink completeness detection, can also be used for eye state detection.

III. Methodology

This section presents Eye-LRCN, a long-term recurrent convolutional network to solve the problem of eye blink detection. Broadly speaking, LRCNs combine CNNs and RNNs to get the best of both architectures. On the one hand, CNNs provide very good performance when working with image data and are excellent feature extractors, but they are not designed to deal with sequential data. On the other hand, RNNs are excellent for working with sequential data, but they are not as good as CNNs when dealing with images. LRCNs can map variable length inputs (e.g., video frames) to variable length outputs (e.g., blink predictions), leveraging the performance of CNNs for visual recognition problems [11]. Figure 1 depicts an overview of the Eye-LRCN method, which receives a video as input and performs image sequence analysis through a three-step process summarized as follows:

1) The eye images are passed through a CNN that acts as a feature extractor. The CNN is trained to discriminate between images of open, closed, and partially closed eyes. The position of the extracted features in the feature space determines the degree of openness of the eye.
2) The extracted features serve as input to a bidirectional LSTM [13] that performs sequence learning taking into account the temporal context of the input data.
3) A fully connected (FC) layer with a softmax activation function is in charge of determining which class each eye image belongs to. The number of units in this layer varies depending on the task the network is trained for.

It is worth noting that this methodology was developed for blink detection and blink completeness detection. However, given the similarity of these problems with eye state detection, our proposal is also suitable for this other task.

A. Feature Extraction

The goal of this step is to train a network capable of discriminating between images of open, closed, and partially closed eyes. For this purpose, the network should capture some relevant properties of eye images in order to generate a feature space where the samples of each class are together and separated from the samples of the other classes. A siamese architecture is used to overcome the high-class imbalance present in blink detection datasets and the small number of samples available for training.

Siamese neural networks are composed of twin networks that receive different inputs, which are joined by an energy function at the top of the network [46]. This function computes some metric between the highest-level feature representation on each side of the network. Siamese networks ensure consistency in their predictions, guaranteeing that similar samples are mapped nearby in the feature space and distinct samples are mapped distantly. In practice, twin networks are represented by a single network through which both inputs are processed, ensuring symmetry and parameter sharing.

As stated before, we use a siamese CNN trained to discriminate between images of open, closed, and partially closed eyes. The siamese architecture is based on the one proposed by Koch et al. [46] for few-shot learning image recognition. The main difference with their approach is the learning task, in addition to the siamese design. Figure 2 shows the architecture used during training. A CNN feature extractor receives an

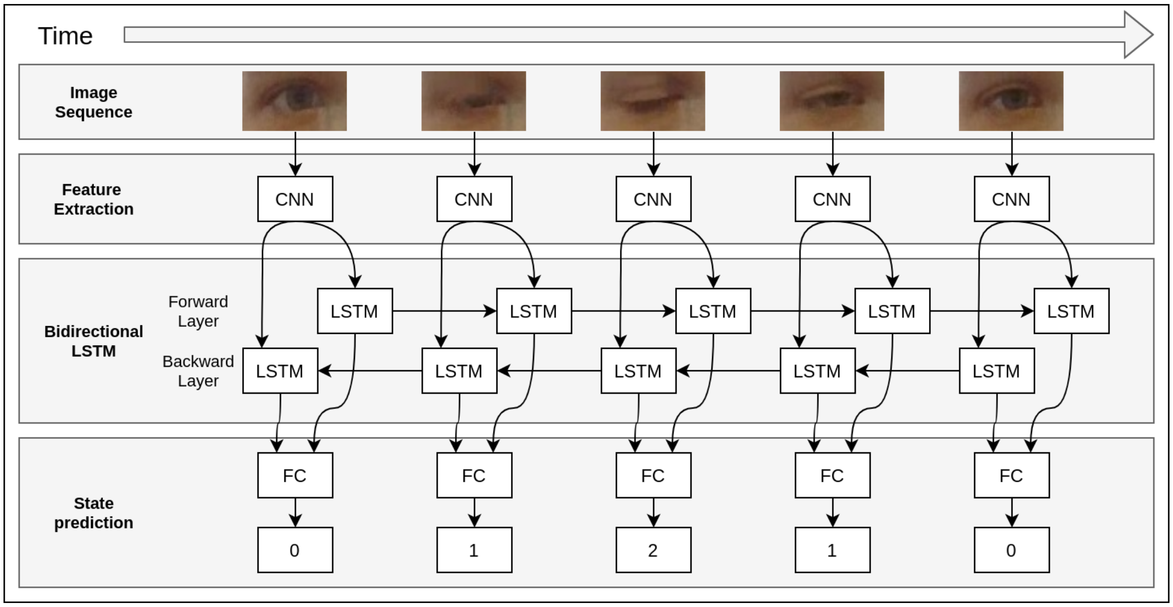

Figure 1. Eye-LRCN: a CNN extract the relevant features from the input images, the generated embeddings are fed to a bidirectional LSTM that performs sequence learning, and its outputs feed a fully connected (FC) layer with a softmax activation function. Note that only one of the two layers of the bidirectional LSTM is shown for simplicity reasons, and that the number of units of the FC layer depends on the learning task. This example illustrates the blink completeness detection problem that requires three units to determine if the eye image does not belong to a blink (0), belongs to a blink (1), or is fully closed (2).

