Blink Detection for Off-Angle Iris Recognition Using Deep Learning

David Chavarro and Mahmut Karakaya
Department of Computer Science
Kennesaw State University
Marietta, GA 30060 USA
Email: mkarakay@kennesaw.edu

***Abstract*— Iris biometric systems offer non-contact authentication, particularly advantageous in controlled environments such as security checkpoints. However, challenges arise in less controlled scenarios such as standoff biometrics where captured images mostly are non-ideal including off-angle. This paper addresses the need for iris recognition models adaptable to various gaze angles by proposing a blink detection algorithm as an additional feature. The study explores different blink detection methods including involving logistic regression, random forest, and deep learning models. For the first methodology, logistic regression and a random forest model were used to classify eye images into four different blink classes. The second methodology involved labeling eye openness percentage. The ground-truth eye blink was calculated using facial landmarks detected by the MediaPipe model. For the deep learning approach, we used a pre-trained Convolutional Neural Network (CNN) model by replacing the output layer with a regression layer. Results show improved precision and recall when incorporating height and width features for the regression model. The AlexNet model achieves superior performance, reaching 90% accuracy with a 10% error threshold. This research contributes valuable insights for developing robust iris recognition models adaptable to diverse gaze angles.**

Keywords*— *blink detection, computer vision, deep learning

I. INTRODUCTION

Iris biometrics has been one of the most common systems for subject identification, offering distinct advantages over traditional biometrics by enabling contactless authentication for restricted access. Notably, the implementation of iris biometric systems at airport security checkpoints and passport control exemplifies their utility in providing convenient and secure authentication for passengers. However, the recognition performance highly depends on the image quality and their performance is susceptible to degradation in the presence of non-ideal images. Traditional iris recognition methods require subjects to maintain fully open eyes throughout the entire process of data acquisition to ensure accurate iris image collection [1]. In contrast, recent advancements in standoff biometric systems have been tailored to recognize subjects from a distance, obviating the need for subjects to gaze directly at the camera. This paradigm shift introduces additional challenges for iris recognition, encompassing factors such as gaze angle, pupil dilation, and occurrences of eye blinking.

Blinking poses a particular challenge for standoff systems, as the natural reflex action may occur during the acquisition process, complicating gaze estimation. There has been research in iris recognition systems that were designed for frontal iris images including occlusion, blur, and lighting shift [2,3,4]. Recent studies focus on addressing challenges such as elliptical unwrapping [5] and perspective projection [6] for standoff iris images. Improving iris recognition algorithm accuracy involves addressing issues like light refraction on the cornea, eliminating limbus occlusions at sclera boundary, and adjusting various parameters.

(a) (b) (c)

Figure 1: The sample iris images from the dataset (a) frontal and (b-c) off-angle images.

Traditional iris recognition systems are designed for frontal iris images, limiting their effectiveness for different gaze angles. The existing iris databases contain only iris codes from frontal iris images. Addressing this limitation requires the development of models capable of recognizing iris patterns from various angles. Introducing a blink detection algorithm provides the first step toward the potential solutions, with its output serving as an additional feature for iris subject recognition models, leading to improved classification accuracy. The blink detection algorithm can also be applied for related use cases.

The exploration of blink detection algorithms constitutes an emerging area of research. In recent studies, Convolutional Neural Networks (CNN) was trained on iris images annotated with categorical eye openness labels [7]. Although these experiments demonstrated the model's capability to classify blink detection, its reliance on categorical labels presents a limitation. This paper advances the field of blink detection algorithms by extending prior work, specifically by training a CNN on datasets enriched with regression labels indicating varying degrees of eye openness. This augmentation of the data labeling scheme contributes to a more nuanced and continuous representation of eye openness levels, enhancing the sophistication and potential accuracy of the blink detection.

The subsequent sections of this paper are structured as follows: Section II provides an overview of related works in the scope of blink detection. The description of our proposed models is covered in Section III. Section IV expounds upon the experimental setup and presents the experimental results from the implemented models. Finally, Section V summarizes the conclusion drawn from the research findings.

