Eye Movement Tracking for Computer Vision Syndrome using Deep Learning Techniques

Manan Popat[1], Divyan Goyal[1], Vibhum Raj[1], Nirmal Jayabalan[2], and Chittaranjan Hota[1]

[1]Department of Computer Science, BITS Pilani, Hyderabad Campus, India
[2]Department of Pharmacy, BITS Pilani, Hyderabad Campus, India
{f20200029, f20200042, f20200247, nirmalj, hota}@hyderabad.bits-pilani.ac.in

Abstract— **Due to the increased usage of digital devices in daily life, particularly among children, symptoms such as drying of the eyes, eye strain, headaches, blurred vision, etc., have become recurrent nowadays. Extensive use of computers and smartphones may lead to a common eye-related condition known as Computer Vision Syndrome (CVS). It is often characterized by a reduced blinking rate of the user. In this paper, we propose a deep neural network and computer vision-based machine learning model that entails training a Convolutional Neural Network (CNN) to detect eye blinks, and monitoring blink rates with a Long Short-Term Memory (LSTM) network. This model can be incorporated into smartphones and computers in the form of background apps and may help prevent the risk of CVS or similar disorders. Inferences about the blink rate and eye movement patterns have also been identified. Our model is implemented using TensorFlow and Dlib libraries and has been trained on the Closed Eyes in the Wild (CEW) dataset. The network achieved an accuracy of 94.2% when trained on non-RGB images of eye patches and 91.4% on RGB facial images in real-time.**

I. INTRODUCTION

Computer Vision Syndrome (CVS) is a term that is commonly used nowadays to refer to a number of vision- and eye-related difficulties caused by the increased usage of computers, tablets, mobile phones, and other electronic devices [1]. When not treated, this syndrome, also known as "digital eye strain", causes problems such as dry eyes, astigmatism, far-sightedness, presbyopia, and others [2]. The use of such digital gadgets has grown dramatically across all age categories, especially among children. It is critical to monitor device usage to avoid potential eyesight problems. Drying of the tear film is the primary cause of the fatigue, burning sensation, and discomfort experienced due to CVS. Blinking is an inbuilt mechanism that prevents tear film drying and ensures the ocular surface is lubricated [3]. Hence, we are proposing to develop an intelligent application to monitor the blinking rate, which is a vital sign of CVS.

Existing research in eye blink detection for CVS has explored various methods, but several limitations have been identified. One study [4] tackled edge extraction but struggled with poor lighting conditions, highlighting the challenge of capturing accurate data under uncontrolled lighting. Another work [5] focused on smartphone-based detection, finding that consistent lighting was essential for accurate results, posing difficulties in outdoor environments. Moreover, a method utilizing Harris corner detection [6] faced limitations as it required setting distinct threshold values for each image, potentially leading to noise-induced point detection when using low thresholds. These works underscore the need for robust illumination strategies, stable lighting conditions, and improved threshold determination methods to enhance the accuracy and reliability of eye blink detection techniques for CVS mitigation.

Bennett *et al.* [7] proposed a method for pupil segmentation and gaze estimation in MR eye videos using a fully convolutional neural network. However, their approach mainly focused on static frame-by-frame training and did not address the real-time monitoring of eye blinking rates in continuous live streams, which is crucial for CVS assessment. In another study, Jurczak *et al.* [8] utilized CNN for eye blink artifacts removal from EEG signals, showcasing the potential of CNNs for classification tasks. However, their research focused on a different data type and domain, and the application to CVS eye blink detection requires a tailored and robust approach.

Vishesh *et al.* [9] developed a blink detection model using MobileNetV2-based CNN to assess drowsiness levels in drivers. Additionally, Gomaa *et al.* [10] proposed a CNN-LSTM based deep learning approach in the same context. The current work lays emphasis on monitoring blink patterns during screen interaction, that often involves close-up images of eyes to accurately assess blink quality and frequency. Datasets in this domain consist of individuals using digital devices under various lighting conditions. In contrast, driver drowsiness detection necessitates real-time monitoring of blinks from a distance, usually through vehicle-mounted cameras. Techniques often involve more robust algorithms due to the need for accurate drowsiness prediction and timely intervention. Consequently, while both contexts utilize blink detection, the strategies and datasets diverge to accommodate distinct goals and operational conditions.

