

Eye blink detection using CNN to detect drowsiness level in drivers for road safety

Pothuraju Vishesh¹, Raghavendra S², Santosh Kumar Jankatti³, Rekha V⁴

^{1,2,4}Department of Computer Science and Engineering, CHRIST Deemed to be University, India

³Department of Computer Science and Engineering, KSSEM, VTU, India

Article Info

Article history:

Received Oct 30, 2020

Revised Jan 17, 2021

Accepted Mar 3, 2021

Keywords:

Convolution neural network

Cross entropy

Deep learning

Dlib

Eye blink

Mobilenet V2

RMSProp

ABSTRACT

Blinking is a regular bodily function and it is the semiautomatic fast closing of the eyelid. A specific blink is examined by dynamic folding of the eyelid. It is a vital function of the eye which helps in spread of tears across and eliminates irritants from the shallow of cornea. In this research work we made use of convolution neural network, the deep learning concepts and image processing to detect drowsiness level in drivers. To train the blink detection model the mobilenet V2 is used as base. The loss function used for training was RMSprop and the optimizer is binary cross entropy. The dlib facial landmark was exploited to perceive and pre-process the detected faces. The dataset used for the training model is selected from the "Xiaoyang Tan" of nanjing university of aeronautics and astronautics. Based on the experimental outcome the projected method achieves an accuracy of 97%. The prototype developed serves as a base for further development of this process to achieve better road safety.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Raghavendra S

Department of Computer Science and Engineering

School of Engineering and Technology

CHRIST Deemed to be University

Mysore Road, Bengaluru, Karnataka, India

Email: raghav.trg@gmail.com

1. INTRODUCTION

According to WHO statistics, millions of peoples are losing their valuable lives each day. Numbers emphasize that the maximum of the lethal collisions are anticipated to driver's exhaustion and negligence. Based on the statistics of the association of the American Automobiles it was found that 7% of the collisions and 21% of the lethal traffic collisions are caused by exhausted drivers [1].

Computer vision (CV) is an integrated arena that demonstrates how processors might be used to acquire advanced understanding from videos/images. It is concerned with spontaneous mining, investigation, and demonstration of beneficial material from a distinct copy of an image or a series of images. Deep learning approaches like convolution neural networks (CNNs) are employed to identify drowsiness [2-5].

Eye Blinks are generally accompanied by the desire to cleanse the eyes by removing dust and to extend tear skin. So the eye blinks take place by dampness, heat, organic influences, and air components [6, 7]. The period amongst blinks upsurges when a person views a pictorial demonstration unit [8] or is abstracted by a challenging task. The period amongst blinks is conveyed to diminution with an aural job throughout driving as associated with driving without a supplementary job [9]. This recommends an affiliation amongst the category of the job and its consequence on the driver's blinking occurrence.

The MobileNetV2 structural design is grounded on a reversed outstanding structure in which the input and output are thin layers as opposed to old-style residual copies which make use of extended illustrations in the input. MobileNetV2 uses trivial depth-wise complications to categorize attributes in the in-between extension layers [10]. RMSProp employs the concept of the exponential biased mean of all the inclines analogous to gradient descent through motion. Here 'b' means the 'bias' showing perpendicular fluctuations and 'W' is the 'weight' showing the progress in flat orders [11].

In the proposed work, we do eye blink detection using CNN for perceiving the drowsiness level in drivers for road safety. The prototype developed serves as a base for further development of this process to achieve better road safety. Various techniques are employed for drowsiness detection and are deliberated in the ensuing segment of the Literature Review.

2. LITERATURE REVIEW

A method to detect drowsiness based on obtaining blood volume pulsation, eyelid flickering, and yawning signals was implemented and the method achieves the accuracy of 90.3%, 94.4%, and 93.7% for a yawn, blood volume pressure, and eyelid flickering respectively based on negative predictive rate indicator [12]. A new methodology termed DriCare was proposed, which perceives the drivers' weariness position identical to yawn, blink, and extent of eye closing of user's pictures. The investigational consequences exhibited that the new system accomplished approximately 92% exactness [13].

The eye blink counts are used as an important parameter for drowsiness detection of the driver. The proposed framework also continuously analyses the eye movement of the driver and alerts the driver [14]. The facial landmark detection (FLD) along with eye aspect ratio (EAR) is exercised to detect eye blink. The time of the closed state of the eye is the main parameter to decide the drowsiness. The proposed method achieved accuracy in an eye blink pattern of 92.7% [15].