II. RELATED WORKS

The scope of this research has been created as a derivative of multiple deep learning frameworks that were developed for iris recognition. We trained AlexNet and ResNet50 CNN models using off-angle iris dataset via transfer learning. It contained 10,000 images from 100 subjects that were taken at several gaze angles between -50° and +50° with 10° increments, as shown in Fig. 1. Once the models were trained, they were evaluated using the classification accuracies of each gaze angle. For the evaluated gaze angles that were adjacent to the trained gaze angle, the model had a high classification accuracy, while more distant evaluated gaze angles had a lower classification accuracy. Based on these results, it was determined that a deep neural network was effective in detecting the differences in the subject's iris region at off-angle images, while a shallow network could effectively identify subjects at a specific angle [8].

Another experiment involved the creation of a neural network designed to classify subjects experiencing fatigue within a vehicular context. In [9], three network models were trained using the frontal images of the driver from an onboard camera. The first model classified whether the subject's eyes were open, while the second model was a MediaPipe model that extracted eye landmarks. Based on the extracted feature landmarks, a height-to-width ratio was created to measure eye openness. The third model classified how much the subject yawned. Please note that the researchers included yawning as a feature since it was one of the symptoms associated with fatigue.

Furthermore, Saealal et al. devised a neural network capable of distinguishing between authentic and Generative Adversarial Network (GAN)-synthesized videos [10]. They determined that using the subject's blink rate as a feature improved their model's accuracy because most GAN-generated videos did not contain the eye blinking. The researchers noted that the blinking time of an actual subject was between 100 and 400 ms at 10-second intervals. For the synthetic videos containing blinking, the subject's blinking time did not match this characteristic. Given this information, their model's architecture was created with cascading CNNs with LSTMs and included a temporal network that tracked eye input and generated blinking probabilities [10].

Eye-LRCN has been proposed to detect the eye openness of the subject [11]. The potential impact of this research was to have this model implemented to reduce the occurrence of computer vision syndrome, a medical condition where prolonged exposure to computer displays causes eye discomfort. The architecture included a long-term recurrent convolutional network with a Siamese architecture. Because the training dataset for this model contained imbalanced classes and was a small size, the Siamese architecture normalized this dataset. After evaluating the dataset, their model had an accuracy of at least 90% and performed better than the model that detected eyelid movements via flow image methods [12][13].

Other notable related works include an eye-blinking model for a computer interface's input [14], and a face tracking model with integrated blink detection[15] where it generates inter-eye-blink rate and integrated the Haar Cascade Classifier and Camshift algorithms. In [16], the blink detection algorithm utilized a CNN and Support Vector Machine (SVM) where the input was smartphone camera images. At last, CNNs frameworks that classified whether a given subject's eye was open. While both models had high classification performance, they could not classify images where the eye openness was in a partial position.

Please note that while most models cited above had good accuracy in detecting blinks, the output of these models contained either a fully open or closed eye state. In addition, the training dataset of these models was captured at the frontal gaze angle. To determine how the classification accuracy is affected for non-frontal angles, the scope of this research will focus on training a new blink detection model for various gaze angles. Since the eye-to-width ratio was used as an extracted feature for the blink detection model [9], this methodology will be incorporated into this research.

III. METHODOLOGY

For this study, we utilized an off-angle iris dataset [8] to develop our algorithms and train our models. Since the dataset does not contain ground-truth values for eye openness, we used the MediaPipe pipeline to generate eye openness by feeding the subject images into the model. This model includes a pre-trained eye landmark detection model that localizes the coordinates of facial features, as shown in Fig. 2. These eye features included seven points in the subject's upper eyelids (marked as purple UE), seven points in the lower eyelids (marked as red LE), four radially outer points on the iris, a point in the pupils (marked as blue P), a point on the left eyelid corner (marked as orange CL), and one point on the right eyelid corner (marked as green CR).

The height and width of the subject's eyelids were calculated using the Euclidean distance. Please note that the distance is defined as:

$$ED(\mathrm{P}_1, \mathrm{P}_2) = \sqrt{\left(\mathrm{P}_{2,x} - \mathrm{P}_{1,x}\right)^2 + (\mathrm{P}_{2,y} - \mathrm{P}_{1,y})^2} \qquad (1)$$

where $ED(\mathrm{P}_1, \mathrm{P}_2)$ is the distance between points P_1 and P_2.

LE_3 is the coordinates for the lower eyelid point 3, and UE_3 is the coordinates for the upper eyelid point 3. Using these definitions, eye height is the distance between LE_3 and UE_3. To calculate the eye width, we use the distance between the left eyelid corner, C_L, and the right eyelid corner, C_R. Then, the height-to-width ratio is calculated for all images as their ratios.