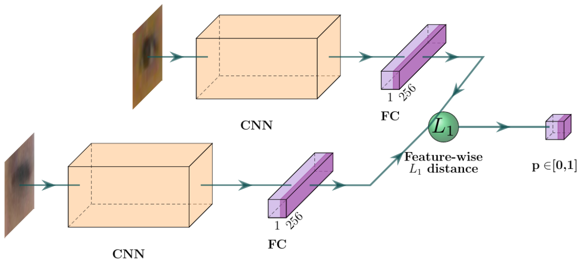

Figure 2. Siamese architecture to train the CNN feature extractor. The output of the network is the probability that both inputs belong to the same class.

eye image as input and converts it into a feature vector. Note that our method can be used with any CNN architecture. The output of the CNN is forwarded to a fully connected layer composed of 256 units. In each training step, a pair of images is propagated through the twin network generating two 256-feature vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$. The siamese networks are joined by applying the feature-wise L_1 distance between the two vectors. Note that L_1 is the most preferable metric for high dimensional applications since it provides the best contrast between the different points [47]. A final single sigmoid unit is used to calculate the probability $\mathbf{p}$ that both images belong to the same class, defined as follows:

$$\mathbf{p} = \sigma \left(\sum_j \delta_j \left| \mathbf{v}_{1,L_1}^{(j)} - \mathbf{v}_{2,L_1}^{(j)} \right| \right) \tag{1}$$

where σ is the sigmoid activation function and δ_j represent the parameters learned by the model during training, weighting the importance of the feature-wise L_1 distance.

Siamese neural networks are trained using pairs of images. Given that we want the network to learn the similarity/dissimilarity between open, closed, and partially closed eye images, the training samples can be reused in different pairs. In this manner, the number of pairs used to train the network can be increased. In particular, the number of training pairs for a dataset composed of n images is calculated as:

$$N = \frac{n^2 - n}{2} \tag{2}$$

The class imbalance between open and closed eyes is solved by training the network with balanced mini-batches. Balanced mini-batches have been shown to be effective for problems with unbalanced data and show greater generalization ability than other techniques such as oversampling and undersampling [12]. Let C be the number of classes in the dataset and B the batch size, a balanced mini-batch is composed of:

- $B/2$ pairs of images such that the two images belong to different classes c_i and c_j, $\forall i, j \in \{1, ..., C\}, i \neq j$.
- $B/2C$ pairs of images such that the two images belong to the same class c_i, $\forall i \in \{1, ..., C\}$.

B. Bidirectional LSTM

In blink detection problems the temporal dimension is very important. During incomplete blinks the eye never closes completely, making it difficult for CNNs to predict if an eye image belongs to a blink or not by just looking at individual frames. In general, CNNs achieve very good performance classifying open and fully closed eyes, but standalone CNNs are not enough when predicting incomplete blinks.

Recurrent neural networks (RNNs) [48] are designed to work with sequential inputs, such as text, speech, or videos, for classification and prediction purposes. Unlike traditional neural networks, RNNs are not limited by the length of the inputs and can use the temporal context to generate better predictions. RNNs allow to retain information from previously processed inputs by using hidden states, allowing to analyze the current input in the context of the previous and next ones.

We use a bidirectional LSTM [13] with two layers composed of 256 LSTM cells. This network receives sequences of feature vectors generated by the CNN feature extractor and, as stated before, each feature vector represents an eye picture and is composed of 256 features. The network performs many-to-many sequence predictions, with a sequence length of 64 frames. Dropout regularization is applied after each LSTM layer with a probability of 0.5 to prevent overfitting.

C. State prediction

The output of the bidirectional LSTM is propagated to a fully connected (FC) layer. The number of units in this last layer depends on the task the network performs; that is, it contains as many units as target classes. Regardless of the learning task considered, a softmax activation function is applied on the output of the FC layer to calculate the probability that the eye image belongs to each class. Finally, the predicted class is the one with the highest probability.

The number of units in the FC layer for each learning task is detailed as follows:

- Simple blink detection: two units. The network has to deal with a binary classification problem to determine if the eye image belongs to a blink (1) or not (0).
- Blink completeness detection: three units. The network has to deal with a multi-class classification problem to determine if the eye image does not belong to a blink (0), belongs to a blink (1), or is fully closed (2). Notice that a blink is complete if the eye is fully closed in at least one of the frames that compose the blink, otherwise the blink is incomplete.
- Eye state detection: two units. The network has to deal with a binary classification problem to determine if the input image corresponds to an open eye (0) or a closed eye (1). Note that Eye-LRCN was designed to solve the other two tasks, but it can be also used for eye state detection due to the similarity between them.

IV. Experimental framework

Several evaluation procedures related to blink detection can be found in the literature, without clear agreement on which one should be used for the problem at hand. This lack of consensus makes it difficult to compare the results obtained in different research works and, although many of them use the same datasets, the ground truth usually differs. Moreover, most datasets are created in laboratory environments and the reported results may not correspond to real-world scenarios.

Another problem is the lack of consensus in the definition of a blink. Many works consider a blink as a frame in which the eye is fully closed [35, 21], while for others a blink consists of a sequence of frames in which the eyelids close and reopen again [19, 26, 27]. In our research, we consider the second approach to be the most appropriate, but we also carried out some experiments using the other one to compare the performance of the proposed method with other works.

