Driver Fatigue Detection Using Convolution Neural Network

Abstract

The main causes of road accidents are driver weariness and drowsiness. Detecting driver drowsiness (DDD) or fatigue is a critical but tough task in avoiding roadside incidents. This device automatically detects driver drowsiness using optical data and artificial intelligence. To increase safety and prevent these events, we developed a Drowsiness Detection System. When the system detects tiredness in the driver, it will alert (alarm) the driver.

Introduction

The negative repercussions of road accidents have increased as the population and use of automobiles has increased, including fatal injuries, loss of life, financial losses, and non-recoverable health and mental disease. The operating of a motor vehicle when psychologically compromised owing to a lack of sleep is known as sleep deprived driving. Driving while fatigued is one of the most common causes of car accidents. A person's ability to work successfully is harmed when they don't get adequate sleep. They have a lengthier reaction time, memory issues, and poor judgment when their capacity to perform is limited. Numerous studies have demonstrated that sleep deprivation has the same effect on driving as alcohol intoxication. Fatigue driving refers to the phenomena in which drivers are unable to obtain timely information about road conditions due to reduced sensory sensitivity, distraction, and even unconsciousness in the driving process as a result of long periods of driving or unsatisfactory rest conditions.

Traditionally, driver fatigue detection has been the cutting-edge field of active safety in automobiles. Several scholars are actively researching various aspects of fatigue detection algorithms. Scholars' detections of driver fatigue mostly rely on the vehicle's condition, physiological signals, the usage of attitude, and other factors. Dr Wier Wille et al. suggested the PERCLOS evaluation method as a real-time vehicle tiredness detection tool (1996). Simon et al. researched the varied values of brain wave signals during driver tiredness and non-fatigue in recent years and developed a model that accurately reflects fatigue condition (Simon et al., 2011).

The current problem was solved using recurrent neural networks (RNNs), long short-term memory (LSTM), auto-encoder (AE), convolutional neural networks (CNNs), and deep stacking networks (DSNs). CNN models are the most commonly used in biological signal classification for anomaly detection due to their high classification accuracy.

With the rise in popularity of CNNs, there has been a surge in interest in data augmentation. Several Deep Learning (DL) research projects have used the DA technique in the training step to reduce over-fitting and improve network performance by improving accuracy. In our research, we applied the DA technique to improve the performance of the suggested system.

The purpose of this study is to develop a Drowsiness Detection System in order to increase safety and prevent such events. When the system detects tiredness in the driver, it will alert (alarm) the driver.

Dataset Description

We are using facial data from UMass Amherst open eye face data and Nanjing University closed eye face data.

Open eye face data from UMass Amherst**:** For the data gathering, almost 13,000 photographs of faces were acquired from the internet. Each face has the name of the person pictured written on it, 1680 of the persons covered in the data set had two or more distinct photos. The sole constraint was that these faces had to be detected using the Viola-Jones face detector.

Nanjing University closed eye face: We generated a dataset for eye closeness detection in the wild to explore the performance of eye closeness detection in these situations. This dataset includes 2423 subjects, including 1192 subjects with both eyes closed who were obtained directly from the Internet and 1231 subjects with both eyes open who are chosen from the Labelled Face in the Wild(LFW[2]) database

Project Description

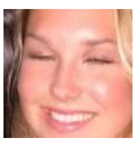
1. Description

Our project is divided into following phases:

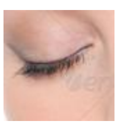
Data Collection: We used full facial data from a variety of sources, including open eye face data from UMass Amherst and closed eye face data from Nanjing University.

Using a simple python script, the eyes were cropped out of the dataset, leaving us with little more than 30,000 cropped eye images. To capture not only the eye but also the area surrounding it, we included a buffer to each image crop. Later, in the webcam portion, this cropping feature will be used.

Input given to data collection phase:



Output of data collection phase:



Data Augmentation Phase: To improve the accuracy of the data, we used the Image Data Generator class from the Keras API. We calculated the statistics needed to perform the transforms on our data after configuring ImageDataGenerator. The model's accuracy was improved by 4.8 percent as a result of this phase.