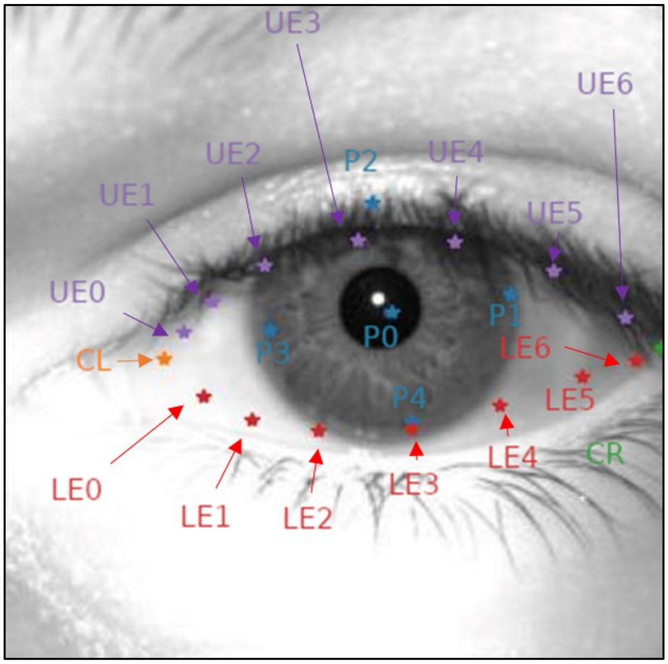

Figure 2: The detected eye landmarks with the MediaPipe model.

The eye openness percentage is calculated as:

$$eyeOpenness = \left(\frac{HtW_i}{HtW_{max}}\right) * 100 \qquad (2)$$

where HtW_{max} is the height-to-width ratio of a fully open eye.

After finding the labels of images in the dataset, we develop two methods for blink detection. The first method consists of training the random forest and logistic regression models on the dataset. We split the dataset into 70%-30% for training and validation datasets. One pair of logistic regression and random forest models was trained using the eyelid and iris points, iris width, and eyelid width and height as features. On the other hand, another set of logistic regression and random forest models was only trained using the eyelid and iris points.

For the second method, a pre-trained AlexNet model was used on the iris dataset using transfer learning, with the eye openness being used as the label and the image used as the feature. AlexNet architecture was trained previously for image classification. It contains five convolutional layers with different kernel sizes. For instance, the first convolutional layer contained a size of 11x11x3, which was followed by a ReLU activation layer. The fourth and fifth convolutional layers consisted of two grouped convolution filters with dimensions 3x3x192. To make the AlexNet model output continuous values of eye openness, the fully connected layer was replaced with a single output and passed into the regression layer.

IV. Experimental Setup and Results

Using the MediaPipe model, we utilized the TensorFlow library to load the model to Python. The model detected the coordinates of the eye landmarks in each iris image and exported the data to a CSV file. This file was used by another script to generate the height-to-width ratio as an additional feature for the dataset. For the first approach, random forest and logistic regression models were implemented by the Sklearn Python library [19]. To ensure these models would not be overfitted, the features of the dataset were normalized using the Sklearn StandardScaler library [20]. In addition, the dataset was randomized, and the data was split 70%-30% between the training and validation datasets.

TABLE I: AlexNet Model Training Parameters

Parameter	Value
Learning Rate Schedule	Piecewise
Learning Rate Drop Factor	0.1
Learning Rate Drop Period	5
Initial Learning Rate	1E-5
Mini Batch Size	64

The performance of the models from the first method was evaluated by calculating the precision and recall for each output class. Precision is calculated by (TP)/(TP+FP) and recall as (TP)/(TP+FN). Note that TP is the number of correct positive classifications, FP is the number of incorrect positive classifications, and FN is the number of incorrect negative classifications. Please note that the logistic regression and random forest models only used eye landmark features for classifying eye openness.

For the second approach, MATLAB was used to train the AlexNet model with stochastic gradient descent where it contains the pre-trained weights for the AlexNet model. This streamlined the transfer learning process. Table 1 describes the training parameters of the AlexNet model. The dataset for the AlexNet model was the images of the iris dataset [8]. As part of the data preparation process for the model, the images were rescaled into the dimensions of 227x227x3. The AlexNet model was trained with regression to estimate the eye blink.