In light of these existing studies, our research introduces a novel application of a combined approach utilizing a convolutional neural network (CNN) to extract eye features, followed by a long short-term memory (LSTM) network to model the temporal evolution of eye blinking in a continuous live stream input of individuals using a digital device. The general motivation behind using a CNN network for classification is their ability to learn and identify patterns in visual data and automatically extract features from raw images. Additionally, LSTM layers help model temporal dependencies in sequential data and are commonly used for taking a continuous input stream (video frames) and monitoring the blink rate in real-time with accuracy and robustness.

Our methodology involved applying the proposed algorithm to a diverse dataset of images of individual faces and eye patches collected from various sources. The output of our model provides frame-level classifications, labelling each frame as either "eye open" or "eye closed". This research makes the following key contributions:

(i) Introduction of a dual-stage network that combines eye features extracted via a CNN with their dynamic representation modeled by an LSTM. This approach is tailored specifically for analyzing live stream videos of digital device users.

(ii) Thorough experimental evaluation of model parameters, such as LSTM cell units and layers, and deploying the model using OpenCV and Dlib for real-time monitoring.

(iii) A survey of the various studies on the impact of usage of electronic devices on the blink rate and eye movement patterns.

The experimental results demonstrated the model's robustness in detecting eye states, effectively handling considerable variations in visual quality and eye positions.

II. INFERENCES ABOUT IMPACT ON EYE BLINK RATE — A SURVEY

A lot of medical studies [11], [12] have shown that the act of blinking regularly is crucial for ensuring the dynamic equilibrium between tear formation and tear on the ocular surface. Abnormalities in blinking causes the tear film to dry up, which leads to irritation and pain in the ocular surface. Various studies have investigated the blink rate and the duration between blinks. According to these studies [13], the average spontaneous blink rate ranges between 12 and 15 times per minute. In addition, the period between blinks ranges from 2.8 to 4 seconds or from 2 to 10 seconds. The differences in the results of previous studies may be attributed to differences in the experimental conditions. Under relaxed conditions, a mean blink rate of up to 22 per minute has been reported [14].

In many studies [15], [16], the blink rate has been observed to have reduced from 17 per minute during conversation to about 6 per minute while reading. Moreover, discomfort and tiredness in the eyes are frequently experienced when engaging in near tasks like reading, especially when using electronic devices. There have been numerous studies conducted to investigate the connection between eye fatigue and the use of visual display terminals (VDTs) [17]. People who use VDTs frequently complain of eye discomfort and fatigue. According to a prior study [18], tired eyes was one of the most commonly reported symptoms among office workers (40%), while 30% reported dry eyes and eye discomfort. On an average, normal individuals and those with dry eyes experienced a 56% and 72% decrease in blink rates, respectively [19].

The blink rate of dry eye subjects was significantly higher compared to the control group (according to a t-test with a p-value of 0.017), as identified by a survey [20]. Both groups showed a significant decrease in blink rate during the game and letter tasks (according to a t-test with a p-value of less than 0.04). Both groups exhibited instances of partial blinks and rapid sequences of blinks, yet there was no significant distinction in the amplitude of blinks between the two groups. Tear film break-up was mostly inferior in normal subjects, while dry eye subjects showed more tear break-ups centrally and superiorly [19]. The interaction between tear break-up and blink behavior was complex and observed in real-time video recordings. The researchers observed that dry eye subjects experienced more intense ocular symptoms post-testing compared to the control group. Both groups showed an increase in corneal staining post-testing, which was primarily inferior. A noticeable positive correlation was observed between the total symptom score and the proportion of incomplete blinks, suggesting that a higher occurrence of incomplete blinks was linked to more symptoms. In contrast, a significant inverse relationship was detected between the blink score and symptoms, indicating that a higher blink rate was associated with fewer symptoms.

