Drowsiness Detection

Date: 20th of June,2021

Let me introduce you to the problem statement which our project aims to solve.
Drowsiness is one of the biggest causes of inefficiency and inconsistency in every field.To detect drowsiness and alert the person when he/she is drowsy can help to prevent undesirable consequences.

- Drowsiness while driving causes fatal accidents
- Drowsiness decreases our efficiency while working
- It also slows us down when we have targets to achieve like sometimes people doze off while studying for exams.

So, a system which can **govern us** and **alert us** when we are drowsy can be very useful for increasing efficiency and also can prevent fatal consequences like when driving drowsy. ***Would that not?***
So,
This project acts as an end-to-end solution for detecting drowsiness in the real-time and alerting if necessary. It uses Live-cam for capturing the state of eyes of a person and then based on the frequency and duration of closing of eyes, it judges whether the person is drowsy, and alerts with an alarm if detected drowsy.

In this project, we use Computer Vision for detecting eyes , through haar cascade classifiers from OpenCV's public libraries and Machine Learning algorithms which detect the state of eyes continually.

Now, I will be explaining about the dataset and the basic working mechanism of our project.
While commissioning this project, a lot of time was spent for creating, cleaning and processing the data for the dataset. At early stages of development of the project, we used a ready made dataset from Kaggle which had about 4000 photos for model training & testing. **But,** later we took the initiative, and built our own dataset by taking videos from various people with various shapes and sizes of eyes.

We created a blemish free dataset of 16.5k images approximately for our final model. The dataset was thoroughly scrutinised while creation, to prevent presence of outliers and irrelevant data. The self-created Dataset contains 16,658 images out of which 8776 are images of open eyes, and 7882 are closed-eye images.

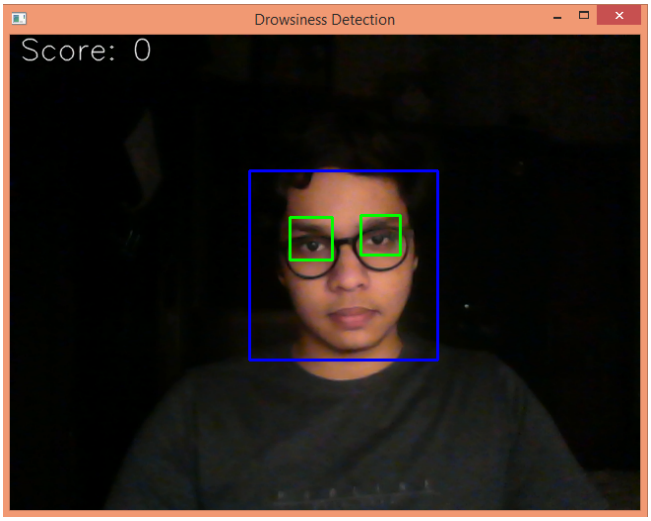
The data used in this project was created from videos of people having various shapes and sizes of eyes. The dataset was manually cleaned and preprocessed thoroughly for minimising errors in training.

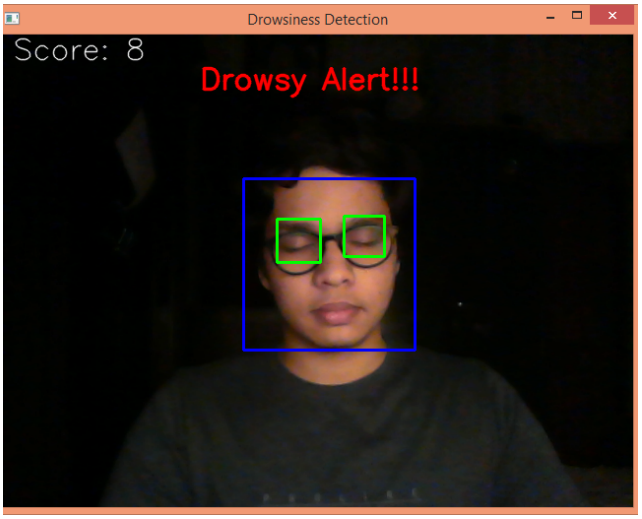
Now, let's see what's the criteria for detecting drowsiness. As you can see, The system continuously extracts frames from live-cam from which eye regions are cropped and passed on to the classifier, the classifier predicts whether the input image is open/closed.