The AlexNet model has been evaluated by two metrics. The first metric consisted of calculating the error thresholds between the model's predicted and actual eye openness percentage. For each error threshold, the accuracy was calculated, where each prediction within the error threshold was indicated as a true prediction. On the other hand, the second metric consisted of the average prediction errors for each class in the dataset, along with the standard deviation. Finally, a visual analysis was created where the predicted labels of the dataset were plotted against the actual labels.

Regarding the results from the first methodology, Fig. 3(a) displays the precision and recall for the logistic regression models and Fig. 3(b) shows results for the random forest models. We observed that recall and precision values of both methods changed from around 70% to 80%. Both models had the lowest precision and recall scores when classifying images with a 32% eye openness label. When the height and width features were included in the inputs as a feature, the precision and recall values for the logistic regression model increased significantly as shown in Fig 4(a) compared with Fig. 3(a). We also observed slightly improved results in the random forest method by including the height and width features. Please note that the logistic regression performed better than the random forest model in terms of both precision and recall values.

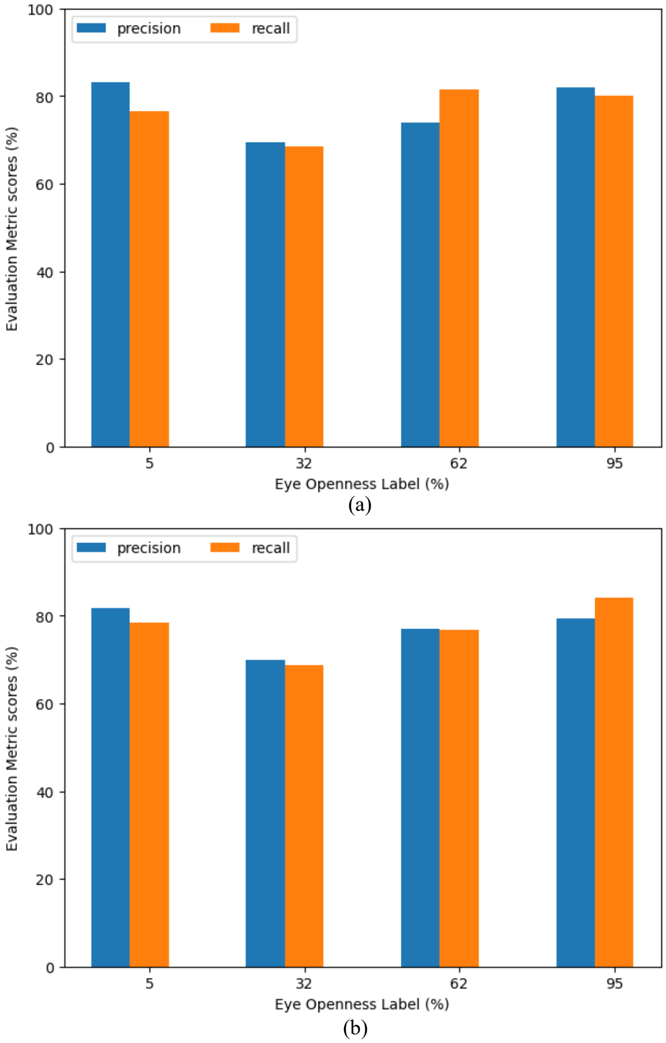

Figure 3: (a) Precision and recall scores of the logistic regression (b) and random forest models without the height and width features.