The framework used to evaluate blink and blink completeness detection is the one proposed by Fogelton and Benesova [26]. We also used their ground truth annotations for the different datasets considered, which include: 1) for each frame, the id of the blink to which it belongs (if the frame does not belong to any blink, it is annotated as -1); and 2) for each eye, if it is fully closed or not. Note that blinks on left and right eyes are evaluated independently. These annotations are translated into three labels: 0 means that the frame does not belong to a blink, 1 means that the frame belongs to a blink but the eye is not fully closed, and 2 means that the frame belongs to a blink and the eye is fully closed. In this context, an incomplete blink is represented as a sequence of frames all annotated with label 1, whilst in a complete blink at least one of the frames is annotated with label 2.

A. Model training

The CNN and the bidirectional LSTM were trained independently. More specifically, the CNN was trained first and then the LSTM was trained using 256-feature vector sequences generated by the CNN. In both cases, a grid search was performed for hyper-parameter optimization using the Researcher's Night dataset [19]. During this process, the network was trained using the training split and the performance was measured on the validation split. Once the hyper-parameters were adjusted, the network was finally trained with the selected hyper-parameters, using both training and validation splits.

1) Feature extractor training: The CNN used as feature extractor was trained using a siamese architecture to overcome the high imbalance of blink detection problems and the limited number of training samples. According to some preliminary results, the backbone CNN used during the experiments was a ResNet18 [38] pre-trained on the ImageNet dataset [40].

Table I shows the grid search performed to fine-tune three hyper-parameters of the CNN: batch-size, learning rate α, and number of units in the fully connected layer. Balanced mini-batches composed of 128 pairs of images were used to counteract class imbalance. The pairs of images were generated randomly, but always satisfying the mini-batch balance. During training, data augmentation techniques were applied to 50% of the images in order to prevent overfitting. Different techniques were used and combined, including horizontal flips, image scaling between 90%-110% of their original size, translations of up to 10% in both axes and more aggressive techniques such as image blurring, sharpening, embossing, noise, and color inversion, among others. The number of training pairs per epoch was equal to the size of the training set. All eye images were resized to 100×100 pixels before being processed by the network.

The network was trained during 20 epochs, and using the Adam optimizer [49] with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

During hyper-parameter optimization, the performance of the network was analyzed after each epoch, with the aim of evaluating its discrimination power to differentiate between

Table I
Grid search performed for hyper-parameter optimization. The selected values are marked in bold face.

Model	Hyper-parameter	Set of values
CNN	Batch size	{32, 64, **128**}
	Learning rate (α)	{0.001, **0.0001**, 0.00001, 0.000001}
	No. of units	{64, 128, **256**, 512}
RNN	Batch size	{128, 256, **512**}
	Sequence length	{16, 32, **64**}
	Learning rate (α)	{0.001, **0.0001**, 0.00001, 0.000001}
	No. of hidden units	{128, **256**, 512}

open, closed, and partially closed eyes. In order to do this, we used clustering to classify the samples in the validation split. All the images in the training split were processed by the backbone network to extract a 256-feature representation of the images. Then, the centroid of each class (open, closed, and partially closed eyes) in the training split was calculated. Each image in the validation split was processed by the backbone network, and the distance between its 256-feature representation and the three centroids was calculated. Finally, the predicted class for each image was the one corresponding to the closest centroid.

2) Bidirectional LSTM training: The bidirectional LSTM receives sequences of 256-feature vectors generated by the feature extractor. Therefore, eye images are first processed by the CNN feature extractor before being fed to the RNN.

Table I shows the grid search performed to fine-tune four hyper-parameters of the RNN: batch size, sequence length, learning rate α, and number of hidden units per layer. The sequence length represents the number of consecutive frames processed by the network on each step. The experiments carried out showed that the performance of the network was better when using a big sequence length. However, most consumer-level webcams record video at 30 or 60 fps and this frame rate should be considered to perform near real-time video processing. For this reason, sequence lengths greater than 64 were not considered. The maximum batch size has been also limited to 512 frames (8 sequences of 64 frames) to reduce memory usage during training.

The network was trained during 25 epochs, and using the Adam optimizer with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

B. Datasets

The number of available datasets for blink detection has increased in recent years. Although a wide variety of datasets are available at this time, their size tends to be quite small. In this section, we introduce several datasets found in the literature, which were used to train and evaluate the proposed method. Table II summarizes the datasets considered in this research, which are following described in depth.

1) Eyeblink8 dataset [50]: It contains eight videos corresponding to four different individuals (one of them wearing glasses). The videos were recorded at 30 fps in a home environment and the individuals act naturally (smiling, covering face with hands, and looking down). There are 480 eye blinks in a total of 70992 annotated frames with a spatial resolution of 640 × 480 pixels.

Table II
Summary of the datasets used in the experimentation.

Dataset	Subjects	Videos	Frames	Blinks
Eyeblink8 [50]	4	8	70992	480
Researcher's Night [19]	107	107	223116	1849
Talking Face [26]	1	1	5000	61
RT-BENE [21]	16	16	243714	-

2) Researcher's Night dataset [19]: It contains 107 videos with 223116 frames of people reading an article on a computer screen while being recorded. In some videos there is more than one person in the image. People act naturally and around 20% wear glasses, with a total of 1849 blinks annotated. The dataset is composed of Researcher's Night 15 and Researcher's Night 30, which were recorded at 15 and 30 fps, respectively, with a spatial resolution of 640×480. The dataset is divided into training $(1/4)$, validation $(1/4)$, and test $(1/2)$ sets.

