# Project Check-in I

Individual Report

Project Info: Drowsiness Detection using CNN

Team: 13

Pallavi Suma Kurmala

Evakattu Muni Eshwar

Venkata Sai Prakash Yerramsetti

## Project Activities Completed Since Last Check-in

Read different papers on this topic, found many methods to achieve the objective. But our data is simpler than theirs, so, the training process can also be made simpler. Found a few methods, but still haven’t tried this practically yet.

## Project Activities in Progress

As I mentioned above, I am still trying to implement these training methods. I did get it trained by taking a sample of data, but the accuracy is still not satisfactory. I am still trying other ways for efficient training.

## Project Activities Planned

Data collection and training are handled by my groupmates. I am assigned with planning on how to extract the required features from each frame of the input video. In this, the eyes are the important feature, to classify whether the person is drowsy or not. For this process, first the face should be identified and then eyes can be extracted. I am still working on this by trying different techniques, nothing is working efficiently yet, I may be able to achieve this by the next check-in.

## GitHub/Google Drive links

The literature survey and my progress is added below this template .

## Peer evaluation

Muni Eshwar Evakattu : He is also trying on an efficient method to train this large dataset.

Venkata Sai Prakash Yerramsetti : He is working on data collection and preprocessing techniques.

**Individual Literature Survey**

**Project Name** : Driver Drowsiness Detection Using CNN

**Check -In** : Done literature Survey on how to capture the eyes from the face by exploring various exiting papers and looked over the OpenCV library and found many methodologies and found out the advantages and disadvantages of the method.

The main reason for focusing solely on eye detection is that, based on my teammate Evakattu Muni Eshwar's analysis of the literature survey, I came to the conclusion that we can only concentrate on the eyes dataset rather than detecting the face and localizing the eyes in the image because training that face dataset may take too long, limiting us to using only a few samples rather than many.

The methods and strategies used in eye tracking and detection were explained in these studies.

1. Rui Min et al. (2018) "Real-time eye detection and tracking for driver monitoring system":

A real-time eye detection and tracking system for driver monitoring is covered in this work.

**Summary**

The paper "Real-time eye detection and tracking for driver monitoring system" proposes a non-intrusive prototype computer vision system for real-time monitoring of a driver's vigilance.

The system estimates eye blinking and head pose using facial detection algorithms, which are critical for driver state analysis.

The paper describes a method for real-time eye detection and tracking that works in a variety of realistic lighting conditions.

Using a normalized cross-correlation function-based classifier, the proposed algorithm can detect eyelid movement and classify whether the eyes are open or closed.

Depending on the consistency of eye closures, the system classifies them as either occasional or sustained drowsiness. The accuracy of the algorithm is demonstrated in the paper using real data under varying conditions for people of different genders, skin colors, eye shapes, and facial hairs. The proposed system analyzes the driver's state using a single camera and does not

address hardware analysis information such as steering data and vehicle speed.

1. Xiaofeng Zhang et al. (2013) "Real-time eye tracking using robust template matching": The main topic of this research is template matching for real-time eye tracking. Even while it doesn't directly address OpenCV, it offers information on template matching methods used in eye tracking.

**Summary**

The paper "Real-time eye tracking using robust template matching" proposes a real-time eye tracking system that detects and tracks the eyes using template matching. To search the image for the eyes, the system employs a generic eye model based on the eye shape. The paper describes a robust template matching method that can handle changes in scale, orientation, and eye appearance, such as blinking. The proposed system searches for eyes in smaller windows using edge information and thresholding, and then tracks them using template matching. The accuracy of the algorithm is demonstrated in the paper using real data under varying conditions for people with different eye shapes and sizes. The proposed system is suitable for computer interfaces. The proposed system can be used for computer interfaces for people with disabilities as well as other applications that require real-time eye tracking.

1. Satya Mallick, "Eye tracking using OpenCV" (2016): Satya Mallick offers helpful advice in this lesson on how to use OpenCV to perform eye tracking. It's not a research article, but it can be a useful tool for learning the fundamentals.

