Driver Fatigue Detection using Convolutional Neural Networks

Abstract

Driver drowsiness and fatigue are common causes of car accidents. Detecting driver drowsiness (DDD) or weariness is a crucial and difficult task in preventing roadside collisions. This device uses optical information and artificial intelligence to identify driver drowsiness automatically. We built a Drowsiness Detection System to improve safety and prevent these incidents. When the driver's drowsiness is detected, the system would alert (alarm) the driver.

Introduction

With the rise in population and use of automobiles, the negative consequences of road accidents have escalated, 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 tired is a leading cause of car accidents. When a person does not receive enough sleep, their capacity to work effectively is compromised. When their capacity to function is reduced, they have a longer reaction time and have memory problems and judgment. Sleep deprivation has been shown in numerous studies to have the same impact on driving as alcohol intoxication. Fatigue driving refers to the phenomenon that the drivers cannot get timely information of road conditions due to sensory sensitivity reduction, distraction, and even unconsciousness in the driving process caused by long-time driving, or unsatisfied rest conditions of themselves.

Driver fatigue detection has traditionally been the cutting-edge field of active safety in vehicles. Various features of fatigue detection algorithms are being actively researched by several academics. 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).

Recurrent neural networks (RNNs), long short-term memory (LSTM), auto-encoder (AE)], convolutional neural networks (CNNs), and deep stacking networks (DSNs) have all been used to solve the current challenge. Because of its excellent classification accuracy, CNNs models are the most utilized in biological signals classification for anomaly detection.

With the growing popularity of CNNs, interest in data augmentation grew significantly. Several DL research projects have used the DA approach in the training step to reduce over-fitting and increase network performance by enhancing accuracy. We used the DA technique to increase the performance of the proposed system in our research.
The aim of this paper to build a Drowsiness Detection System to improve safety and prevent these incidents. When the driver's drowsiness is detected, the system would 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.

<u>UMass Amherst open eye face data:</u> More than 13,000 photos of faces were gathered from the internet for the data collection. The name of the individual pictured has been written on each face. In the data set, 1680 of the people featured had two or more distinct photographs. The Viola-Jones face detector was used to detect these faces, which is the only constraint

<u>Nanjing University closed eye face</u>: 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 were chosen from the Labeled Face in the Wild (LFW [2]) database.

Project Description

1. Description

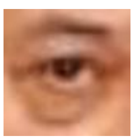
Our project is divided into following phases:

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

The eyes were then cropped out of the dataset using a simple python script, leaving us with slightly over 30,000 cropped eye images. We added a buffer to each image crop to capture not only the eye but also the area around it. This cropping feature will be used in the webcam section later.

Input given to data collection phase:

Output of data collection phase:



Data Augmentation Phase: We have used a Image Data Generator class from Keras API to augment the data we have for increasing the accuracy.After configuring ImageDataGenerator,we calcualted statistics need to perfom the transforms to our data.This phase resulted in increasing the accuracy of the model by 4.8%.

Model Building Phase: We will be building a CNN network with 9 layers. This layers, rather than creating the entire image, produces sections of pixels, allowing for speedier models. This may be dense than the original photos, depending on the number of filters you use, but it will allow the model to learn more complex associations with fewer resources. We used 64 filters in total. Use at least one convolutional layer, and two or more is usually recommended. Two 3x3s pooled together followed by three 3x3s pooled together was the best setup for me.

The complexity of the relationships acquired by the network will rise as the number of neurons in these layers grows. Convolutional layers are used to prevent having to create a thick layer scheme that is too deep.

Finally, because this is a binary classification task, we make sure outside layer uses sigmoid activation.

We have followed following steps in building model.

i. Instantiate the Sequential Model

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

iii. Added a pooling layer, Pooling layer are used to reduce the dimensions of the maps. It reduces the number of parameters to learn, and the amount of computation performed in the neural network

iv. Adding 3 more dense convolutional layers.

v. Adding a dropout layer to avoid overfitting.

vi. Store the model 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 technique that allows for better over-adjustment on neural networks, lowering the error rate during the classification process. The value of dropout in the suggested model is equal to 0.2. We have deactivated 20% of the neurons to minimize overfitting. In our model, we used three dropout layers.

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

iv. Dense Layer: The dense layer's job is to describe how the following and intermediate layers of neurons are connected. In our architecture, we used two fully connected layers. We employed a hidden layer of 128 neurons in the first dense of our model to improve classification results. The value of the final neuron in the second dense is one. In this study, binary classification is used, therefore a single neuron is enough to denote class "1" or "0."

Final Phase: We'll use a webcam to capture photographs as input. So, to gain access to the webcam, we created an infinite loop that captures each frame. We employ the cv2 technique offered by OpenCV

To find the face in the image, the image must be converted it to grayscale, as the OpenCV object detection algorithm only accepts grayscale images as input. It produces an array of detections with x,y coordinates as well as height, which is the width of the object's border box. We can now iterate across the faces, drawing boundary boxes for each one.