3) Talking Face dataset [26]: It contains a single video with 5000 frames corresponding to one single subject talking in front of a camera. The video was recorded at 25 fps, with a spatial resolution of 720×576. This dataset[1] was originally created to evaluate the precision in facial landmark detection. This implies that there is no official ground truth for eye blinks. For this reason, we used the annotations provided by Fogelton and Benesova [26], who reported 61 blinks per eye.

4) RT-BENE dataset [21]: Unlike the datasets annotated by Fogelton and Benesova [26], blinks are not considered as sequences of frames. Eye images are classified between open eyes (at least part of the sclera or pupil is visible) and closed eyes (the eyelids are fully closed). There are 243714 annotated images, corresponding to 16 subjects, and with the following distribution: 218548 are open eyes, 10444 are closed eyes, and 14722 are labeled as uncertain. The dataset is divided into train and test splits, with the images of 12 subjects used for train and the remaining 4 used for test.

C. Performance measures

Blink detection is a highly unbalanced problem in which the number of open eyes samples is much greater than the number of closed eyes samples. In this type of the scenario, the F1-score is a robust metric that is in fact used as a standard in the state-of-the-art. The different methods analyzed in the experimentation were evaluated with the F1-score, which is defined as the harmonic mean of precision and recall:

$$\text{F1-score} = 2 * \frac{\text{precision} \ * \ \text{recall}}{\text{precision} \ + \text{recall}} \tag{3}$$

We also used the inference time to analyze the performance of the proposed method. Note that it refers to the average time that a model takes to infer a single input. This time only measures the feed-forward of the network, whilst other factors

[1] https://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talking_face/talking_face.html

such as the transfer time of the images from CPU to GPU are not taken into account.

V. Results

Four experiments were performed to evaluate Eye-LRCN, including a comparison with some variants of it and the state-of-the-art methods. The first two experiments analyze the blink detection problem following the procedure proposed in [26], which defines a blink as a sequence of frames in which the eyelids close and open again. The third experiment was designed for eye state detection and the fourth to analyze the performance of Eye-LRCN in terms of inference time.

A. Experiment 1 – Simple blink detection

The target of the first experiment is to evaluate Eye-LCRN on simple blink detection and compare it with the state-of-the-art approaches [19, 26]. For this purpose, we used the evaluation method proposed in [26], which is based on the intersection over union (IOU) metric. If the predicted blink has an IOU greater than 0.2 with respect to the ground truth, then it will be considered as a true positive. Other relevant information includes the fact that two consecutive blinks are merged into a single one, and blinks on left and right eyes are evaluated independently.

The model was trained using the train and validation splits of Researcher's Night dataset. The performance of the model was evaluated on the Researcher's Night test set, as well as on the Eyeblink8 and the Talking Face datasets. It is worth mentioning that the last two datasets were not used in the training process, showing the ability of the method to generalize and be applicable to new scenarios.

In order to shed light on the role played by the different elements that make up Eye-LRCN, we also included in the comparison three variants of Eye-LRCN with slight modifications:

1) Eye-LRCN without data augmentation: In order to measure the impact of data augmentation techniques on the quality of Eye-LRCN results, we introduce a variant of Eye-LRCN in which no data augmentation techniques were applied during the training of the CNN feature extractor.

2) Non-Siamese Eye-LRCN: The CNN feature extractor was not trained using a siamese architecture. Instead, the network was trained as a standard CNN that receives as input an image of an eye and returns as output whether the eye is open, closed, or partially closed. Balanced class weights were applied in the loss function to counteract class imbalance. Once the network was trained, the classifier layer of the network was removed and the rest of the network is used as the feature extractor.

3) Unidirectional Eye-LRCN: The bidirectional LSTM that performs sequence learning was replaced by a unidirectional LSTM, which allows making predictions based solely on past inputs with the main advantage of being much less computationally expensive.

Table III shows the results obtained with the different methods considered on the three evaluated datasets. The number of ground truth blinks is also reported. As left and right eyes are evaluated independently, the number of ground truth blinks is almost doubled with respect to the ones reported in Table II. In particular, this number is slightly lower than the double because consecutive blinks are merged and, in some cases, one of the eyes is not visible.

Table III
F1-score of the different methods considered applied to simple blink detection. RN stands for Researcher's Night dataset. Best results are marked in bold face.

	RN (test)	Eyeblink8	Talking Face
No. of ground truth blinks	1447	804	122
Fogelton and Benesova [19]	0.800	0.916	0.930
Fogelton and Benesova [26]	0.879	0.913	0.971
Eye-LRCN	**0.906**	0.946	**0.979**
Eye-LRCN (without DA)	0.847	0.840	0.958
Non-Siamese Eye-LRCN	0.893	**0.953**	0.971
Unidirectional Eye-LRCN	0.856	0.898	0.958

It is worth noting that the results presented in [19] were achieved using a different evaluation procedure. Fogelton and Benesova [26] proposed a new evaluation procedure, and used it not only to evaluate their proposal but also to re-evaluate the method presented in [19]. Therefore, the results reported in [26] for both methods are the ones included in Table III.

