Driver Drowsiness Detection System using Convolution Neural Networks in Deep Learning

Bhanu Soni (CSE013) and Dhruv Soni (CSE016)
Department of CSE, IET, JK Lakshmipat University

Objective

The main objective of this project is to prepare a prototype of a driver drowsiness system that alerts the driver when he is drowsy and sleepy.

This is achieved by the use of the Deep Learning model to detect the face of the driver using a camera and analyze the state of the driver.

In our Deep Learning model, we applied the famous Deep Learning algorithm i.e., Transfer Learning.

Requirements:-

- Python- 2.7 or 3.9 version
- OpenCV- Face and Eye Detection
- Keras- To build our classification model.
- TensorFlow- Keras uses TensosrFlow as a backend.

Introduction

Drowsiness is the biggest reason for road accidents and Data Science is the best remedy for it.

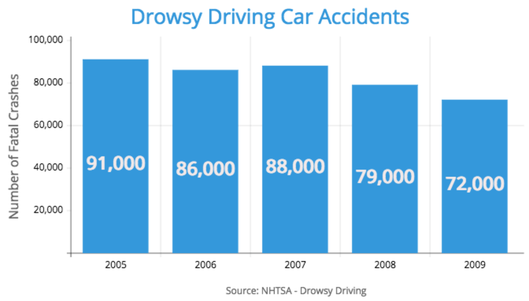
Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.

According to a survey, 20% of the Road Accidents are caused due to the drowsiness of the driver.

This turns out to be a big problem not only for the driver but also for other people who use that road.

This Drowsy Detection System is a safety alarm system that alerts the driver whenever he feels drowsy.

The eye movement of the driver is monitored live and whenever the driver feels asleep or closes their eye for more than 1 sec. Then it alerts the driver with the help of a loud alarm. Thus, preventing any accidents from happening.



Methodology

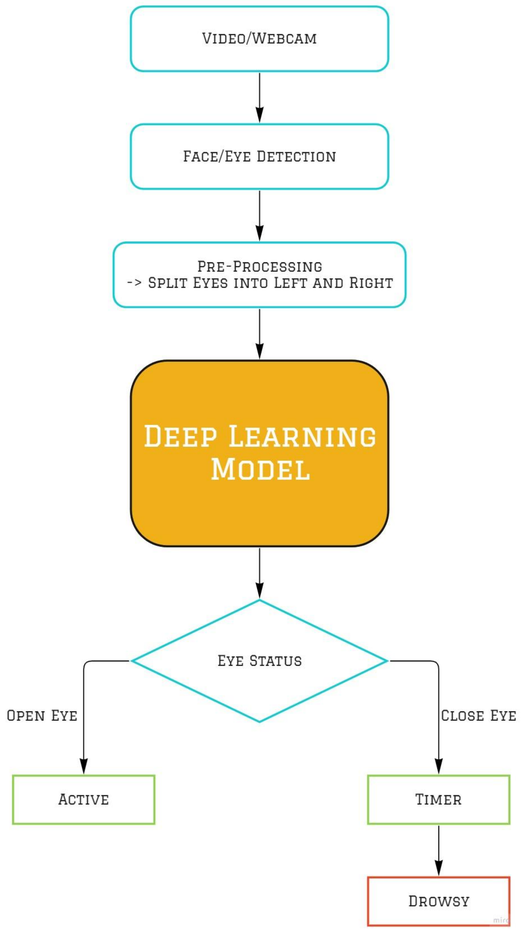
The model we used is built with Keras using Transfer Learning.

A transfer learning is a special type of deep neural network which perform extremely well on a large amount of data for image classification purposes. Therefore, we only train them by fine-tuning the model. The benefit that we will get is the model will train in a short time.

With the help of Transfer Learning in our MRL Eye dataset, we resize the image from 86 X 86 to 224 X 224. according to selected Pre-trained Models(AlexNet, GoogleNet, MobileNetv2, VGG19, and RestNet101)

Using the ImageNet classifier technique apply this technique to 2 classes on the MRL-Eye dataset which modifies the Classification layer and the last fully connected layer.

MRL Eye dataset containing images of 37 different individuals (33 men and 4 women).



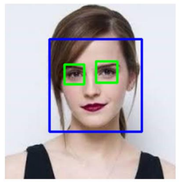
Working Procedure:

Step 1- Take Image as Input from MRL Eye Dataset

- Classify images into the closed eye and open eye.
- Reading all the images and converting them into an array for data and labels.

Step 2- Detect the Face in the image and create a Region of Interest(ROI)

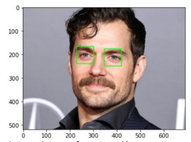
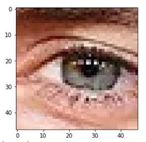
- To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes grey images in the input.
- We don't need color information to detect the objects. We will be using a haar cascade classifier to detect faces.



OpenCV

Step 3- Detect the eyes from ROI and feed them to the classifier

- The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for the eyes in the left eye and right eye respectively then detect the eyes.
- Now we need to extract only the eyes data from the full image.
- This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code
- Now the image data of the eye will be fed into our Transfer Learning classifier which will predict if the eyes are open or closed.

Step 4- The classifier will categorize whether Eyes are Open or Closed

- We are using a Transfer Learning classifier for predicting the eye's status. To feed our image into the model.
- First we convert the color image into grayscale using cv2.cvtColor(img_array, cv2.COLOR_GRAY2RGB).
- The we resize the image 224 X 224 cv2.resize(eyes_roi, (224, 224)).
- Now we predict each eye with our model new_model.predict(final_image).
- If the value of 1, it states that eyes are open, if the predicted value is 0 then, it states that eyes are closed.

Step 5- Check whether the person is Drowsy or not.

- If eyes are opened, then it will show false, and we determine how long the person has closed the eyes
- So, if both eyes are closed, in the output it shows eyes are not detected.
- We are drawing the result on the screen using the print function which will display the real-time status of the person.
- Now it the person's eyes are closed for a long period then an alarming sound is played.

Real World Application

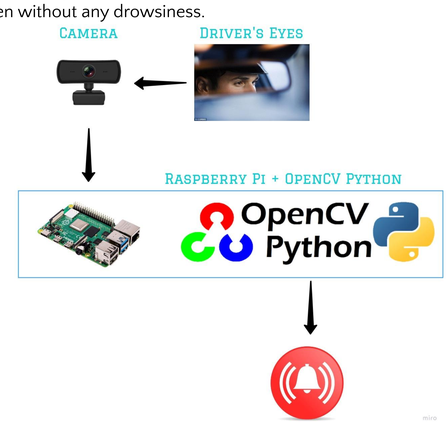
- This System can be used in any automobile with ease.
- This can be done by using Raspberry Pi and with the help of a Web Camera and an alarm for the alarm sound.

Raspberry Pi

Web Camera

Buzzer

- This Driver's eyes are constantly monitored by Webcam.
- This Data is sent to the image classifier model which is running on Raspberry Pi.
- The classifier checks every frame and classifies the frame as open or close
- If the eyes are being closed for a few seconds straight, then the Alarm is Triggered.
- The alarm goes off only when the driver is active, and his eyes are open without any drowsiness.



Conclusion

- This system can be used to reduce the number of road accidents that happen to a great extent.
- This can save a lot of lives, which is the main motive of this system.
- This system does not need any complex system to work effectively.
- Taking the facts into consideration driver drowsiness detection system is the future of road safety.