III. DATA

The dataset we have used for training the model is *Closed Eyes in the Wild* (CEW) [21]. The dataset, prepared by Xiaoyang Tan, includes 2423 images. Of those, 1192 were obtained from the Internet with their eyes closed, while the remaining 1231 were selected from the Labeled Face in the Wild database with their eyes open. Eye patches were collected using a face detector and eye localization software based on the coarse face region and eye position. The cropped coarse faces were resized to 100x100 pixels, and eye patches measuring 24x24 pixels were extracted from the eye position. The dataset consists of three versions: Raw face images with background; Face images warped; and Eye patches only (non-RGB). Fig. 1 shows some samples from the two kinds of images used in this study.

Pre-processing: Firstly, we load the images separately as open and closed (with labels 0 and 1, respectively) from the directory and then combine them in a single list. The list is then shuffled and divided into features (image strings) and target (binary labels), followed by test-train split (with test size = 0.2) and the required reshaping of images to 224x224. The target column is then converted into one-hot encoded vectors.

IV. BINARY CLASSIFICATION MODEL

Transfer learning: Next, we build the model. We used Inception-v3 as the base model (Fig. 2), which serves as a feature extractor, and made the layers non-trainable (*i.e.*, their weights are frozen during training). By doing so, the model can learn to generalize better and hence requires less training since the base model has already learned the complex patterns in visual data (it is one of the state-of-the-art models for image classification). This approach also

Fig. 1. Sample images from "Closed Eyes in the Wild" dataset: closed and open eyes from *(i)* Eye patches only (non-RGB), and *(ii)* Face images warped.

TABLE I

ARCHITECTURAL PARAMETERS FOR THE MODEL.

Parameter	**Value**
Base model	Inception-v3
Pretraining dataset	ImageNet
Reshape layer target shape	(1, 2048)
Number of LSTM units	128
Depth of stacked LSTM	3
Number of neurons in first dense layer	2048
Activation function for first dense layer	ReLU
Dropout layer argument	0.3
Activation function for final dense (classification) layer	Sigmoid

TABLE II

COMPILATION PARAMETERS FOR THE MODEL.

Parameter	**Value**
Model optimizer	Adam
Loss function	Binary cross-entropy
Metric	Accuracy
Number of epochs for training	15

helps reduce the risk of overfitting and allows the model to achieve better accuracy with limited training data.

CNN and LSTM model: This is followed by sequentially arranging the layers to build the model as is typically done for deep learning architectures. The model first takes the base model (which itself is a CNN), followed by an average Pooling layer that reduces the dimensions of the extracted features while retaining all the important information. This is followed by a Reshape layer that reshapes the output of the previous layer to a shape of 1x2048. Next, we have three LSTM layers (with 128 cells each) that learn the temporal dependencies in the features extracted by the previous layers. After the LSTM layers, the output is passed through a Dense (fully connected) layer with 2048 neurons and a ReLU activation function that learns a higher level representation of the input. The next layer is a Dropout layer that randomly drops 30% of the neurons to reduce overfitting. Finally, the output is passed through a Dense layer with a single neuron and a Sigmoid activation function to predict the probability of the input image belonging to one of the two classes (eye open or eye closed) (Fig. 3). The model is compiled using the Adam optimizer, Binary Cross-Entropy loss, and Accuracy as the metric. The model is fitted on the training dataset for 15 epochs, and the predictions are then made over the testing dataset.

V. USING OPENCV AND DLIB FOR EYE-BLINK DETECTION

Extending portability and classification function: The weight parameters obtained from the trained model are then exported for portability. This is followed by a wrapper function that, given a localized eye patch within the usual color space and normal image density, does the required pre-processing (resizing to shape 224x224 and normalizing the image patch) before running it through the classifier and returning the processed output.