As can be observed in Table III, Eye-LRCN and Non-Siamese Eye-LRCN achieve better results than the ones proposed by Fogelton and Benesova [19, 26] in the three datasets considered. Eye-LRCN obtains the best results in Researcher's Night and Talking Face datasets, with an improvement of 0.027 and 0.008 with respect to [26]. Non-Siamese Eye-LRCN obtains the best results in Eyeblink8 dataset, with an improvement of 0.037 with respect to [26].

Comparing the different Eye-LRCN variations, Non-Siamese Eye-LRCN results are similar to Eye-LRCN, but they are slightly worse in the Researcher's Night dataset. Notice that Researcher's Night dataset is the most similar dataset to real-world environments, demonstrating the adequacy of our method to work on these scenarios. Regarding Unidirectional Eye-LRCN, we observe that the results are considerably worse than those of Eye-LRCN. From these results, we can state that the use of bidirectionality in the LSTM has a great impact on the quality of the predictions made by our method. Eye-LRCN without data augmentation has the worst performance of all proposed Eye-LRCN variants in all datasets. The results show that the use of data augmentation techniques has a great impact on the quality of the results obtained by Eye-LRCN, helping the model to generalize and reducing overfitting.

B. Experiment 2 – Blink completeness detection

The objective here is to evaluate Eye-LRCN on blink completeness detection. As in Experiment 1, we compare it with the three variants previously described (Eye-LRCN without data augmentation, Non-Siamese, and Unidirectional) and the state-of-the-art method [26]. Regarding the evaluation procedure, we also used the one previously described but extended to differentiate between complete and incomplete blinks. Note that the F1-score for the complete and incomplete blinks was calculated independently. The model was also trained using the training and validation splits of the Researcher's Night dataset, and evaluated over Researcher's Night test set and the Eyeblink8 and the Talking Face datasets.

Table IV shows the results achieved by the different methods considered on the three evaluated datasets. The number of ground truth complete and incomplete blinks is also reported. Notice that these numbers differ from those presented in Experiment 1 because complete and incomplete blinks are processed independently and, therefore, only double blinks of the same type are merged.

Table IV
F1-SCORE OF THE DIFFERENT METHODS CONSIDERED APPLIED TO BLINK COMPLETENESS DETECTION, BOTH FOR COMPLETE (COMP.) AND INCOMPLETE (INCOM.) BLINKS. RN STANDS FOR RESEARCHER'S NIGHT DATASET. BEST RESULTS ARE MARKED IN BOLD FACE.

		RN (test)	Eyeblink8	Talking face
No. of ground truth blinks	Comp.	1042	762	119
	Incom.	433	43	3
Fogelton and Benesova [26]	Comp.	0.744	0.893	0.939
	Incom.	0.466	**0.337**	0.250
Eye-LRCN	Comp.	**0.893**	**0.900**	**0.952**
	Incom.	**0.585**	0.257	**0.500**
Eye-LRCN (without DA)	Comp.	0.835	0.873	0.888
	Incom.	0.482	0.133	0.182
Non-Siamese Eye-LRCN	Comp.	0.882	0.884	0.934
	Incom.	0.532	0.177	0.462
Unidirectional Eye-LRCN	Comp.	0.850	0.878	0.833
	Incom.	0.571	0.224	0.154

As can be seen in Table IV, Eye-LRCN achieves more competitive results in complete blink detection for the three datasets evaluated. In the Researcher's Night and Talking Face datasets, our approach obtains better results both on complete and incomplete blinks, being considerably better when detecting incomplete blinks, with an improvement of 0.119 and 0.25 in the Researcher's Night and Talking Face datasets, respectively. Regarding the Eyeblink8 dataset, our method achieves better results on complete blinks, but the method proposed by [26] performs better on incomplete blinks.

Regarding Eye-LRCN variants, Eye-LRCN outperforms Non-Siamese Eye-LRCN, especially in incomplete blink detection. Unidirectional Eye-LRCN also obtains considerably worse results with respect to Eye-LRCN, but it works better than Non-Siamese Eye-LRCN for incomplete blink detection. This seems to indicate that the use of a siamese architecture is more suitable for the detection of incomplete blinks. As in the previous experiment, Eye-LRCN without data augmentation obtains the worst results of all the proposed Eye-LRCN variants, showing the impact of the use of data augmentation techniques on the generalizability of the model.

C. Experiment 3 – Eye state detection

This experiment aims to study the performance of Eye-LRCN in eye state detection tasks. In this case, the model was trained to perform binary classification between open and fully closed eyes. Note that our approach is not designed for eye state detection, but with this experiment, we aim to show that it can also obtain competitive results on this task.

The experiment includes a comparison with two different approaches [21, 35]. There are some important differences between these two methods and ours, which must be taken into account in the evaluation procedure. In our approach left and right eyes are evaluated independently, while the blink prediction is performed per frame in [21], using both left and right eyes as input. For this reason, and with the aim of providing a fair comparison, our method was slightly modified: unlike the first two experiments, the bidirectional LSTM receives as input the concatenation of the left and right 256-feature vectors instead of evaluating each one independently.

In eye state detection, blinks are evaluated as individual closed eye images and not as sequences of frames. Hence, the temporal context of the inputs may not be as relevant as in simple blink and blink completeness detection. In order to assess the impact of temporal context in this task, we also evaluated a modified version of our method in which the RNN was replaced by a feedforward neural network. We call this modified version Eye-FFCN. The network has a unique hidden layer with 32 units, followed by a single sigmoid unit in the output layer. Dropout regularization with probability 0.5 is applied after the hidden layer to prevent overfitting. The model was trained during 25 epochs, with a learning rate α of 0.0001, and a batch size of 32. We also included Non-Siamese Eye-LRCN in the comparative in order to determine the impact of the siamese architecture in eye state detection tasks.