The image can then be running through the model to obtain a prediction. We display "Open" on the screen if the prediction is closer to 0. Otherwise, we display "Closed" (i.e., it's closer to 1). If the model identifies open eyes, the counter is reset to 0, and if the eyes are closed, the counter is increased by 1. Using cv2.putText, we can display some basic text to indicate whether the eyes are closed or open ().

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.**

- The authors of above-mentioned references devised a method based on physiological phenomena that is well thought-out as the most correct process to anticipate DDD driver's state among all those state-of-the-art DDD methods. These methods are accurate, but all inputs must be physically attached to the driver's body. As a result, the driver was agitated and distracted because of this process. Furthermore, prolonged driving might produce sweating on sensors (particularly in KSA, where the temperature is rather high), lowering their accuracy and making it more difficult to monitor precisely. This approaches, while less invasive, are nonetheless too intrusive in practice.
- Over this decade, many EEG-based research works related to machine learning (ML) have been suggested in medical diagnosis, for classification-based drowsiness detection tasks. Nevertheless, some limitations appear in ML applications such as the need for a massive dataset to train, limitation predictions in return, the need of an intermediary step for feature representation and drawing conclusions to detect anomalies.
- We also need to keep in mind that this reference 1 used ECG sensors and PERCLOS evaluation method, while we used computer vision to capture the and find the face from an image from a live feed.
- While both the references found the results solely based on the data given, we have performed data augmentation which is one of the reasons for increasing the accuracy
- Keeping in mind above mentioned problems and issues, we suggested a method that captures numerous latent facial characteristics and complex nonlinear properties using features learnt using a CNN. This technique is meant to avoid road accidents by alerting the driver to tiredness. The results of the trained classifier show a precision-recall area under the curve score as 92.33.

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

- While the above-mentioned references used accuracy to measure the performance the models, we used precision-recall area under the curve to measure the performance. The higher the recall the smaller number of tired drivers the model wrongly predicts are awake.
- While the reference 1 have an accuracy of 94.3%. We also need to keep in mind that this reference used ECG sensors and PERCLOS evaluation method, while we used computer vision to capture and recognize the face from the live feed sent by the web cam.
- While the reference has an accuracy of 90% using advanced CNN architecture and EEG based research.

Analysis

1.What did we do well?

- We were able to successfully do data augmentation by using transformations to increase the number of photos in the dataset, which improved the model's accuracy.
- We successfully implemented computer vision ideas to integrate a webcam into the model, which gathers images, recognizes faces, and then model does the remaining.
- This system was evaluated in a real-time environment mode at night and during the day under a variety of scenarios.
- If the eyes have been closed for four or more frames, the suggested method 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.

```
# 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	Accuracy Validation	Precision-Recall Score Train	Precision-Recall Score Validation
92	86	94	93

2.What could I have done better?

- Increase the model accuracy by using advance CNN architecture.
- Using GPU enabled resources to decrease the training time of the model, thereby increasing the performance.
- Use advanced Deep Learning algorithms to instead of the CNN to increase the accuracy of the model.
- Applying emotional analysis as a one more feature while classifying to increase the performance. We can use Driver Emotion Detection Tool which gives us one more attribute to work with.

3.What is left for future work?

- To improve dependability and accuracy, we can combine behavioral approaches with vehicle-based metrics. We may integrate the present system, which simply relies on visual data to detect fatigue, with an ECG sensor to provide a BPM readout for improved performance. To improve accuracy, we can combine AER, MAR, and BPM ratio values.

- Although face recognition technologies have a high accuracy rate for detecting tiredness, they are both obtrusive and unnecessary. However, by adopting contactless electrode implantation, this intrusive character can be eliminated. As a result, in the development of an effective sleepiness detection system, it would be worthwhile to combine physiological indicators such as ECG with behavioral and vehicle-based measures. To achieve optimal results, it is also necessary to consider the driving environment.

Conclusion

This work conducts a comparison of papers on driver sleepiness detection and alert systems. An arithmetic-based strategy is employed to provide a solution to the challenge of detecting the state of drowsiness. To identify fatigue, this technique uses eye movement. A camera is used to detect eye movement. This is done so that fatigue signs can be recognized, and accidents can be avoided. It is based on the eye-tracking idea. We are using facial data from UMass Amherst open eye face data and Nanjing University closed eye face data.

A software algorithm is developed. It was partially tested and found to be effective. There is much scope for further improvements. The proposed system detects drowsiness if the eyes have been closed for a period of 2 frames. The detection system differentiates the normal eye blink from drowsiness. The developed system is a non-invasive system. The system can be further developed by adding various types of sensors.