Dlib landmark detection: Dlib is an open-source library used heavily for computer vision. It is one of the most widely used libraries providing state-of-the-art performance in facial recognition. Dlib's facial landmark detection model can identify the exact location of specific points on a face, such as the corners of the eyes or the tip of the nose. We used this model for the localization of eye and face patches (which is further used to identify the ROI for the model). By calculating the extreme left and right ends of the eye and applying a buffer, we get the location of individual eyes and their corresponding image patches. These can then be directly run through the model by using the function we created above, and the results be visualized and analyzed in real-time.

Extracting the eye patch: After obtaining the location of the extremities of an individual eye, some calculations were done to extract the Region of Interest (ROI) of the eye, which involved finding the x and y centers and applying a bit of padding to extract the said ROI. This is used for directly running through the classification wrapper to get a label value, which is further extended to measure the eye blink rate in blinks per minute (as outlined in Algorithm 1). These results are finally rendered on the screen.

Note that the model parameters, the sequence of the layers, the reshaping and activation functions, etc., have been found to be optimal for the given dataset. Hyperparameter tuning has been extensively done on the number of sequential

Algorithm 1 Blink Rate Detection

Input:
A video stream or webcam feed;
The trained model for binary classification of images (eye open or closed);

Output:
The blink rate of the user in blinks per minute;

```
Initialize the detector and predictor classes from the dlib library;
for each frame in the video stream or webcam feed do
    Convert the frame to grayscale;
    Detect faces in the grayscale frame using the 'detec--tor' object;
    for each detected face do
        Predict the shape of the face using the 'predictor' object;
        Calculate coordinates of ROI around eye;
        Extract grayscale image patch of ROI and normalize it;
        Resize image patch to required size;
        Classify image patch using the model;
        if current label is 'CLOSE' and previous label was 'OPEN' then
            Record a blink event with current timestamp[1];
        end if
        Update previous label with current label;
    end for
    Augment the frame with model's findings[1];
    Display the augmented frame;
end for
```

model layers[2], dropout parameters, type of pooling layers, and activation functions. The training time was reasonable according to modern standards, and the model size was compact (around 200 MB). Integrating the model with the Dlib library and pre-processing the images using OpenCV has been observed to perform sufficiently well in real-time. It is worth noting that since an average blink lasts between 0.1 and 0.4 seconds, an FPS of around 10 would suffice. Additionally, the actual duration of the blink does not matter as long as the blink duration is not shorter than the frame time (0.1 milliseconds). However, currently the experimental script employed for the eye images model acquires live video input at a frame rate ranging from 8 to 10 FPS. Notably, the entire process is executed solely on the CPU. The potential for performance enhancement is substantial if a GPU-based iteration is pursued. However, there might not be a need to capture such fast blinks because the current setup is enough for real-time monitoring.

[1]These steps serve as placeholders for CVS detection logic, explained in Future Work.

[2]Except the CNN layers, since they are provided by the base Inception-v3 model and are frozen during training.

VI. RESULTS

The combination of CNN and LSTM layers is effective for eye blink classification as it captures both spatial and temporal information in input data. CNN layers automatically extract important features from raw images for blink or non-blink classification. Trained on large image datasets, CNN layers learn filters to identify features, generating a feature map indicating their presence in the image. However, blink classification also involves monitoring temporal changes. LSTM layers, designed for modeling temporal dependencies, process input frames sequentially to learn blink rate patterns over time.

The model was first trained on individual eye patches so as to make it focus solely on the region of the image containing the eye. This was originally intended to result in a more accurate classification of eye blinks, but the fact that the model may struggle to generalize to new images with different backgrounds or lighting conditions was also considered. Moreover, according to the original hypothesis, training the model on complete face images could allow the model to learn to recognize subtle changes in facial expression or head pose, which could be important in detecting eye blinks. This can improve the efficiency and speed of the model in some instances.