The performance of the models was evaluated on Eyeblink8, Researcher's Night, and RT-BENE datasets using the same strategy as in [21]. Therefore, we used a 3-fold cross-validation in Eyeblink8 and RT-BENE datasets (see Tables VI and V); and the standard evaluation procedure using the training, validation, and test splits in the Researcher's Night dataset.

Table V
3-FOLD CROSS-VALIDATION PERFORMED IN THE EYEBLINK8 DATASET. EACH NUMBER REPRESENTS ONE OF THE EIGHT VIDEOS IN THE DATASET. VIDEOS ARE NUMBERED FROM 1 TO 11, BUT NUMBERS 5, 6 AND 7 DO NOT EXIST IN THE DATASET.

	Train split	Test split
Fold 1	1, 2, 10, 8, 4, 11	3, 9
Fold 2	2, 3, 10, 8, 9	1, 4, 11
Fold 3	1, 3, 9, 4, 11	2, 10, 8

Table VI
3-FOLD CROSS-VALIDATION PERFORMED IN THE RT-BENE DATASET. EACH NUMBER REPRESENTS ONE OF THE 16 INDIVIDUALS IN THE DATASET. VIDEOS ARE NUMBERED FROM 0 TO 16, BUT NUMBER 6 WAS DISCARDED FROM THE DATASET BY THE ORIGINAL AUTHORS.

	Train split	Validation split	Test split
Fold 1	3, 4, 5, 7, 9, 12, 13, 14	0, 11, 15, 16	1, 2, 8, 10
Fold 2	1, 2, 5, 8, 10, 12, 13, 14	0, 11, 15, 16	3, 4, 7, 9
Fold 3	1, 2, 3, 4, 7, 8, 9, 10	0, 11, 15, 16	5, 12, 13, 14

Table VII shows the results obtained with the different methods considered on three datasets. Note that Eye-LRCN and Non-Siamese Eye-LRCN are not designed for this task, since they predict blinks as sequences of frames rather than as individual frames.

Table VII
F1-score of the different methods considered applied to eye state detection. RN stands for Researcher's Night dataset. Best results are marked in bold face.

	RN (test)	Eyeblink8	RT-BENE
Anas et al. [35]	-	0.834	0.529
Cortacero et al. [21]	**0.913**	**0.976**	**0.721**
Eye-LRCN	0.807	0.910	0.602
Non-Siamese Eye-LRCN	0.742	0.902	0.702
Eye-FFCN	0.804	0.909	0.590

As can be observed, Eye-LRCN and its two variants outperform the approach of Anas et al. [35] for the two datasets in which it has been evaluated. In particular, Eye-LRCN achieves an improvement of 0.076 and 0.073 in the Eyeblink8 and the RT-BENE datasets, respectively. In contrast, the approach proposed by Cortacero et al. [21] achieves better results than our method in the three datasets evaluated. These results suggest that a CNN-based architecture is more appropriate for eye state detection than our approach. That is, our architecture is more suitable for problems in which a blink is defined as an action over time, as is the case with simple blink and blink completeness detection.

Eye-LRCN obtains slightly better results than the feedforward version in the three datasets evaluated. These results suggest that the temporal context of the inputs has a very limited impact in eye state detection tasks. That is, the quality of the results depends mostly on the quality of the feature extractor. Regarding Non-Siamese Eye-LRCN, it obtains worse results than Eye-LRCN in Researcher's Night and Eyeblink8 datasets, but beats Eye-LRCN in RT-BENE dataset.

D. Experiment 4 - Inference time

The objective of the last experiment is to make a comparison of the interference times of the proposed model and its variants. In this experiment, the inference time is defined as the average time it takes the network to feedforward a single frame. The average inference time per frame was calculated using 2000 iterations and a batch size of 512 frames. To better illustrate the topology of the evaluated networks, Table VIII shows the size, the number of parameters, and the depth of the models. Notice that the experimentation was carried out on an NVIDIA Titan XP GPU.

Table VIII
Topology of the different Eye-LRCN versions analyzed. The backbone network used in the CNN was a ResNet18 [38].

		Params.	Size (MB)	Layers
Eye-LRCN	CNN	11.307.840	45.23	18
	LSTM	2.630.658	10.52	2
Non-Siamese	CNN	11.307.840	45.23	18
Eye-LRCN	LSTM	2.630.658	10.52	2
Unidirectional	CNN	11.307.840	45.23	18
Eye-LRCN	LSTM	1.053.186	4.21	2
Eye-FFCN	CNN	11.307.840	45.23	18
	FF	16.449	0.07	2

Table IX
Inference time (milliseconds) of the different methods for simple blink and blink detection (B) and eye state detection (E) tasks. Notice that left and right eyes are evaluated at the same time in eye state detection. Note also that the classifier is a LSTM in all cases but Eye-FFCN, which uses a feedforward neural network.

