IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

LIVE DROWSINESS DETECTION USING IMAGE PROCESSING AND DEEP LEARNING

Mukesh Prasad, E.Raju, P. Malavika
Department of Computer Science and Engineering
St. Peter's Engineering College
Hyderabad, India

K. David Raju (Associate Professor)
Department of Computer Science and Engineering
St. Peter's Engineering College
Hyderabad, India

Abstract— **Driver fatigue and drowsiness are some of the prime reasons for road accidents around the globe. Anyone can become a victim of drowsiness while driving after tiring physical conditions, short periods of sleep, or during long journeys. Drowsiness or inactivity causes effects driving in these couple of major areas. It increases reaction time, loss of coordination which makes drivers respond late which may result in an occurrence of accidents, several injuries, may loss of people. Every year more than 1,000,000 police-reported crashes involve drowsy driving. The occurrence of road accidents has been increasing, with the rise in population. Due to this, various studies were done in designing systems that can examine inactivity and drowsiness or driver fatigue and alert him/her beforehand thus preventing them from falling asleep and causing an accident. Our proposed method is to develop a deep learning algorithm using Convolutional Neural Networks, Computer Vision to detect drowsiness of drivers. The trained model predicts the driver's condition and alerts him/her thus preventing them from falling asleep and pushing them to focus on driving.**

Keywords— Convolutional Neural Network; Open CV; Face Detection; Transfer Learning; Keras;

I. INTRODUCTION

Accidents caused due to drowsy driving are a major problem in many countries. Interaction between driver and vehicle such as supporting each other and monitoring is one of the important solutions for keeping ourselves safe in the vehicles. Several studies have been done to detect the drowsiness and fatigue of drivers considering various parameters. Some approaches implemented psychological Measures to monitor the drivers' fatigue status by recording psychological signals using either electroencephalogram(EEG), electromyography(EMG), or electrocardiography(ECG). However, such techniques are obtrusive or intruding since sensors are required to be placed on the drivers' bodies to collect the data. This may make the driver feel uncomfortable and also divert his/her attention from driving. Considering the above issues and with the advancement of Computer vision technologies, we have proposed a method to create an algorithm that detects drowsiness quickly and can alert the driver. We trained a neural network using CNNs and applying Transfer Learning to our model. Live images of drivers are captured and fed to the model that predicts the state of the driver and alerts him/her based on the conditions of the drivers.

II. IMPLEMENTATION

As proposed earlier, this recognition method uses eye-tracking and image processing. A robust eye detection algorithm is introduced to address the problems caused by changes in illumination and driver posture. The technologies we have used are Python, TensorFlow, Keras OpenCV.

A. Python

Python is our major programming language. It is used for creating our Neural Network model. It provides various tools and libraries that help in consistently creating our model.

B. TensorFlow

TensorFlow is one of the open-source libraries for machine learning and deep learning. Tensorflow is written on top of many programming languages. The one we used is built on top of Python. Tensorflow library provides simple and convenient pre-trained Neural Network models.

C. Keras

Keras is another open-source machine learning library that provides additional support to TensorFlow in our application. It provides a python interface and also supports multiple backends.

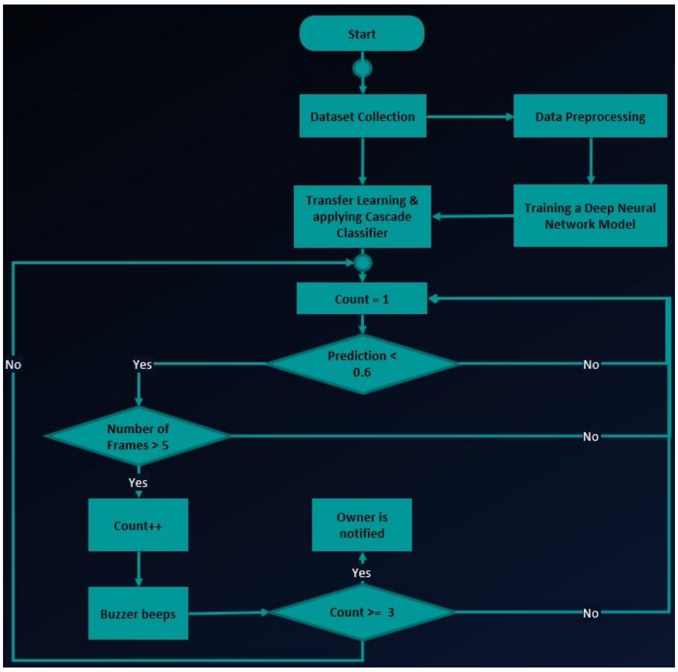
D. OpenCV

OpenCV is another library of several programming functions that aims at real-time computer vision. We used this library to process images in real-time and access a camera and display the outcome to the person.