A facial landmark blink detector (FLBD) and multi-layer perceptron (MLP) composited with various hidden layers, neurons, and activation functions are used as the classifier for detecting the eye blink [16]. The algorithm that mainly analyses the eye blink pattern (EBP) and mean eye landmarks' distance (MELD) of the drivers to spot the sleepiness of the driver was implemented. The duration of time of eye closure is considered to regulate the drowsiness condition. The method achieves an accuracy of 93% [17]. In each section, the pixel's worth is calculated according to the pixel fluctuation ratio. The model is trained with carefully chosen training trials along with SVM and achieved good results [18]. The challenges faced were to discriminate amongst eye blink trials and gaze-related closing of the eyelid, mainly the glimpses towards the dashboard. This used the threshold for the maximum velocity of the eyelids for discrimination [19].

A driver fatigue recognition scheme in real-time grounded on the SVM algorithm is projected. Fatigue discovery chiefly emphasizes drivers' facial appearances and behaviors [20]. The model that employs the infrared videos aimed at perceiving and a method using CNN aimed at identifying eye state was projected. The model ultimately calculates the proportion of eyelid closing concerning the pupil with time (PERCLOS) and the rate of blink to perceive the weariness [21]. An efficient method for detecting the eye-blink is CNN and SVM. The main aim was to detect the eye-blink reliably. From the experimental results, it was found that the precision was 94.4% [22]. The use of eye-blink sensors for the detection of eye-blink during driving was developed. In this research, the speed will be reduced or the vehicle will be stopped. Along with this, the owner will be informed. By this, the driver and the owner will be notified and can avoid further consequences [23]. To recognize the mental attention level of the driver the system makes use of a brain-computer interface along with two sensors. A microcontroller is used to find the difference between the eye blink of a normal person with the drowsiness person. A blinking LED is used to alert the driver [24].

Datasets are key when working out a DNN. So for improving the performance of drowsiness detection for varying cultural assemblies a framework based on CNN and training based on generative adversarial networks (GAN) was proposed. This was primarily directed on data extension grounded on the public bias conception approach that clusters appearances with comparable facial characteristics. The proposed framework resulted in an accuracy of 96.75% and 91.62% for training and testing respectively [25].

The crucial hypothesis was that the sitting position correlated index can signpost feeble drowsiness, which the drivers cannot feel. At first, the sensitivity of the sitting position and conventional indices are determined and a method that can detect the drowsiness was projected by combining the various indices related to the sensitivity of the feeble and the robust drowsiness. The projected method improved the accuracy of the feeble drowsiness detection with an RMSE of 62% [26].

A method based on multiple feature fusion was developed by combining the driver's attributes where YOLOv3 CNN was used to take the facial areas beneath difficult driving circumstances. Based on the dlib toolkit, EFU and MFU are used as assessment limitations for the eye state and mouth state of the driver

correspondingly. A library was built for the identification of the driver with information such as eye state, mouth state, and driver biometric. At last, the verification model for the driver was built along with the fatigue valuation model. Using virtual applications, the method was capable of detecting the fatigue state with an accuracy of 95.10% [27].

A new technique to notice Deepfakes using the GANs model through the DeepVision algorithm was developed. DeepVision was capable of analyzing any major modification in the blinking pattern. The blinking pattern of human's changes from one to another and this is mainly based on biological features, physical circumstances, cognitive actions, etc. Using a heuristic model, any changes in the eye blink patterns can be determined using Deepfakes. DeepVision was capable of detecting the Deepfakes with an accuracy of 87.5% [28].

EEG signal objects lead to some complications in the investigation and management of EEG signals. Even though many methods are available to identify these objects, the accuracy achieved by this is not satisfactory. So, there is a requirement for developing new methods for improving accuracy. To detect the eye blink objects in EEG signals, the semantic segmentation methods are used and the classification accuracy achieved was found to be 94.4% [29].

One of the major sources for the lethal road accident is the driver's drowsiness. One of the solutions to avoid road accidents is automatic drowsiness detection and is considered a gifted solution. An embedded CNN based solution was proposed where the eye blink is detected through smart connected glasses. From the experimental results, it was found that CNN gives better results when compared to the algorithms based on the threshold method and the accuracy achieved was 90.8% [30].