This classification task executes continuously, based on which drowsiness score is calculated. Now, if it is detected that eyes are fully/partially closed then drowsiness score increases, and if open eyes are detected, then the drowsiness score reduces. And then the whole alert system comes into the picture.

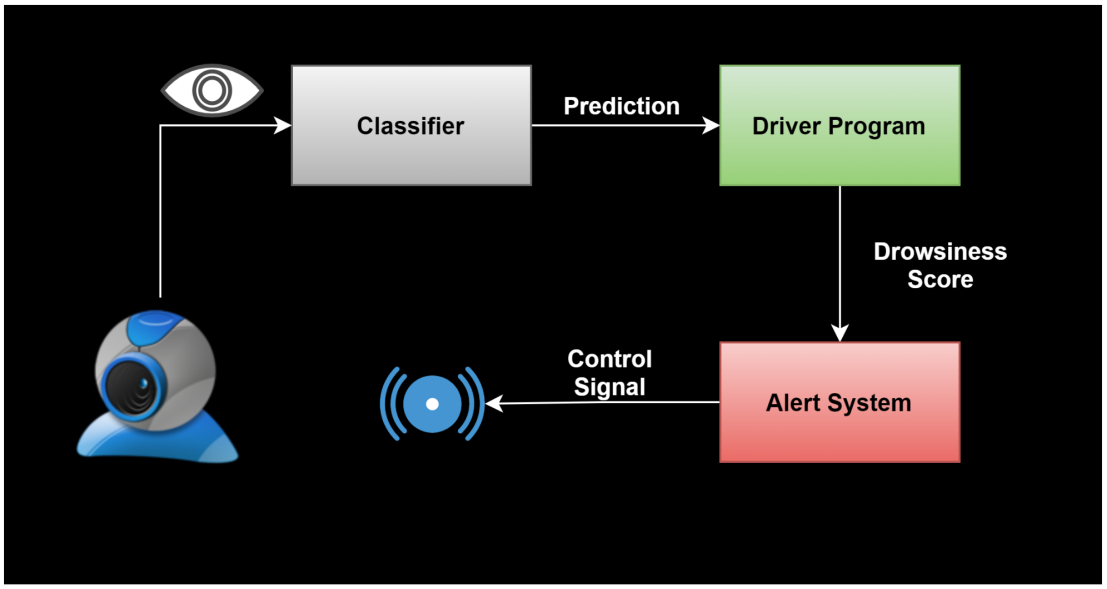
Now, about the alert system, the drowsiness score is used for determining whether the person is drowsy or not.

As you can see in the snapshot, the score is 8 which is greater than the threshold, hence the alarm system is active and its showing the Drowsy alert.





Next, I will be briefing about the overall working of the project, and the various machine learning & deep learning algorithms which we tried to implement for the project. Before proceeding to the details of the models, I would like to bring more clarity to what exactly our ML models do.



As you would have had a glimpse of the dataset, it contains static images of eyes in both open and closed states. So, it is clear that our machine learning models take images of the eyes as input and predict whether they are open or closed one at a time. For this project, we manage to get predictions for 3 frames per second from the live-cam, which adequately calculates the drowsiness score without any delay. Once the drowsiness score crosses the threshold, the beep alarm goes off, until open eyes are detected. We have also done some

fine tuning to improve the overall experience. For instance, the drowsiness score increases at a higher rate, but if it lies above the threshold, then the rate at which it decreases is relatively slower, in order to make sure that the person is not drowsy.