**Summary**

The article "Eye tracking with OpenCV" describes how to incorporate eye tracking into any video using the OpenCV library and its functions. The article describes the basic steps of the implementation, which include converting each frame of the video to grayscale, detecting a face in the grayscale using the dlib library's get\_frontal\_face\_detector() method, defining the region of interest (ROIs) dimensions, which may include the eyes, and converting each ROI into a threshold image to aid in determining the contouring dimensions. The dimensions of the contouring list are then used to draw circles around the calculated eye position in the article. The article also mentions how OpenCV and techniques like contour detection can be used to detect eye blinks and track pupil size. The article shows how to build a low-cost eye-tracking system using only an off-the-shelf webcam, as well as how to design and implement a gaze-estimation method that estimates gaze positions using a simple yet powerful machine learning-based

calibration technique.

1. Z. L. Zhang et al., "An Eye Movement Analysis Method Based on OpenCV" (2018): -

An OpenCV-based technique for analyzing eye movements is presented in this research. It might provide information on how OpenCV is utilized to analyze eye movements.

**Summary**

The article "An Eye Movement Analysis Method Based on OpenCV" proposes an OpenCV-based method for analyzing eye movements. The basic steps of the implementation are described in the paper, which include converting each frame of the video to grayscale, detecting a face in the grayscale using the Viola-Jones algorithm, and detecting the eyes using the Hough transform. The detected eye positions are then used to calculate eye movement parameters such as saccade amplitude, saccade velocity, and fixation duration. The proposed method has applications in human-computer interaction, assistive technologies, and medical diagnosis.

Using a simple webcam and OpenCV, the proposed system can perform real-time eye tracking and analysis.

1. Shubhendra Yadav's (2018) article "Real-time eye gaze tracking using OpenCV on Python":

This tutorial uses OpenCV in Python to show real-time eye gaze tracking. It may serve as a useful manual for putting eye tracking apps into practice.

**Summary**

The article "Real-time eye tracking using OpenCV and Dlib" shows how to use OpenCV and Dlib to create a real-time gaze detector through a webcam in Python. The article describes the basic steps of implementation, which include downloading the necessary packages, using a pre-trained network in the Dlib library to detect 68 key points that can detect eyes in real-time, and calculating the gaze direction based on the detected eye positions. The proposed system, according to the article, can be used for a variety of applications, including online proctoring, human-computer interaction, and assistive technologies. The accuracy of the algorithm is demonstrated in the article using real data under varying conditions for people with different eye shapes and sizes. Using a simple webcam and OpenCV, the proposed system can perform real-time eye tracking and analysis.

Apart from this I had explored different methodologies present to Capture the eye detection.

1. Viola-Jones Face Detection:-

Benefits:

- Effective for real-time face detection.

- It is implemented in OpenCV, which makes it easily accessible.

Disadvantages:

- May not perform well in low-light conditions.

- Detection accuracy may differ depending on the quality of the training data.

2. Haar Cascades for Eyes:

- Benefits:

- Part of the Viola-Jones framework, designed specifically for detecting eyes.

- Quick and appropriate for real-time applications.

Disadvantages:

- Susceptible to changes in lighting and head poses.

When dealing with partially occluded eyes, accuracy can suffer.

3. Eye Tracking with Pupil Detection:

Benefits:

- Enables precise tracking of eye movements.

- Useful for eye tracking and human-computer interaction.

Disadvantages:

- Accurate pupil detection necessitates the use of additional hardware (such as infrared cameras).

- Complex algorithms are potentially computationally expensive.

4. OpenCV's Built-in Eye Detection Functions:

Benefits: -

Convenient and easily accessible within the OpenCV library.

Performs admirably in a variety of scenarios.

Disadvantages:

- For specific tasks, may not be as accurate as custom-trained models and Susceptibility to lighting and occlusions.