A lot of characteristics such as yawning, movement of the head, eye blinks can be extracted from the face that can be used to detect the amount of drowsiness. But developing a system that can detect this and that can yield consistent and exact outcomes is an interesting task and it needs precise and robust methods. An analysis, of many techniques used by different researchers for drowsiness detection, was done and from the analysis, it was identified that SVM is the technique that is used most often, but the CNNs gave the best results compared to other methods [31]. Even though many techniques have been proposed to date and there is substantial scope to progress in the detection systems, the main challenge here is that different researchers use diverse datasets and the outcomes cannot be straightforwardly assessed. And also the datasets utilized are found to be inadequate and are collected from the controlled environment and may fail in real-world circumstances. In our proposed work the dataset used for the training model is selected from the "Xiaoyang Tan" of Nanjing University of Aeronautics and Astronautics. With this proposed research work, an effort is made to better the accurateness of the forecast.

3. FRAMEWORK OF THE PROPOSED WORK

This work is about developing an artificial intelligence (AI) model that is capable of identifying a person and telling whether his/her eyes are open or close. The model developed is a deep learning model and should take the images as input and result in an output specifying the eyes in the image are open or closed. This work aims to provide a solution for eye blink detection in automotive safety. The approach followed in this work follows retraining on the previously build models, to utilize the metrics of those models to achieve better performance. The general flow of CNN is given in Figure 1.

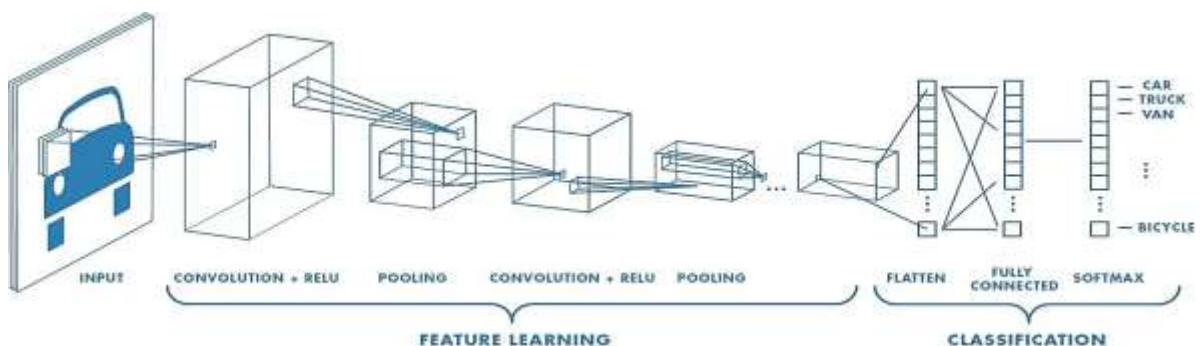


Figure 1. Eye blink detection model

The basic components mentioned in Figure 1 consists of the operations as follows:

- The convolution step

To mine, the characteristics of the entered image ConvNet are employed. Convolution gives the spatial association among pixels by understanding image characteristics employing small shapes of entered facts. The Convolution phase is shown in Figure 2.

- Design (optimizer, loss function and layers)

CNN consists of an input layer, an output layer, and various hidden layers. Each sequence of layers in the CNNs hidden layers ‘Convolve’ with exponentiation. ReLU is the very frequently employed activation function in CNN and is shown in Figure 3. Activation functions are mathematical estimates that regulate the CNNs output.

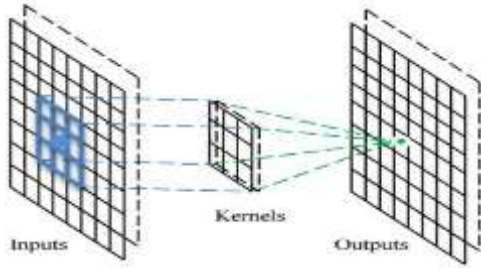


Figure 2. Convolution step

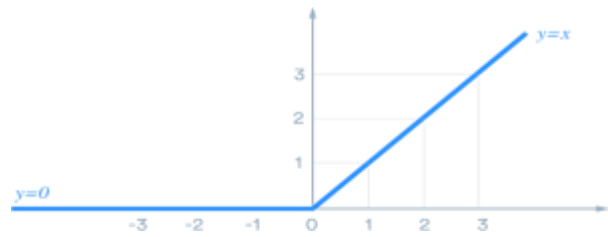


Figure 3. ReLU activation function

- Optimizer

Optimizers are methods that are employed to transform the characteristics of any CNN in terms of learning proportions and weights to lessen the shortfalls. The feature extraction layer of this work makes use of RMSProp. Optimization policies are accountable for dropping the losses and to deliver the utmost precise consequences possible. The RMSProp uses dW and db for each epoch and it considers the exponentially weighted means of squares of dW and db as shown in (1) and (2):

$$S_{dw} = \beta \cdot S_{dw} + (1 - \beta) \cdot dW^2 \tag{1}$$

$$S_{db} = \beta \cdot S_{db} + (1 - \beta) \cdot db^2 \tag{2}$$

Here, β varies from 0 to 1 and is a hyperparameter.