However, on running the latter model, we observed a slight drop in the speed of detecting eye blinks, as well as the accuracy of measuring the frequency (Fig. 4). In principle, passing complete facial images to the model makes more sense since the layers would then also take into account the lighting conditions, eyebrows, etc. while classifying. However, in practice, it is computationally inefficient and more prohibitive than the former. It can be assumed that since the facial images dataset is smaller than the eye images dataset, the model finds it difficult to train itself in the presence of outliers (due to lighting condition variations, etc.), and increasing the number of epochs results in overfitting, or that the model may not be as effective at capturing the subtle changes in the eye region that are indicative of an eye blink.

TABLE III
PERFORMANCE OF THE FINAL MODEL (TRAINED OVER 15 EPOCHS).

Dataset	Train accuracy	Test accuracy
Eye patches only (non-RGB)	99.4 %	94.2 %
Face images warped (RGB)	96.7 %	91.4 %

After building the model with the aforementioned architecture, we first trained it on the individual non-RGB images of eye patches from the dataset (Fig. 5). We observed that the training accuracy obtained after 15 epochs was 99.4%, and the testing accuracy was 94.2% (Precision = 91.7% and Recall = 95.9%). The trained model worked significantly well in real-time as well, in terms of speed and accuracy. This was followed by running the model on face-pass images (RGB), giving a 96.7% training accuracy and a 91.4% testing accuracy (Precision = 84.6% and Recall =

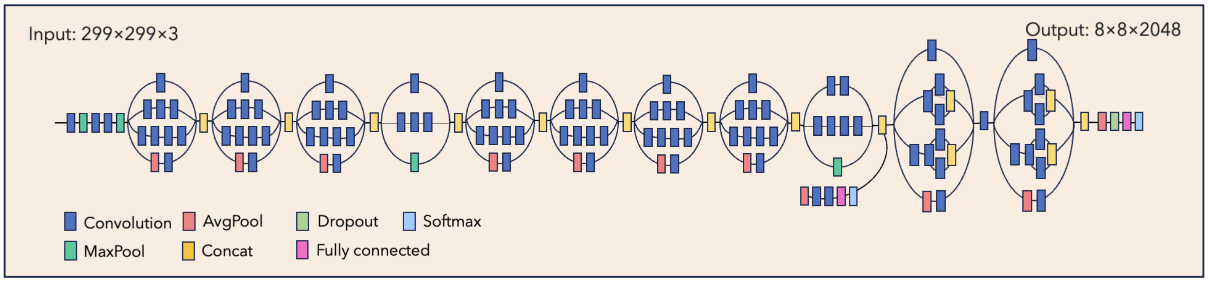

Fig. 2. Leveraging Transfer Learning: Inception-v3 base model (trained on ImageNet dataset) provides pre-trained CNN layers.