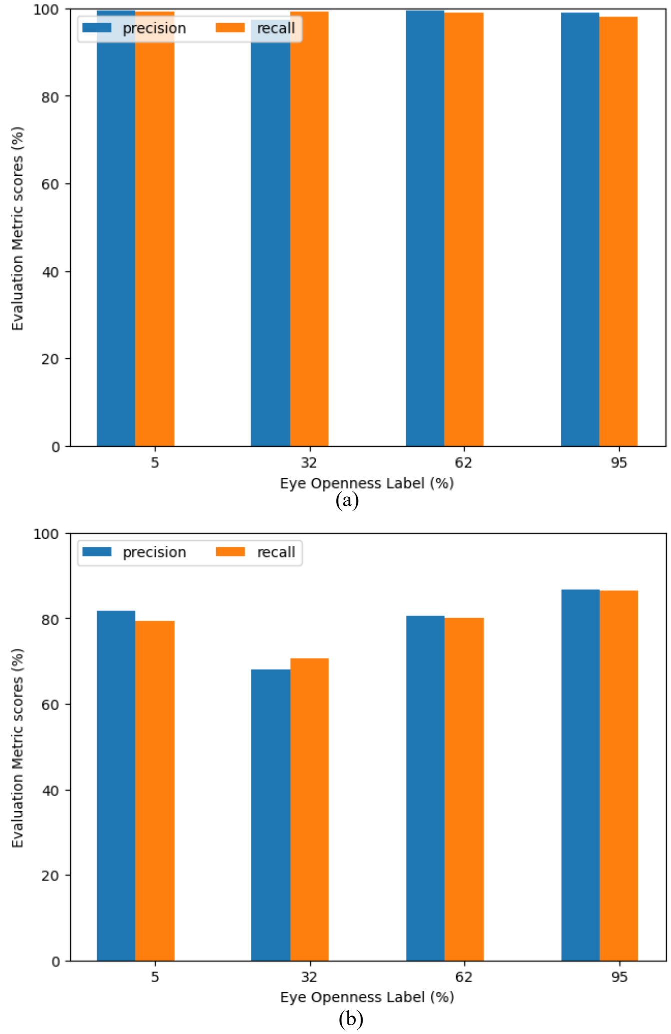

Figure 4: (a) Precision and recall scores of the logistic regression and (b) random forest models with the height and width features.

Fig. 5 shows both the prediction accuracy per error threshold and the scatterplot between the actual and predicted labels for the second set of experiments. Fig. 5(a) compares the actual and predicted values on a scatterplot where most of the values are very close to each other, and they located around the diagonal axis. The results were mostly correlated with minor outliers. To quantify the error between actual and predicted values, we calculated the prediction accuracy per error threshold as shown in Fig. 5(b). The prediction accuracy reached 80% when the error threshold was approximately 5%, and 90% when the error threshold was at 10%.

Fig. 6 shows the mean and standard deviation values of the predicted labels per actual labels. Please note that there were some outliers in the predicted labels on mean and standard deviation plots compared to the actual labels. However, standard deviation values are mostly around 2-5, and the mean plot follows the diagonal axis. This shows the results are consistent with different eye blink levels.

The first approach suggests that generating new features regarding the dimensions of the subject's eyes allow both the random forest and logistic regression models to predict eye openness more accurately. Since these features are expressed as a ratio, the models are not sensitive to outliers regarding the size of the subject's eyes. Please note that to calculate these features, the subject's facial landmarks must be extracted by the MediaPipe model first. Therefore, it is feasible to classify eye openness on limited hardware under this proposed methodology.

On the other hand, the second approach suggests that AlexNet can effectively calculate the subject's eye openness without relying on using the facial landmarks points as features. The reason behind these results may be due to the architecture of AlexNet. Having multiple hidden layers along with convolutional filters results the model in extracting more relevant features than the logistic and linear regression models under the first methodology. Therefore, the classification under the AlexNet model is more accurate than the random forest model from the first methodology.