It uses the weighted mean of previous estimates and the squares of the present estimates to figure out the future weighted mean and the parameters are updated using (3) and (4):

$$W_{i+1} = W_i - \frac{\text{learning_rate} \cdot dW^2}{\sqrt{S_{dw}}} \tag{3}$$

$$b_{i+1} = b_i - \frac{\text{learning_rate} \cdot db^2}{\sqrt{S_{db}}} \tag{4}$$

- Loss function

Loss function supports improving the limitations of the CNNs by calculating the loss. This work makes use of Binary cross-entropy for its binary nature, it measures the prediction from the true value (which is either 0 or 1) is for every class and then finds the mean of these class-wise faults to acquire the ultimate loss. The binary cross-entropy loss is defined using (5).

$$CE = \sum_{i=1}^{C'} t_i \log(f(s_i)) = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1)) \tag{5}$$

Here, $C' = 2$, C_1 , and C_2 are two classes assumed, t_1 and s_1 are the value for C_1 , $t_2=1 - t_1$, and $s_2=1 - s_1$ is the value of C_2 . The loss can be represented as shown in (6).

$$CE = \begin{cases} -\log(f(s_1)) & \text{if } t_1 = 1 \\ -\log(1 - f(s_1)) & \text{if } t_1 = 0 \end{cases} \tag{6}$$

– Pooling

CNN may comprise general or universal pooling layers to rationalize the fundamental calculation. Pooling layers diminish the magnitudes of the facts by merging the yields of neuron groups at one layer with particular neuron in the subsequent layer. Generic pooling unites minor clusters, naturally 2x2. Universal pooling proceeds on every other neuron in the convolutional layer. Pooling might calculate a maximum or a mean. The pooling phase is shown in Figure 4.

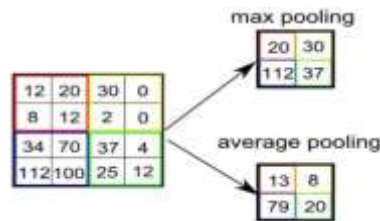


Figure 4. Pooling step

– Fully connected layer

This is typically the layer in any CNN upon which the Flatten and Dense layers are applied. The pooled features are transformed into a specific column by the Flatten function. This column is dispatched to the fully connected layer. The fully connected layer is inserted into CNN with the help of Dense. A fully connected layer links each neuron in a single layer with each other neuron in the next layer and is similar to MLP. The flattened matrix moves around this layer to categorize the images.

4. DESIGN AND IMPLEMENTATION

CNN's are used to implement this projected work. CNN's are versions of MLPs and uses fully connected layer. The Design of this work is given briefly in the high-level design diagram in Figure 5, where the loading of the current dataset (closed eyes in the wild – (CEW)) and adding of feature extraction layer on the CNN are mentioned.

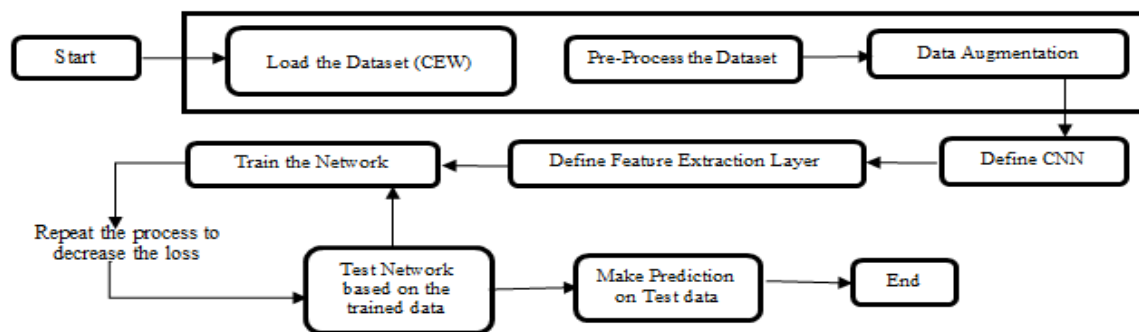


Figure 5. High level diagram

The steps followed are:

Step 1: Start.