Model Building Phase: We'll be constructing a CNN network with nine layers. Rather than producing the complete image, this layer creates chunks of pixels, allowing for faster models. Depending on how many filters you use, this may be more dense than the original photographs, but it will allow the model to learn more complicated correlations with fewer resources. We utilized a total of 64 filters. At least one convolutional layer is suggested, but two or more is frequently preferred. For me, the ideal layout was two 3x3s pooled together, followed by three 3x3s pooled together.

As the number of neurons in these layers increases, the network's ability to acquire links becomes more complicated. Convolutional layers are utilized to avoid having to design an overly complex layer scheme.

Finally, because this is a binary classification problem, we utilize sigmoid activation in the outer layer.

We have followed following steps in building model.

i. Instantiate the Sequential Model

ii. Adding the first three Convolutional Layers. Each layer has 32 filters, kernel size of 3 and a relu activation function.

iii. Added a layer for pooling, Pooling layers are used to minimize the map's dimensionality. It decreases the number of parameters that must be learned as well as the amount of computation that the neural network must execute.

iv. Adding three more convolutional layers which are dense.

v. A dropout layer is added to avoid overfitting.

vi. The model is stored in h5 version for storing in the weights and reusing.

Proposed CNN model has following layers:

i. *Convolutional layers*: The layers allow filter application and features extraction based on the input pixel array obtained from converting input images.

ii. *Drop Layer*: Each dropout layer is a regularization approach that allows for improved over-adjustment of neural networks, lowering the classification error rate. Dropout is equal to 0.2 in the proposed model. To prevent overfitting, we silenced 20% of the neurons. Three dropout layers were used in our design.

iii. *Flatten Layer*: A multidimensional data output was provided in the previous stage, which could not be read directly from this neural network, so the model was flattened.

iv. *Dense Layer:* The dense layer's job is to characterize the connections between the following and intermediate layers of neurons. We employed two fully connected layers in our architecture. To improve classification results, we used a hidden layer of 128 neurons in the first dense of our model. The final neuron in the second dense has a value of one. Because binary classification is utilized in this study, a single neuron is sufficient to signify class "1" or "0."

v. Final Phase: We'll take photos with a webcam and use them as input. As a result, we constructed an infinite loop that captures each frame in order to acquire access to the webcam. The cv2 method provided by OpenCV is used.

Because the OpenCV object detection algorithm only accepts grayscale images as input, the image must be converted to grayscale to discover the face in the image. It generates an array of detections with x,y coordinates as well as height, which is the object's border box width. We can now draw boundary boxes for each face as we iterate across them.

After that, the image can be ran through the model to provide a prediction. If the prediction is close to 0, we display "Open" on the screen. Otherwise, we'll show "Closed" (meaning it's getting closer to 1). The number is reset to 0 if the model detects open eyes, and increased by 1 if the model detects closed eyes. We can display some basic text to indicate whether the eyes are closed or open using cv2.putText ().

2. Main References used for your project

- Qaisar Abbas, "Hybrid Fatigue: A Real-time Driver Drowsiness Detection using Hybrid Features and Transfer Learning" International Journal of Advanced Computer Science and Applications (IJACSA), 11(1), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110173.
- Sahayadhas, A.; Sundaraj, K.; Murugappan, M. Detecting Driver Drowsiness Based on Sensors: A Review. *Sensors* 2012, *12*, 16937-16953.