III. WORKING MODEL

A. Dataset Collection and Preprocessing

We could not find the perfect dataset, but we collected data from the MRL Eye Dataset and modified it as per our requirements. The dataset consists of gray images. We converted the dataset to a fixed size each of (224, 224). The dataset collected is huge, and can be difficult to train, so we shrunk the data to 2600. Having them resized, we then separated the data for open eyes and closed eyes. Data is then shuffled so that the integrity remains the same throughout the dataset. Then the data is preprocessed and normalized.



Workflow

B. Building Deep Learning Model - Transfer Learning

The model we created is a customized deep learning model. We have used Mobilenet which is a pre-trained model for 1000 classes. It consists of Convolutional Neural Networks, applied batch normalization, ReLU activation function. Since we require only two classes i.e open, close. We replaced the connected layer of Mobilenet with our layer consisting of two classes. The change in the parameters after replacing them with the new layer is huge.

C. Training the Model

The model thus created is fed with the preprocessed data. We have used Google Colaboratory as our platform to train and fit the model. Colab provided us with a GPU which made the training a bit easier. Below are the details of our model.

```
history1 = my_model.fit(X, Y, epochs=10, verbose=1, validation_split=0.2)

Epoch 1/10
68/68 [==============================] - 20s 211ms/step - loss: 0.1126 - accuracy: 0.9640 - val_loss: 0.6520 - val_accuracy: 0.67
Epoch 2/10
68/68 [==============================] - 13s 186ms/step - loss: 0.0298 - accuracy: 0.9940 - val_loss: 0.5838 - val_accuracy: 0.73
Epoch 3/10
68/68 [==============================] - 13s 187ms/step - loss: 0.0052 - accuracy: 0.9972 - val_loss: 0.2444 - val_accuracy: 0.88
Epoch 4/10
68/68 [==============================] - 13s 187ms/step - loss: 0.0127 - accuracy: 0.9963 - val_loss: 0.3109 - val_accuracy: 0.89
Epoch 5/10
68/68 [==============================] - 13s 187ms/step - loss: 0.0140 - accuracy: 0.9958 - val_loss: 0.0994 - val_accuracy: 0.96
Epoch 6/10
68/68 [==============================] - 13s 190ms/step - loss: 0.0065 - accuracy: 0.9986 - val_loss: 0.1636 - val_accuracy: 0.92
Epoch 7/10
68/68 [==============================] - 13s 189ms/step - loss: 0.0035 - accuracy: 0.9991 - val_loss: 0.0216 - val_accuracy: 0.98
Epoch 8/10
68/68 [==============================] - 13s 189ms/step - loss: 0.0014 - accuracy: 0.9995 - val_loss: 0.2753 - val_accuracy: 0.88
Epoch 9/10
68/68 [==============================] - 13s 190ms/step - loss: 0.0398 - accuracy: 0.9898 - val_loss: 0.1114 - val_accuracy: 0.96
Epoch 10/10
68/68 [==============================] - 13s 191ms/step - loss: 0.0170 - accuracy: 0.9945 - val_loss: 0.0583 - val_accuracy: 0.97
```

D. Application

The application starts here using the OpenCV library, it starts capturing the video of the person sitting in front of the camera and captures the image of him/her. Having been captured, it then identifies and detects the face and eye positions using the Haar Cascade Classifier. The detected eyes are then fed to the trained model which predicts the state of the person (Active or Drowsy-inactive).

E. Buzzer and Alert

The outcome from the prediction is then used to start the buzzer and alert. It is done and achieved using Python module beepy or winsound that provides a feature to beep. If the person is found inactive or drowsy for a continuous duration or for a fixed range of time, the buzzer starts beeping and alerts the user. That makes the person focus on driving thus preventing the action of an accident.