Step 2: Load the Dataset (closed eyes in the wild).

- In this work, we have employed the open dataset published by 'Xiaoyang Tan' from Nanjing University of Aeronautics and Astronautics. A sample of the images belonging to the set closed eyes or open eyes is shown in Figure 6 and Figure 7.

Step 3: Resize and Reshape every image in the Dataset.

Step 4: Rotate and flip every image for better performance during training.

Step 5: Select the model and architecture according to the problem statement.

- The CNN used for this model is 'MobileNetV2' and the output of the model is binary i.e. eyes closed or eyes open.

Step 6: Write the top Feature Extraction Layer to detect the state of the eyes.

Step 7: Training

- The last layers are fully connected network layers followed by ‘softmax regression’ for classification in the output layer. The weights in the top layers of the pre-trained prototypes are adjusted further to upsurge the functioning. This will adjust the weights to associate with the feature related to the dataset.

Step 8: Testing the model and tuning of hyper-parameters. Repeat step 7.

- The fine-tuning approach was employed in training the model. CNN’s are similar to normal NN. They consist of neurons with learnable biases and weights. Each neuron takes the data and does the dot product. The complete network gives one differential score function and still has a loss function.

Step 9: Make Predictions on completely new unseen data.

Step 10: End. Save the model and deploy it.



Figure 6. Closed eyes Figure 7. Open eyes

5. PROTOTYPING AND TESTING

The testing phase of the project was initially focused on developing a mobile app to test the model trained. Finally, a webcam-based prototype was developed to test the model trained for the problem statement, using dlib facial landmarks to detect faces in the video stream and then pre-process the detected face for inference with the model. The results after the training show graphs about the loss and accurateness details throughout the training and validation phase.

5.1. The webcam approach

The webcam approach takes the live video input from the webcam and was used for testing the model. Since the model was trained with only images of face data with eyes closed or open, the model can only inference on face images. To detect and crop faces from the input stream, dlib facial detector was used to detect faces in the input feel. The dlib facial detector works on the facial landmarks. The dlib facial landmark is a collection of points and is used to detect various parts of the face using 68 points on the face. These facial landmarks used by dlib facial detector are given in Figure 8 showing how the points make us the face of any human. The whole flow of the demo application is given in Figure 9.

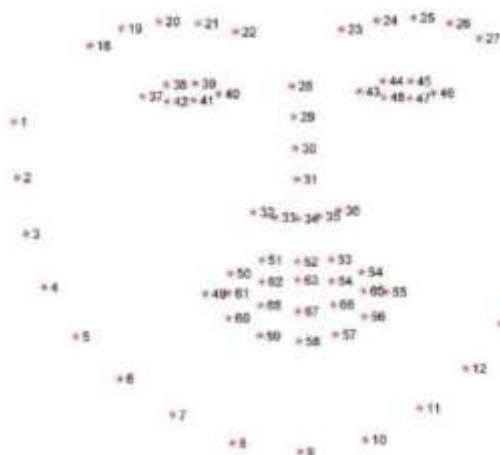


Figure 8. 68-point facial landmarks

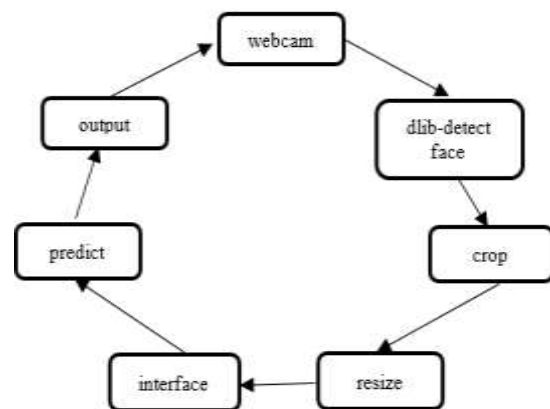


Figure 9. Flow of demo application

The step followed are:

- Webcam: any webcam with a picture quality of 720p or greater.
- dlib: the facial detector to detect faces in the webcam feed, to locate the face in the wild.
- Crop: The location of faces is obtained from dlib facial detector and is cropped consisting of only the face and nothing else.
- Resize: The model is more efficient to process images of dimensions 244*244, so the image of the face is resized accordingly whatever the size maybe.
- Inference: The image is converted into tensors and inferences ready for prediction.
- Predict: The image is passed for prediction with a model trained for EBD.
- Output: The output given out by the model is translated into open or close and displayed on the screen.