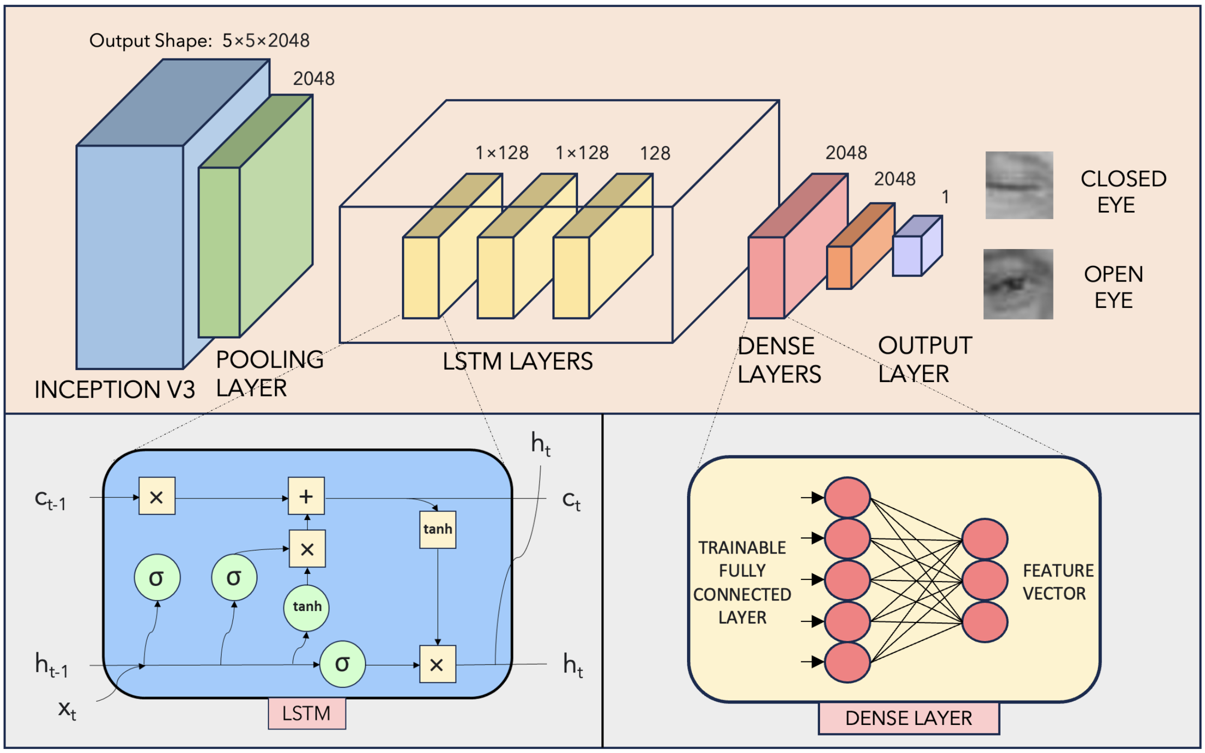

Fig. 3. Model architecture diagram.

93.2%). On increasing the number of epochs used to train the model, a significant improvement was noticed in the training accuracy, which was not the case with individual eye images. However, testing accuracy did not change much. Note that these metrics indicate the ability of the model to classify static images as open or closed eyes. However, the final script used in the experiment, which takes a live stream video of the user as input, also demonstrates notably high performance in accurately assessing the blink rate.

VII. CONCLUSION

After training the model and importing it, the user's live video stream (captured via a webcam) is segmented into a sequence of image inputs, classified independently to detect blinks and consequently the blink rate (in blinks per minute). This script runs well in the background, enabling users to identify the initial stages of CVS and implement preventive actions (such as eye exercise and acupressure [22]).

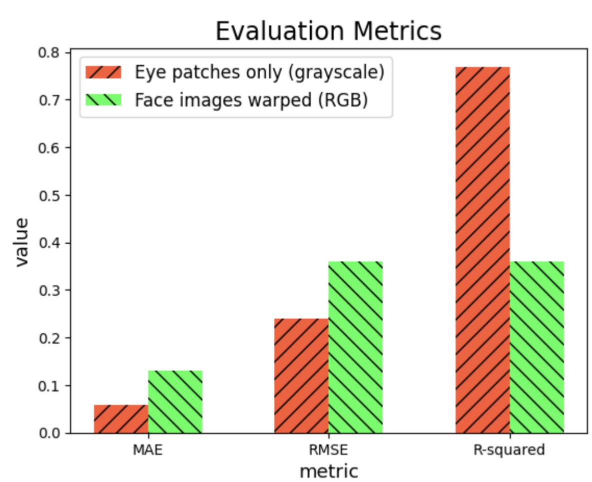

Fig. 4. Bar chart showing the comparison of the performance of the model on the two datasets. The evaluation metrics used are Mean Absolute Error, Root Mean Squared Error, and Coefficient of Determination.