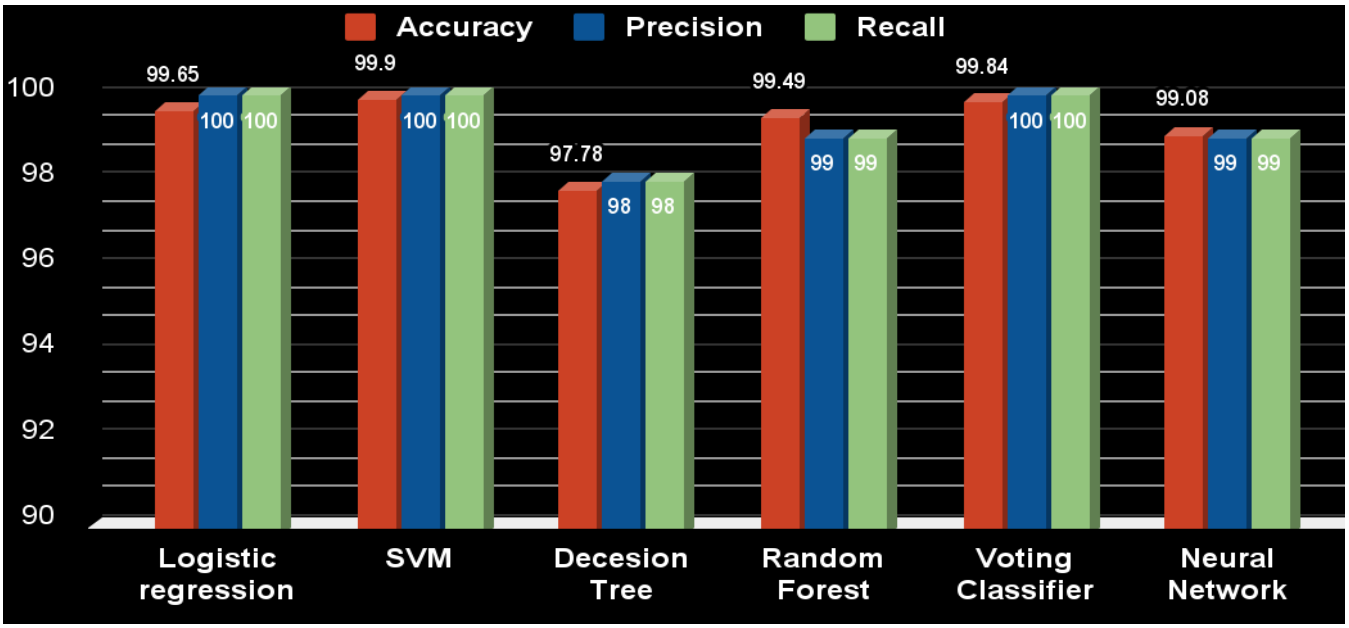
Now, about the ML models implemented in this project. For selection of the model which performs best for our project, we have tested a lot of models, out of which the top 6 are shown here.

As it is evident from the description, we basically had a binary classification problem to solve. And **logistic regression** is one of the finest binary classifiers because of the sigmoid function which it uses internally for prediction. And you can see that its performance is pretty good.

Similarly, we also used **Support Vector Classifiers** and Decision Trees. Both of them perform pretty well.

Moving on, next we have used ensemble methods, like random forest, and voting classifiers. In voting classifiers, we used Logistic Regression, Decision Tree and Support Vector Machines as an ensemble.

We also implemented neural networks for classification. It was a 3 layer sequentially compiled dense neural network model. Its performance was pretty good as well.



Here you can see the comparative analysis of the models. Though performance is pretty much the same for all, **in real** time testing Logistic Regression, SVMs and Neural Networks were observed to be relatively more robust.

You can see the video presentation of our project here:
https://github.com/usray07/drowsiness-detection/blob/7fe916927d4ae6fd6c199e43a5712c5e849f5e0a/Prashant.mp4

Next, I will be presenting about the applicability of this project. To begin with, this project can be used in various fields.

We all know road accidents have fatal consequences, according to a national survey 40% of road accidents are caused due to driver's fatigue. So, an alert system like this if mandated in vehicles can help reduce the no. of fatalities **significantly.**

This also has application in professional workspace, Many times due to stress or incompetencies people doze off at work, which has a negative impact on their professional career.

Also, In the present scenario, where all the meetings and classes are being conducted through video conferencing, this software can be used to track attentiveness of the participants.

This software can also be used personally, to increase efficiency. For example, when studying for exams overnight, many people fall asleep by mistake and end up losing time, but this software can ensure that they don't fall asleep. Hence very helpful.

It's evident that we tried to do our best to prepare theis model, but nothing in the world is perfect. So there is always scope for improvement. Though our project works for even partially open and closed eyes, the performance can be improved further. For instance, we have plans to make our detection system more robust, by including more features like yawn detection.

That's all from our side!

Now, we would appreciate your reviews and recommendations, about our project. So, please guide us with your comments and suggestions.

Thank you!!

Narrations by:

Uttam Sundar Ray

Prashant Kumar