6. RESULTS AND DISCUSSION

Results obtained during the training of the network is given in Figure 10. The accuracy mentioned here is concerning the validation data given during the training time and is subject only to the data in the dataset. The output of the partially developed mobile app is given in Figure 11. The faces are detected by Firebase Vision ML and then cropped and resized according to the model and then inference.

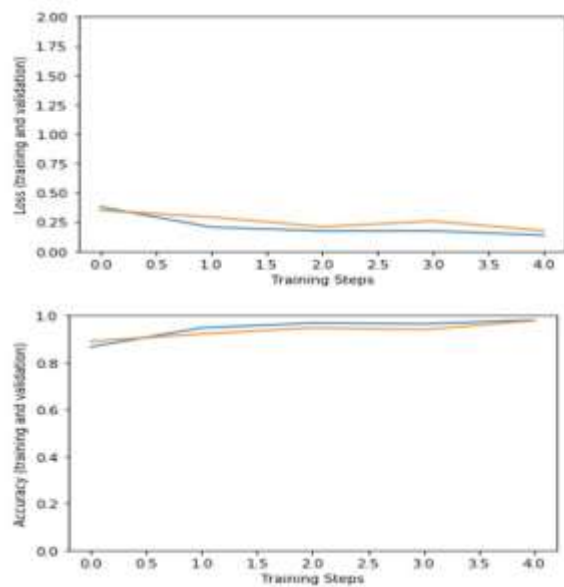


Figure 10. Accuracy and loss during training and validation



Figure 11. Object detection App

The android app was experimented with using various approaches using Android's firebase vision face detector to detect and track faces.

- The image classification approach: This approach used the normal classification model for classifying the detected faces as open or close. But the obstacle was that the app was not able to detect and crop face before running inference on the image. The issue that was identified was related to synchronous and asynchronous ways of executing tasks.
- The object detection approach: The object detection approach follows a totally new way of approaching the problem statement. The research in this field is still in the process. Where the pre-processing of the image to detect faces is not required. The training process itself learns to detect the faces in the wild and then run inference on the detected faces.
- Wink detection: The model was not trained with images of subjects with one eye closed and the other eye open. A data-set of a total of 400 images from different angles consisting of two categories was made. Firstly, the left eye closed and the right eye open. Secondly, the left eye open and the right eye closed. The Hardware needed was not available to carry out the experiment. So, the use-case was suspended from work.

The number of images and the selection of hyper-parameters play a vital part in deciding the accuracy of the model. The results of using different hyper-parameters are given below;

- Dataset used: closed eyes in the wild (CEW)
- No of images: 1194(closed eyes), 1239(open eyes)
- Total: 2433images (100X100) JPG
- Total no. of epochs: 5
- Training images: 2181
- Testing images: 252

The process of improving the model performance and nearing the objective is given in the table below in Table 1 to show the result obtained during training. Figure 12 displays the assessment of the projected work with some of the prevailing work mentioned in the literature survey.

Table 1. The Results obtained during training on the dataset (CEW)

Sl. No	Model	Training Time	Optimizer	Accuracy
1	3 Layer (Sequential)	1 mins per epoch	Adam	34%
2	3 Layer (Sequential)	1.3 mins per epoch	Adam	57%
3	2 Layer (Sequential)	1 mins per epoch	Adam	60%
4	MobileNetV2	8 mins per epoch	Adam	81%
5	MobileNetV2	8.3 mins per epoch	RMSProp	95%
6	MobileNetV2	8.7 mins per epoch	RMSProp Loss: binary cross entropy	97%
7	Inception	40 mins per epoch	RMSProp	82%