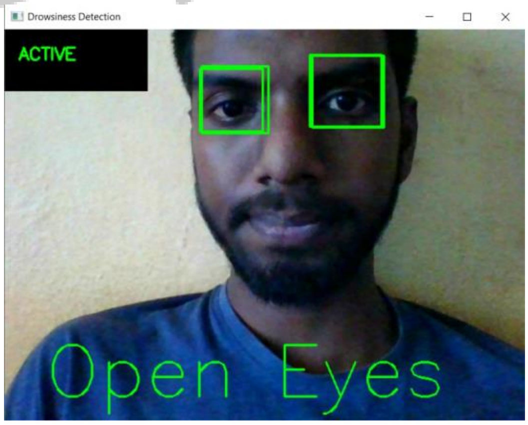
IV. VISUALISING PERFORMANCE OF THE MODEL

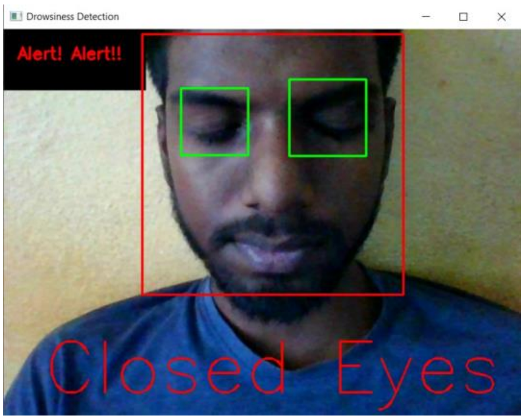
Our model has reached an accuracy of 99.45% with a loss of 0.55%.

A. Accuracy (Train vs Validation)

B. Loss (Train vs Validation)

V. RESULT





VI. CONCLUSION

The predominant motive of this research is to provide a Drowsiness Detection System and detect Driver Fatigue and a method to detect driver's drowsiness in real-time. The current approaches have used psychological measurements but these are intruding and they depend on the physical characteristics. We have proposed and developed an algorithm that is a non-intrusive technique for determining driver's fatigue and alerts the person.

VII. REFERENCES

[1] T. Soukupova and J. Cech, "Real-time eye blink detection using facial landmarks," in 21st Computer Vision Winter Workshop (CVWW 2016), 2016, pp. 1-8

[2] Accidents-https://www.nsc.org/road-safety/safety-topics/fatigued-driving

[3] "DRIVER FATIGUE AND ROAD ACCIDENTS A LITERATURE REVIEW and POSITION PAPER" (PDF). Royal Society for the Prevention of Accidents. February 2001.

[4] E. Vural, "Video-based detection of driver fatigue," Graduate School of Engineering and Natural Sciences, 2009

[5] Z. Ma, B. C. Li, and Z. Yan, "Wearable driver drowsiness detection using electrooculography signal," 2016 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet), Austin, TX, 2016, pp. 41-43.

[6] Wijnands, J.S.; Thompson, J.; Nice, K.A.; Aschwanden, G.D.P.A.; Stevenson, M. (2019). "Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks". Neural Computing and Applications. 32 (13): 9731–9743. arXiv:1910.06540. Bibcode:2019arXiv191006540W. doi:10.1007/s00521-019-04506-0. S2CID 204459652.

[7] Yan, Chao. "Vision-based Driver Behaviour Analysis." Ph.D. diss., University of Liverpool, 2016.

[8] Z. Li, L. Chen, J. Peng, and Y. Wu, "Automatic detection of driver fatigue using driving operation information for transportation safety," Sensors, vol. 17, no. 6, p. 1212, 2017

[9] Tinoco De Rubira, T. (December 11, 2009). "Automatic Fatigue Detection System" (PDF). Stanford University.

[10] Adrian Kaehler; Gary Bradski (14 December 2016). Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library

[11] Bartlett, M.S., Littlewort, G.C., Frank, M.G., Lainscsek, C., Fasel, I., Movellan, J.R.: Automatic recognition of facial actions in spontaneous expressions. Journal of Multimedia 1(6), 22–35

[12] 1.Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: FG 2000. Proceedings of the Fourth IEEE International conference on automatic face and gesture recognition, Grenoble, France, pp. 46–53. IEEE Computer Society Press, Los Alamitos (2000).