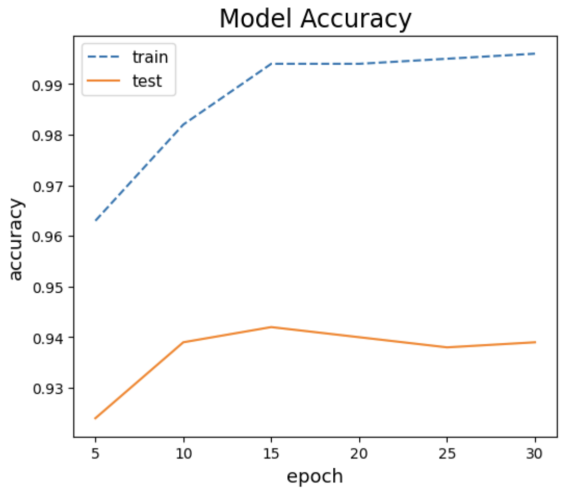

Fig. 5. Training accuracy and testing accuracy plotted as a function of the number of epochs during training of the final model on the "Eye patches only (grayscale)" dataset.

Additionally, the experimental script may prompt reminders for the user to blink. A conclusion can then be drawn by contrasting the user's current blink frequency with the average blinking patterns. However, confirming the presence of CVS is a much more onerous task and there do not exist sufficient datasets and parameter studies that can help in the correct diagnosis. Moreover, we recognize that long-term monitoring and evaluation of several other parameter measurements are imperative to confirm its presence. As such, future research endeavors must encompass extended observation periods and a broader array of metrics to achieve a more conclusive understanding of CVS and its potential associations with ocular health in the context of digital device usage.

VIII. FUTURE WORK

To confirm CVS in users, it is crucial to gather data on various eye conditions linked to it. Red eyes, for instance, are a common CVS symptom and a valuable indicator. Additionally, dry eyes, eye fatigue, and eye strain can suggest CVS. Collecting this data can enable the creation of a predictive model to alert users about CVS risk. Using eye-tracking technology to monitor users' eye movements when using digital devices and analyzing the data through unsupervised learning techniques like clustering and anomaly detection can help identify relevant patterns. It is evident that the blink rate decreases with prolonged use of digital devices, but researchers further suggest that eyes may get constricted and remain half closed while digitally strained [23]. Thus, in this context, the Eye Aspect Ratio (EAR) [24], which could be easily calculated by using Dlib's facial landmarks, could be used as an unsupervised learning parameter to determine the level of openness or closure of the eyes. This information could be collectively used to develop and implement a predictive model for early detection of eye-related issues through an intelligent/cognitive mobile application.

REFERENCES

[1] Rosenfield, M. (2016). Computer vision syndrome (a.k.a. digital eye strain). Optometry in Practice, 17, 1-10.

[2] Pavel, I. A., Bogdanici, C. M., Donica, V. C., Anton, N., Savu, B., Chiriac, C. P., Pavel, C. D., & Salavastru, S. C. (2023). Computer Vision Syndrome: An Ophthalmic Pathology of the Modern Era. Medicina (Kaunas, Lithuania), 59(2), 412.

[3] Lapa, I., Ferreira, S., Mateus, C., Rocha, N., & Rodrigues, M. A. (2023). Real-Time Blink Detection as an Indicator of Computer Vision Syndrome in Real-Life Settings: An Exploratory Study. International journal of environmental research and public health, 20(5), 4569.

[4] Jennifer, J & Sharmila, T.. (2017). Edge based eye-blink detection for computer vision syndrome. 1-5. 10.1109/ICCCSP.2017.7944084.

[5] Jumpamule, Watcharee & Thapkun, Tanakron. (2018). Reminding System for Safety Smartphone Using to Reduce Symptoms of Computer Vision Syndrome. 1-4. 10.1109/ICSEC.2018.8712747.

[6] Joshi, Apurv & Kadethankar, Atharva & Patwardhan, Vedant. (2017). Eye blinking detection for the detection of computer vision syndrome. 1-3. 10.1109/IPACT.2017.8244933.