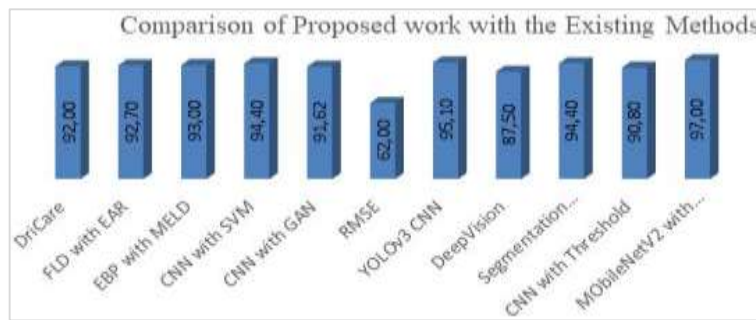


Figure 12. Comparison of proposed work with the projected work

7. CONCLUSION

The eye blink detection work finally served its purpose of detecting the state of the eyes in a given image. This work aims to provide a solution for reducing road accidents caused by human error. Many eye blink detection applications developed are serving this purpose, this is a better model to help improve the performance of the detection rate and identify possible threats. This work provides a solution by implementing the retraining of the Google MobileNetV2 model to identify eye blink. This work aims to make the world a safer place on roads for travelers and the total accuracy achieved was 97%. In future work, we try to concentrate on detecting the drowsiness in the faces with sunglasses, only one eye closed, and side view.

ACKNOWLEDGEMENTS

I would like to thank CHRIST (Deemed to be University), Benagluru in supporting us by providing the environment to get the work done.

REFERENCES

[1] B. C. Tefft, "Acute Sleep Deprivation and Risk of Motor," *Vehicle Crash Involvement*, 2016.
 [2] D. H. Ballard and Christopher M. Brown, "Computer Vision," *Prentice Hall*, 1982.
 [3] T. Huang, *et al.*, "Computer Vision: Evolution and Promise," *19th CERN School of Computing. Geneva: CER*, pp. 21-25, 1996, doi: 10.5170/CERN-1996-008.21.
 [4] M. Sonka, *et al.*, "Image Processing, Analysis and Machine Vision," Thomson. 2008.