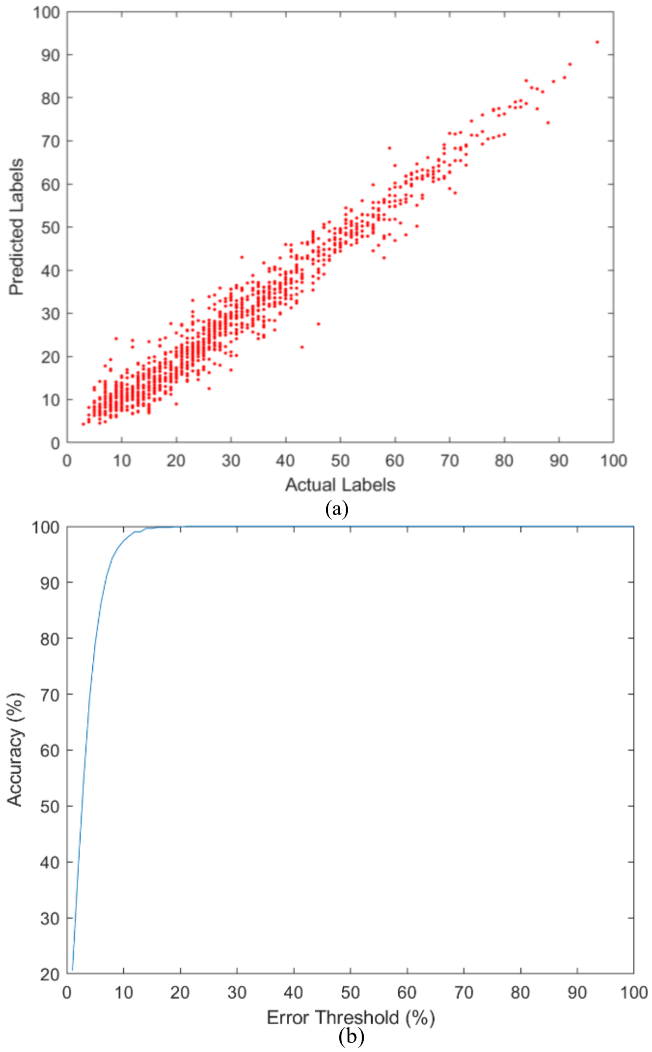

Figure 5: Prediction accuracy per error threshold (a) and the scatterplot of Actual Labels vs. Predicted Labels (b).

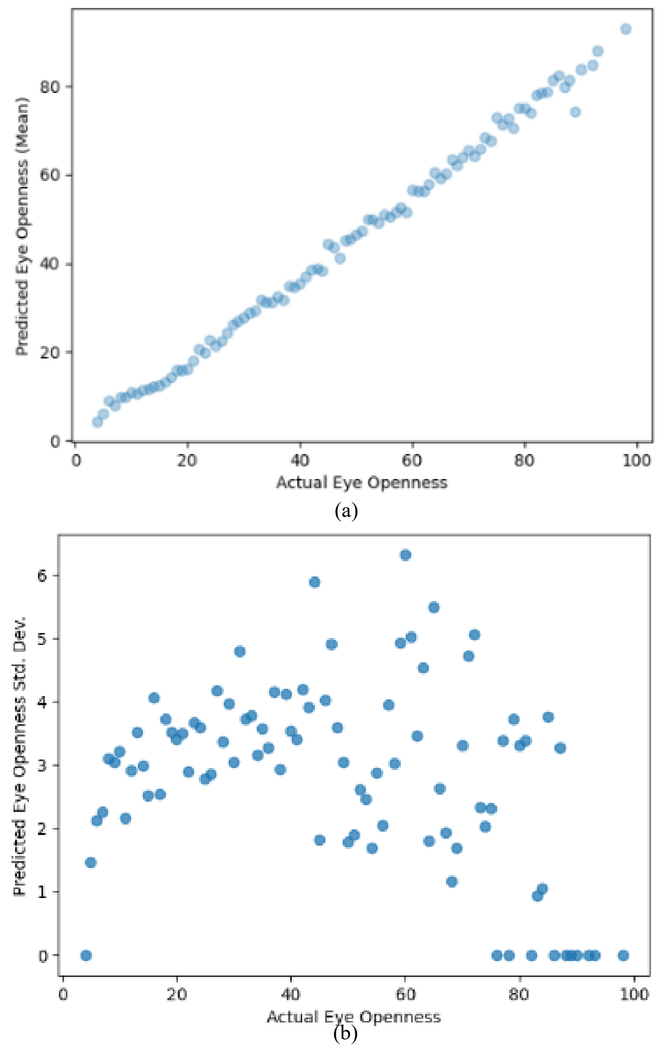

Figure 6: (a) Prediction accuracy per error threshold and (b) the scatterplot of Actual Eye Openness vs. the Standard Deviation of Predicted Labels.

V. Conclusion

In this research, we conducted a comparative analysis of two distinct methodologies for eye blink detection algorithms. Firstly, logistic regression and random forest models were trained on extracted facial landmark features. Secondly, a CNN-regression-based blink detection model was trained directly on input images. The initial phase of the experiment involved training multiple CNN-based models with varied parameter values. Subsequently, a new dataset was generated using eye landmark features extracted through a face detection model, and these features were employed to train random forest and logistic regression models. The random forest model, trained on the facial landmark dataset, exhibited a precision score of at least 75% for each eye openness value but demonstrated a lower score for images where the subject had blinked. Conversely, the AlexNet blink detection model outperformed the logistic regression and random forest models in terms of classification performance. This superiority is attributed to AlexNet's more effective feature extraction capabilities compared to logistic regression and random forest models. It is noteworthy that the random forest and logistic regression models can be optimized for deployment on iris recognition hardware with limited computing resources. In contrast, AlexNet, owing to its enhanced performance, is more suited for utilization on high-performance computers. These models hold significance not only for blink detection but also for broader applications, including the more accurate identification of subjects based on the iris texture.