3. Difference in Approach/Method between our project and the references.

- Among all the state-of-the-art DDD approaches, the authors of the above-mentioned sources devised a method based on physiological phenomena that is well-thought-out as the most correct process to anticipate DDD driver's state. These methods are precise, but they require that all inputs be physically linked to the driver's body. As a result of this procedure, the driver became angry and preoccupied. Furthermore, extended driving may cause sweating on sensors (especially in KSA, where the temperature is quite high), decreasing their accuracy and making exact monitoring more difficult. While less intrusive, these methods are still overly obtrusive in practice.
- Many EEG-based machine learning (ML) research efforts have been recommended in medical diagnosis for classification-based drowsiness detection jobs throughout the last decade. Nonetheless, significant limits exist in machine learning applications, such as the need for a large dataset to train, limited predictions in return, and the need for an intermediate step for feature representation and drawing conclusions to detect anomalies.
- It's also worth noting that this reference 1 used ECG sensors and the PERCLOS evaluation method, whereas we used computer vision to collect and locate the face in a live feed image.
- We have undertaken data augmentation, which is one of the reasons for boosting the accuracy, although both references found the results simply based on the data provided.
- Considering the aforementioned concerns and problems, we proposed a method that uses CNN-based features to capture a variety of latent face characteristics and complex nonlinear properties. This strategy is intended to prevent road accidents by alerting the motorist to the fact that he or she is sleepy. The precision-recall area under the curve score for the trained classifier is 92.33.

4.Difference in Accuracy/Performance between your project and the main projects of your references.

- We utilized precision-recall area under the curve to measure the performance of the models, whereas the above-mentioned sources used accuracy. The better the recall, the fewer weary drivers the program incorrectly thinks will be awake.

- The correctness of the reference 1 is 94.3 percent. It's also worth noting that this study relied on ECG sensors and the PERCLOS evaluation method, whereas we employed computer vision to record and recognize the face from a live webcam stream.

- The reference, on the other hand, has a 90% accuracy rate because of superior CNN design and EEG-based research.

Analysis

1.What did we do well?

- We were able to successfully do data augmentation by increasing the amount of photographs in the dataset using transformations, which increased the model's accuracy.

- We successfully integrated a webcam into the model, which takes photos, recognizes faces, and then the model handles the rest.

- Under a variety of scenarios, this system was tested in a real-time environment mode at night and during the day.

- The suggested method detects tiredness if the eyes have been closed for four or more frames. The detection technique can distinguish between a normal eye blink and fatigue. The method that has been developed is completely non-invasive.

2.What could I have done better?

- Increase the model accuracy by using advance CNN architecture.

- Using GPU-enabled resources to reduce the model's training time and thereby improve performance.

- Instead of using the CNN, use advanced Deep Learning techniques to improve the model's accuracy.

- Adding emotional analysis as an element to the classification process to improve performance. We may use the Driver Emotion Detection Tool to add another attribute to our toolbox.

3.What is left for future work?

- We can integrate behavioural techniques with vehicle-based measures to improve dependability and accuracy. We could combine the current system, which relies solely on visual data to detect weariness, with an ECG sensor to offer a BPM reading for better performance. We can combine AER, MAR, and BPM ratio values to increase accuracy
- Although face recognition has high accuracy rate for detecting tiredness, to achieve optimal results it is also necessary to consider driving environment.

```
# fitting the model
model.fit(X_train,
          y_train,
          batch_size=500,
          validation_data=(X_test, y_test),
          epochs=30)
```

Accuracy = [0.09342602849006653, 0.9100980520248413]

Accuracy Train	Validation	Precision Recall Score Train	Precision Recall Score Validation
92.3	84.5	94	93

Conclusion

The purpose of this study is to compare articles on driver sleepiness detection and alarm systems. To solve the problem of detecting drowsiness, an arithmetic-based technique is used. This technique employs eye movement to detect weariness. To detect eye movement, a camera is employed. This allows fatigue indicators to be identified and accidents to be prevented. It is based on the concept of eye tracking. We used open eye face data from UMass Amherst and closed eye face data from Nanjing University.

It is decided to create a software algorithm. It was tested in part and found to be useful. There is a lot of room for improvement. If the eyes have been closed for two frames, the suggested technique identifies drowsiness. The detection technology can tell the difference between a regular eye blink and tiredness. The system that has been created is non-invasive. Various types of sensors can be added to the system to further develop it.