[7] Bennett, R., & Joshi, S. H. (2021). A CNN and LSTM Network for Eye-Blink Classification from MRI Scanner Monitoring Videos. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 3463–3466).

[8] Jurczak, M., Kołodziej, M., & Majkowski, A. (2022). Implementation of a Convolutional Neural Network for Eye Blink Artifacts Removal From the Electroencephalography Signal. Frontiers in neuroscience, 16, 782367.

[9] Vishesh, Pothuraju & S, Raghavendra & Jankatti, Santosh & V, Rekha. (2021). Eye blink detection using CNN to detect drowsiness level in drivers for road safety. Indonesian Journal of Electrical Engineering and Computer Science. 22. 222. 10.11591/ijeecs.v22.i1.pp222-231.

[10] Gomaa, M., Mahmoud, R., & Sarhan, A. (2022). A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction. Journal of Engineering Research, 6(3), 59-70.

[11] Wang, M. T. M., Tien, L., Han, A., Lee, J. M., Kim, D., Markoulli, M., & Craig, J. P. (2018). Impact of blinking on ocular surface and tear film parameters. The ocular surface, 16(4), 424–429.

[12] Pflugfelder, S. C., & Stern, M. E. (2020). Biological functions of tear film. Experimental eye research, 197, 108115.

[13] Abusharha A. A. (2017). Changes in blink rate and ocular symptoms during different reading tasks. Clinical optometry, 9, 133–138.

[14] Tsubota, K., & Nakamori, K. (1993). Dry eyes and video display terminals. New England Journal of Medicine, 328(8), 584-584.

[15] Brych, M., Murali, S., & Händel, B. (2021). How the motor aspect of speaking influences the blink rate. PloS one, 16(10), e0258322.

[16] Schlote, T., Kadner, G., & Freudenthaler, N. (2004). Marked reduction and distinct patterns of eye blinking in patients with moderately dry eyes during video display terminal use. Graefe's archive for clinical and experimental ophthalmology, 242, 306-312.

[17] Parihar, J. K., Jain, V. K., Chaturvedi, P., Kaushik, J., Jain, G., & Parihar, A. K. (2016). Computer and visual display terminals (VDT) vision syndrome (CVDTS). Medical Journal, Armed Forces India, 72(3), 270–276.

[18] Chisari, G., Stagni, E., Rampello, L., Malaguarnera, M., & Chisari, C. (2013). The ocular surface in patients video display terminal (VDT). Acta Medica Mediterranea, 29, 369-374.

[19] Himebaugh, N. L., Begley, C. G., Bradley, A., & Wilkinson, J. A. (2009). Blinking and tear break-up during four visual tasks. Optometry and vision science, 86(2), E106-E114.

[20] Portello, J. K., Rosenfield, M., & Chu, C. A. (2013). Blink rate, incomplete blinks and computer vision syndrome. Optometry and vision science : official publication of the American Academy of Optometry, 90(5), 482–487.

[21] Alparslan, K., Alparslan, Y., & Burlick, M. (2020). Towards Evaluating Driver Fatigue with Robust Deep Learning Models.

[22] Tanamal, Barbizu & Naibey, Rosdiana & Wadiastuti, Sri & Yulidia, Herlina & Pesurnay, Yanti. (2023). Computer Vision Syndrome (CVS) in Medical Students Reduced by Eye Exercise and Acupressure. Babali Nursing Research. 4. 314-329.

[23] Sheppard, A. L., & Wolffsohn, J. S. (2018). Digital eye strain: prevalence, measurement and amelioration. BMJ open ophthalmology, 3(1), e000146.

[24] Sathasivam, S., Mahamad, A. K., Saon, S., Sidek, A., Som, M. M., & Ameen, H. A. (2020). Drowsiness Detection System using Eye Aspect Ratio Technique. IEEE Conference Publication | IEEE Xplore.