	Task	Feat. extractor	Classifier	Total
Eye-LRCN	B	0.1057	0.0060	0.1117
Non-Siamese Eye-LRCN	B	0.1055	0.0059	0.1114
Unidirectional Eye-LRCN	B	0.1057	0.0037	0.1094
Eye-LRCN	E	0.1939	0.0055	0.1994
Non-Siamese Eye-LRCN	E	0.1933	0.0057	0.1990
Eye-FFCN	E	0.1932	0.0004	0.1936

Table IX shows the inference time of the different methods on blink and blink completeness detection tasks (B in column Task). As can be seen, almost 95% of the total inference time of the model corresponds to the feature extractor. The LSTM network performs sequence learning on 256-feature inputs, which is not very computationally expensive. Using a unidirectional LSTM network cuts this component's inference time by almost half. This reduction in time corresponds with a reduction in the number of network parameters with respect to its bidirectional counterpart, as can be seen in Table VIII. However, as it constitutes a very small part of the total inference time, this reduction does not have a significant impact on the inference time.

The inference time of the different methods on eye state detection task are also depicted in Table IX (E in column Task). As can be seen, the inference time of the feature extractor practically doubles with respect to those of column B. This is because in eye state detection problem the blink is analyzed at the frame level, and therefore the feature extractor has to process both left and right eyes. It is worth mentioning that the Eye-FFCN network classifier is notably faster than the methods that use an LSTM, since it uses a feedforward network that does not take into account the temporal context when making predictions.

VI. Conclusions

Computer vision syndrome is highly related to a decrease in blink rate and an increase in the prevalence of incomplete blinks and can have a significant impact on visual comfort and occupational productivity. For this reason, eye blink assessment is fundamental to determine the causes of the syndrome. Eye blink detection is a challenging problem because of the fast and spontaneous nature of blinking. In this context, we propose Eye-LRCN, a novel approach to blink detection and blink completeness detection based on a LRCN that combines a siamese CNN for feature extraction with a bidirectional LSTM for sequence learning.

We tackled two of the main issues that arise in blink detection research. On the one hand, we used a siamese architecture during CNN training to mitigate the class imbalance between closed and open eyes, allowing to generate relevant features that define the eye state based on their position in the feature space. On the other hand, we used transfer learning and data augmentation to deal with a small number of samples, as is usually the case in blink detection datasets.

Our approach outperforms the state-of-the-art methods in simple blink detection and blink completeness detection in most of the datasets evaluated. More specifically, it is clearly superior when detecting incomplete blinks, showing a noticeable improvement over the other methods considered. Furthermore, it obtains remarkable results in eye state detection, even though it was not designed for this task.

As future research, we plan to continue exploring new ways of training and improving our siamese network. In particular, we will use image triplets and perform on-line triplet selection to train the network with the most challenging image combinations. Given that blinks are actions based on eye movements, we also plan to explore the use of optical flow to predict eye blinks. In recent years, some effective CNN-based methods for optical flow estimation have been introduced and could potentially be applied to blink detection. Since the feature extractor accounts for 95% of the total inference time of our model, we would also like to explore the use of other lightweight and efficient CNN architectures with a better balance between performance and computational cost. Finally, we would also like to address the timeframe variation problem, using normalization techniques to standardize the number of frames per second (fps) of the videos processed by the model based on the average eye state time.

References

[1] C. Blehm, S. Vishnu, A. Khattak, S. Mitra, and R. W. Yee, "Computer vision syndrome: a review," *Survey of Ophthalmology*, vol. 50, no. 3, pp. 253–262, 2005.

[2] M. Rosenfield, "Computer vision syndrome: a review of ocular causes and potential treatments," *Ophthalmic and Physiological Optics*, vol. 31, no. 5, pp. 502–515, 2011.

[3] M. Argilés, G. Cardona, E. Pérez-Cabré, and M. Rodríguez, "Blink rate and incomplete blinks in six different controlled hard-copy and electronic reading conditions," *Investigative Ophthalmology & Visual Science*, vol. 56, no. 11, pp. 6679–6685, 2015.

[4] P. Wolkoff, J. K. Nøjgaard, P. Troiano, and B. Piccoli, "Eye complaints in the office environment: precorneal tear film integrity influenced by eye blinking efficiency," *Occupational and Environmental Medicine*, vol. 62, no. 1, pp. 4–12, 2005.

[5] Z. Yan, L. Hu, H. Chen, and F. Lu, "Computer vision syndrome: A widely spreading but largely unknown epidemic among computer users," *Computers in Human Behavior*, vol. 24, no. 5, pp. 2026–2042, 2008.

[6] G. Cardona, C. García, C. Serés, M. Vilaseca, and J. Gispets, "Blink rate, blink amplitude, and tear film integrity during dynamic visual display terminal tasks," *Current Eye Research*, vol. 36, no. 3, pp. 190–197, 2011.

[7] C. Coles-Brennan, A. Sulley, and G. Young, "Management of digital eye strain," *Clinical and Experimental Optometry*, vol. 102, no. 1, pp. 18–29, 2019.

[8] J. K. Portello, M. Rosenfield, and C. A. Chu, "Blink rate, incomplete blinks and computer vision syndrome," *Optometry and Vision Science*, vol. 90, no. 5, pp. 482–487, 2013.

[9] P. J. Driver and M. A. Lemp, "Meibomian gland dysfunction," *Survey of Ophthalmology*, vol. 40, no. 5, pp. 343–367, 1996.

[10] C. A. Blackie, D. R. Korb, E. Knop, R. Bedi, N. Knop, and E. J. Holland, "Nonobvious obstructive meibomian gland dysfunction," *Cornea*, vol. 29, no. 12, pp. 1333–1345, 2010.

