Drowsiness Detection with Anti Spoofing

Group - Aviato

Aryan Jain (11840210)

Chirag Poonia (11840350)

Abstract

Every year the amounts of deaths and injuries are increasing in traffic accidents due to human errors. Drowsy driving is very hazardous and it is very difficult to identify. After alcohol drowsiness is the second leading cause of road accidents. If the Driver fails to concentrate on driving it reduces the driver reaction time and impairs steering behavior. Driver drowsiness can cause several physical and economical losses; One way to detect driver's drowsiness is to observe the driver's driving patterns, if the driver is not concentrating on driving alerts the driver with the alarm sound. In this study, machine learning was applied to predict drowsiness and improve drowsiness prediction using facial recognition technology and eye-blink recognition technology.

Introduction

According to government statistics, exhausted drivers who doze off while driving account for more than 100,000 road accidents in the country. As a solution to resolve these problems, studies aimed at detecting and preventing this kind of driving are being actively researched. With the advent of autonomous driving, this problem is becoming increasingly common. The techniques generally used are recognizing the driver's eyes through a camera and using various biological signals such as breathing, temperature and heart rate and also identifying patterns such as abnormal use of pedals and steering wheel. In this project, we use driver's eye patterns to identify whether he/she is drowsy or not. Using a CNN we check whether the eyes are open or closed. If the eyes are closed, a score counter

increases. An alarm to alert the driver is automatically played when the score reaches a certain threshold. This is done to prevent incorrect registering of blinking as drowsiness.

Also, implemented is a face liveness detector, which prevents spoofing attacks, using photos and videos.

Problem Definition

The objective of this challenge is to design a detector/classifier that will detect whether the driver is alert or drowsy using facial recognition.Since we had already made the drowsiness detector, we had to make the liveness detector. The liveness detector will decide whether the person in the webcam is real or fake using various parameters

Technology Used

Python Libraries used : Tensorflow, OpenCV, Keras, pygame, dlib

Model

The problem of detecting fake faces vs real/legitimate faces is treated as a binary classification task. Basically, given an input image, we'll train a Convolutional Neural Network capable of distinguishing real faces from fake/spoofed faces.

In order to increase accuracy, some other features are also considered alongside with the result of CNN. Now the result from the CNN and other features will have to be positive in order to show a person live.

Other feature -

Eye aspect ratio using dlib.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 32, 32, 16)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 16)        64
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 16)        2320
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 16)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 32, 16)        64
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 32)        4640
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 32)        0
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 16, 32)        128
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 32)        9248
_________________________________________________________________
activation_4 (Activation)    (None, 16, 16, 32)        0
_________________________________________________________________
batch_normalization_4 (Batch (None, 16, 16, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                131136
_________________________________________________________________
activation_5 (Activation)    (None, 64)                0
_________________________________________________________________
batch_normalization_5 (Batch (None, 64)                256
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
_________________________________________________________________
activation_6 (Activation)    (None, 2)                 0
=================================================================
Total params: 148,562
Trainable params: 148,242
Non-trainable params: 320
_________________________________________________________________
```

IMPLEMENTATION

The model is pretrained. To improve the accuracy of the whole system, the model is paired with dlib's facial recognition system to detect eye aspect ratio.

Result

The model performs with great accuracy.

Conclusion and further work

Both the models work great individually. We tried to integrate them together, but due to the lag caused by running two models simultaneously, the result became less accurate.

References

https://github.com/sakethbachu/Face-Liveness-Detection