- [5] F. You, *et al.*, "A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration," *Special Section on Artificial Intelligence (ai)-empowered Intelligent Transportation Systems, IEEE Access*, vol. 7, pp. 179396-179408, Dec 2019, doi: 10.1109/ACCESS.2019.2958667.
- [6] J. A. Stern, *et al.*, "Blink rate: A possible Measure of Fatigue," *Human Factors*, vol. 36, pp. 285-297, 1994, doi: 10.1177/001872089403600209.
- [7] P. Wolkoff, *et al.*, "Eye Complaints in the Office Environment: Precorneal Tear Film Integrity Influenced by Eye Blinking Efficiency," *Occupational and Environmental Medicine*, vol. 62, pp. 4-12, 2005, doi: 10.1136/oem.2004.016030.
- [8] S. Patel, *et al.*, "Effect of Visual Display Unit Use on Blink Rate and Tear Stability," *Optometry and Vision Science*, vol. 68, pp. 888-892, 1991, doi: 10.1097/00006324-199111000-00010.
- [9] Y. F. Tsai, *et al.*, "Task Performance and Eye Activity: Predicting Behavior Relating to Cognitive Workload," *Aviation, Space, and Environmental Medicine*, vol. 78, pp. B176-B185, 2007.
- [10] M. Sandler, *et al.*, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.
- [11] S. Vani and T. V. M. Rao, "An Experimental Approach Towards the Performance Assessment of Various Optimizers on Convolutional Neural Network," *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 331-336, doi: 10.1109/ICOEI.2019.8862686.
- [12] C. Zhang, *et al.*, "Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications," *IEEE Access*, vol. 7, pp. 11829-11843, 2019, doi: 10.1109/ACCESS.2019.2891971.
- [13] W. Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features," *IEEE Access*, vol. 7, pp. 118727-118738, 2019, doi: 10.1109/ACCESS.2019.2936663.
- [14] Fouzia, *et al.*, "Driver Drowsiness Detection System Based on Visual Features," *2nd international Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 1344-1347, doi: 10.1109/ICICCT.2018.8473203.
- [15] A. Islam, *et al.*, "A Study on Tiredness Assessment by Using Eye Blink Detection," *Journal of Engineering-UKM*, vol. 31, no. 2, pp. 209-214, 2019.
- [16] C. Chou, *et al.*, "Blink Detection Using Facial Landmarks Blink Detector and Multi-Layer Perceptron," *National Computer Symposium*, 2019, <http://dx.doi.org/10.6927%2fNCS.201911.0105>.
- [17] A. A. Miah, *et al.*, "Drowsiness Detection Using Eye-Blink Pattern and Mean Eye Landmarks' Distance," *International Joint Conference on Computational Intelligence, Springer*, 2019, pp. 111-121.
- [18] D. Dou and Z. Zhang, "Blink Detection Based on Pixel Fluctuation Ratio of Eye Image," *Journal of Physics: Conference Series*, vol. 1463, no. 012073, 2020, pp. 1-5, doi: 10.1088/1742-6596/2F1453/2F1%2F012073.
- [19] M. H. Bacour, *et al.*, "Camera-Based Eye Blink Detection Algorithm for Assessing Driver Drowsiness," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 987-993, 2019, doi: 10.1109/IVS.2019.8813871.
- [20] B. K. Savas, and Y. Becerikli, "Real Time Driver Fatigue Detection Based on SVM Algorithm," *6th International Conference on Control Engineering and Information Technology (CEIT)*, 2019, doi: 10.1109/CEIT.2018.8751886.
- [21] F. Zhang, *et al.*, "Driver Fatigue Detection Based on Eye State Recognition," *International Conference on Machine Vision and Information Technology (CMVIT)*, 2017, pp. 105-110, doi: 0.1109/CMVIT.2017.25.
- [22] Y. J. Han, *et al.*, "Efficient Eye-blink Detection Using Smartphones: A hybrid Approach Based on Deep Learning," *Mobile Information System, Hindawi*, vol. 2018, pp. 1-8, 2018, doi: 10.1155/2018/6929762.
- [23] B. M. K. Kumari, *et al.*, "Detection of Driver Drowsiness Using Eye Blink Sensor," *International Journal of Engineering and Technology*, vol. 7, no. 3.12, pp. 498-504, 2018, doi: 10.14419/ijet.v7i3.12.16167.
- [24] O. Sinha, *et al.*, "Development of a Drowsy Driver Detection System Based on EEG and IR-Based Eye Blink Detection Analysis," *Advances in Communication, Devices and Networking*, vol. 462, pp. 313-319, 2018, doi: 10.1007/978-981-10-7901-6_34.
- [25] M. Ngxande, *et al.*, "Bias Remediation in Driver Drowsiness Detection Systems Using Generative Adversarial Networks," *IEEE Access*, vol. 8, pp. 55592-55601, 2020, doi: 10.1109/ACCESS.2020.2981912.
- [26] M. Sunagawa, *et al.*, "Comprehensive Drowsiness Level Detection Model Combining Multimodal Information," *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3709-3717, April 2020, doi: 10.1109/JSEN.2019.2960158.
- [27] K. Li, *et al.*, "A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion," *IEEE Access*, vol. 8, pp. 101244-101259, June 2020, doi: 10.1109/ACCESS.2020.2998363.
- [28] T. Jung, *et al.*, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," *IEEE Access*, vol. 8, pp. 83144-83154, April 2020, <https://doi.org/10.1109/ACCESS.2020.2988660>.
- [29] M. Tosun and Ö. Kasım, "Novel eye-blink artefact detection algorithm from raw EEG signals using FCN-based semantic segmentation method," *IET Signal Processing*, vol. 14, no. 8, pp. 489-494, October 2020, doi: 10.1049/iet-spr.2019.0602.
- [30] A. A. Jordan, *et al.*, "Deep Learning for Eye Blink Detection Implemented at the Edge," *IEEE Embedded Systems Letters*, pp. 1-1, October 2020, doi: 10.1109/LES.2020.3029313.
- [31] M. Ngxande, *et al.*, "Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques," *Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pp. 156-161, 2017, doi: 10.1109/RoboMech.2017.8261140.

BIOGRAPHIES OF AUTHORS

Mr. Pothuraju Vishesh is currently studying in Final year B.Tech in Computer Science and Engineering in CHRIST Deemed to be University, Bengaluru. He completed Internship in LG Soft, India. His interests include AI, Data Mining and Data.



Dr. Raghavendra S is currently working as Associate Professor in the Department of Computer Science and Engineering at CHRIST Deemed to be University, Bengaluru. He completed his Ph.D. degree in Computer Science and Engineering from VTU, Belagavi, India in 2017 and has 15 years of teaching experience. His interests include AI, Data Mining.



Mr. Santosh Kumar J is currently working as Associate Professor in the department of Computer Science and engineering at KSSEM Bengaluru affiliated to VTU BELagavi. He pursuing Ph.D. In CSE at research centre BGSIT BG NAGAR VTU Belagavi and have 13 Years of experience and his area of interest are Big data Analytics.



Dr. Rekha V is currently working as Assistant Professor in the Department of Computer Science and Engineering at CHRIST Deemed to be University, Bangalore. She completed her Ph.D. degree in Computer Science and Engineering from VTU, Belagavi, India in 2017 and has 15 years of teaching experience. Her interests include Computer Networks and Cyber Security.