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[12] R. Shimizu, K. Asako, H. Ojima, S. Morinaga, M. Hamada, and T. Kuroda, "Balanced mini-batch training for imbalanced image data classification with neural network," in *First International Conference on Artificial Intelligence for Industries*, 2018, pp. 27–30.

[13] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[14] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.

[15] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," *arXiv preprint arXiv:1801.02143*, 2018.

[16] K.-A. Kwon, R. J. Shipley, M. Edirisinghe, D. G. Ezra, G. Rose, S. M. Best, and R. E. Cameron, "High-speed camera characterization of voluntary eye blinking kinematics," *Journal of the Royal Society Interface*, vol. 10, no. 85, p. 20130227, 2013.

[17] J. A. Stern, L. C. Walrath, and R. Goldstein, "The endogenous eyeblink," *Psychophysiology*, vol. 21, no. 1, pp. 22–33, 1984.

[18] B. Remeseiro, A. Fernández, and M. Lira, "Automatic eye blink detection using consumer web cameras," in *International Work-Conference on Artificial Neural Networks*, 2015, pp. 103–114.

[19] A. Fogelton and W. Benesova, "Eye blink detection based on motion vectors analysis," *Computer Vision and Image Understanding*, vol. 148, pp. 23–33, 2016.

[20] B. D. Eddine, F. N. Dos Santos, B. Boulebtateche, and S. Bensaoula, "EyeLSD a robust approach for eye localization and state detection," *Journal of Signal Processing Systems*, vol. 90, no. 1, pp. 99–125, 2018.

[21] K. Cortacero, T. Fischer, and Y. Demiris, "RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environments," in *IEEE International Conference on Computer Vision Workshops*, 2019, pp. 1–10.

[22] T. Appel, T. Santini, and E. Kasneci, "Brightness- and motion-based blink detection for head-mounted eye trackers," in *ACM International Joint Conference on*

Pervasive and Ubiquitous Computing: Adjunct, 2016, pp. 1726–1735.

[23] R. Hershman, A. Henik, and N. Cohen, "A novel blink detection method based on pupillometry noise," *Behavior Research Methods*, vol. 50, no. 1, pp. 107–114, 2018.

[24] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[25] T. Soukupová and J. Cech, "Real-Time Eye Blink Detection using Facial Landmarks," in *21st Computer Vision Winter Workshop*, 2016, pp. 1–8.

[26] A. Fogelton and W. Benesova, "Eye blink completeness detection," *Computer Vision and Image Understanding*, vol. 176, pp. 78–85, 2018.

[27] G. Hu, Y. Xiao, Z. Cao, L. Meng, Z. Fang, J. T. Zhou, and J. Yuan, "Towards real-time eyeblink detection in the wild: Dataset, theory and practices," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2194–2208, 2019.

[28] P. S. Lamba, D. Virmani, and O. Castillo, "Multimodal human eye blink recognition method using feature level fusion for exigency detection," *Soft Computing*, vol. 24, pp. 16 829–16 845, 2020.

[29] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[30] W. Han, Y. Yang, G.-B. Huang, O. Sourina, F. Klanner, and C. Denk, "Driver drowsiness detection based on novel eye openness recognition method and unsupervised feature learning," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1470–1475.

[31] A. A. Jordan, A. Pegatoquet, A. Castagnetti, J. Raybaut, and P. Le Coz, "Deep learning for eye blink detection implemented at the edge," *IEEE Embedded Systems Letters*, pp. 1–4, 2020.

[32] M. Wang, L. Guo, and W.-Y. Chen, "Blink detection using Adaboost and contour circle for fatigue recognition," *Computers & Electrical Engineering*, vol. 58, pp. 502–512, 2017.

[33] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in *IEEE International Workshop on Information Forensics and Security*, 2018, pp. 1–7.

[34] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, 2014.

[35] E. R. Anas, P. Henriquez, and B. J. Matuszewski, "Online eye status detection in the wild with convolutional neural networks," in *International Conference on Computer Vision Theory and Applications*, vol. 7, 2017, pp. 88–95.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] K. W. Kim, H. G. Hong, G. P. Nam, and K. R. Park, "A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor," *Sensors*, vol. 17, no. 7, p. 1534, 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[42] P. Bedi, N. Gupta, and V. Jindal, "Siam-ids: Handling class imbalance problem in intrusion detection systems using siamese neural network," *Procedia Computer Science*, vol. 171, pp. 780–789, 2020.

[43] B. Mac, A. R. Moody, and A. Khademi, "Siamese content loss networks for highly imbalanced medical image segmentation," in *Medical Imaging with Deep Learning*, 2020, pp. 503–514.

[44] K. Malialis, C. G. Panayiotou, and M. M. Polycarpou, "Data-efficient online classification with siamese networks and active learning," in *International Joint Conference on Neural Networks*, 2020, pp. 1–7.

[45] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[46] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015, pp. 1–8.

[47] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International Conference on Database Theory*, 2001, pp. 420–434.

[48] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *11th Annual Conference of the International Speech Communication Association*, 2010, pp. 2877–2880.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015, pp. 1–15.

[50] T. Drutarovsky and A. Fogelton, "Eye blink detection using variance of motion vectors," in *European Conference on Computer Vision*, 2014, pp. 436–448.