Acknowledgments

This project was made possible by support from Secure and Trustworthy Cyberspace (SaTC) program of The National Science Foundation (NSF) under grant award CNS-1909276 and CNS-2100483.

VI. References

[1] M. Karakaya. A study of how gaze angle affects the performance of iris recognition. Journal of Pattern Recognition Letters, 82(2):132-143, 2016.

[2] H. Proenca, and L. Alexandre. Toward noncooperative iris recognition: A classification approach using multiple signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(4):607 –612, 2007.

[3] S. Pundlik, D. Woodard, and S. Birchfield. Non-ideal iris segmentation using graph cuts. In Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference on, pages 1 –6, June 2008.

[4] Y. Si, J. Mei, and H. Gao. Novel approaches to improve robustness, accuracy and rapidity of iris recognition systems. IEEE Trans. on Industrial Informatics, 8(1):110 –117, 2012.

[5] J. Daugman. New methods in iris recognition. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 37(5), 1167-1175, 2007.

[6] S.A.C. Schuckers, N.A. Schmid, A. Abhyankar, V. Dorairaj, C.K. Boyce, and L.A. Hornak. On Techniques for Angle Compensation in Nonideal Iris Recognition. IEEE Transactions on Systems, Man, and Cybernetics, 37(5), pp.1176-1190, 2007.

[7] H. Palta, and M. Karakaya. Blink Detection for Off-Angle Iris Recognition using Deep Learning. In Pattern Recognition and Tracking XXXIV, SPIE Defense + Commercial Sensing, pages 52-59, 2023.

[8] D. Chavarro, and M. Karakaya. Off-Angle Iris Recognition using Single and Multiple Deep Learning Frameworks. In Proceedings of Multimodal Image Exploitation and Learning 2023, SPIE Defense + Commerical Sensing, Volume 12526, 2023.

[9] F. Safarov, F. Akhmedov, A. B. Abdusalomov, R. Nasimov, and Y. I. Cho. Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety. Sensors, 23(14), pp. 6459, 2023.

[10] Saealal MS, Ibrahim MZ, Mulvaney DJ, Shapiai MI, Fadilah N (2022) Using cascade CNN-LSTM-FCNs to identify AI-altered video based on eye state sequence. PLoS ONE 17(12): e0278989.

[11] G. d. l. Cruz, M. Lira, O. Luaces and B. Remeseiro, "Eye-LRCN: A Long-Term Recurrent Convolutional Network for Eye Blink Completeness Detection," in IEEE Transactions on Neural Networks and Learning Systems, 2022.

[12] Fogelton, A., & Benesova, W. (2016). Eye blink detection based on motion vectors analysis. Computer Vision and Image Understanding, 148, 23–33.

[13] Fogelton, A., & Benesova, W. (2018). Eye blink completeness detection. Computer Vision and Image Understanding, 176, 78–85.

[14] Medeiros, Silva, Fernandes, Sánchez-Gendriz, Lins, Barros, Nagem, Valentim, "Efficient machine learning approach for volunteer eye-blink detection in real-time using webcam", Expert Systems with Applications". Volume 188, February 2022, 116073.

[15] Ayudhya C.D.N. and Srinark T., "A Method for Real-Time Eye Blink Detection and Its Application" (2009).

[16] Han Y.J. , Kim W., and Park J.S., "Efficient Eye-Blinking Detection on Smartphones: A Hybrid Approach Based on Deep Learning", Hindawi, Mobile Information Systems, Article ID 6929762 (2018).

[17] F. M. Sigit, E. M. Yuniarno, R. F. Rachmadi and A. Zaini, "Blinking Eyes Detection using Convolutional Neural Network on Video Data," 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2020, pp. 291-296.

[18] Vishesh P., Raghavendra S, Jankatti S.K , Rekha V, "Eye blink detection using CNN to detect drowsiness level in drivers for road safety", Indonesian Journal of Electrical Engineering and Computer Science, Vol 22, No 1, April 2021, pp. 222~231.

[19] Scikit Learn. "scikit-learn: machine learning in Python." Scikit Learn. https://scikit-learn.org/stable/.

[20] Scikit Learn. "sklearn.preprocessing.StandardScaler." Scikit Learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.