# UNIVERSITY OF NAIROBI

## FACULTY OF SCIENCE AND TECHNOLOGY

## DEPARTMENT OF COMPUTING AND INFORMATICS

### CSC 317: ARTIFICIAL INTELLIGENCE PROGRAMMING

# DRIVER DROWSINESS DETECTION SYSTEM

**GROUP 2**

P15/141298/2020 AUSTIN ONGWAE

P15/139806/2020 CLIFF EZRA

P15/140390/2020 TED MWAI

P15/141267/2020 MELISA MICHUKI

# ABSTRACT

Driving accidents are a major source of loss of lives worldwide. A major cause of these accidents are drivers being drowsy due factors such as lack of enough sleep. If their drowsiness could somehow be detected and then the driver alerted until they are out of the drowsiness state, then many fatal accidents could be avoided. This system attempts to do just that with the use of computer vision and machine learning models to detect the drivers face and more specifically eyes and determine whether they are open or closed for a certain duration of time. This information should be sufficient to determine whether a driver is drowsy or not and if the former is true trigger a warning sound.

# CHAPTER 1: INTRODUCTION

## BACKGROUND

Majority of the most fatal accidents happen early in the morning or late in the evening. One of the major reasons is that taxi drivers, truck drivers, saloon car drivers or bus drivers ferrying people who are travelling long distances suffer from lack of enough sleep. The situation become very dangerous because the driver is in a state in which he/she is less attentive on the road. The driver is said to be drowsy, and this can be detected by the gradual closing of eyes. If this can be detected, and the driver alerted then many potential accidents can be avoided.

## PROBLEM STATEMENT

A lot of accidents on the roads are caused by drowsy drivers who have not had enough sleep or have been driving for too long during long journeys. Drowsiness causes the driver to be less alert and thus prone to making mistakes some of which can be fatal, therefore it is important that drivers get alerted out of this state.

## PROJECT OBJECTIVES

To create a system that is able to detect a driver's facial features and determine how drowsy they are.

The system should alert a driver who is found to be drowsy

The system should be easy to install on a wide range of vehicles

## PROJECT JUSTIFICATION

Drowsy drivers cannot always rely on their fellow passengers to alert them. Sometimes they are the only ones in the vehicle and therefore there is nobody to wake them up if they start drifting to sleep. Therefore, this creates a need for an autonomous and reliable system that can both detect and alert the driver when he/she is sleeping.

## PROJECT SCOPE

The scope of this project will be limited to creating the software portion of the system that can detect and alert a drowsy driver. The design and actual implementation of the vehicle hardware necessary for the system to work on a variety of vehicles will not be developed.

# CHAPTER 2. LITERATURE REVIEW

## TOP REASONS FOR VEHICLE ACCIDENT

According to wkw.com the top causes of accidents are as follows:

- Distracted driving
- Drunk Driving
- Speeding
- Reckless/Aggressive driving
- Running red lights and stop signs
- Fatigue

From their research fatigue was found to be a major cause of accidents.

## CURRENT DROWSINESS DETECTION SYSTEMS

Driver drowsiness detection systems can use cameras, eye tracking sensors and other hardware to monitor visual cues, where drowsiness can be detected through yawning frequency, eye-blinking frequency, eye-gaze movement, head movement and facial expressions. The systems can also monitor driving input behavior to notice when there are erratic steering movements, pedal use, and lane deviations.

Currently existing systems are quite experimental:

- Driver Drowsiness Detection System for vehicle safety
- Bosch’s driver drowsiness detection system
- IoT-Based Smart Alert System for Drowsy Driver Detection

Some systems fully rely on presence of specialized physical hardware like an eye blink sensor that should be worn by the driver or an alcohol breath sensor that should be installed in the vehicles.

Other systems employ the use of easily accessible hardware such as simple digital cameras to capture a live footage of the driver and the footage is then fed into a machine learning model that has been previously trained to either detect yawning, eye-gaze movement or other facial expressions that may indicate sleeping.

## LIMITATIONS OF EXISTING SYSTEMS

The existing systems are very complex and expensive in terms of money or time to implement due to the need of other specialized hardware that often need to be developed from scratch such as the eye blink sensor. Furthermore, some machine learning models used have been found to take in too many unnecessary parameters that increase the computational power needed to run the system at a speed that will be practical and effective when using cheap hardware. The idea behind this is that often at times things such as yawning and squinting of the eyes happen simultaneously therefore detecting only one of them should most of the time be sufficient.

# CHAPTER 3: METHODOLOGY

The Waterfall and Iterative Methodologies were used to ensure systematic and sequential flow of the system stages and to allow for error fixing during and after development.

## The Waterfall Model

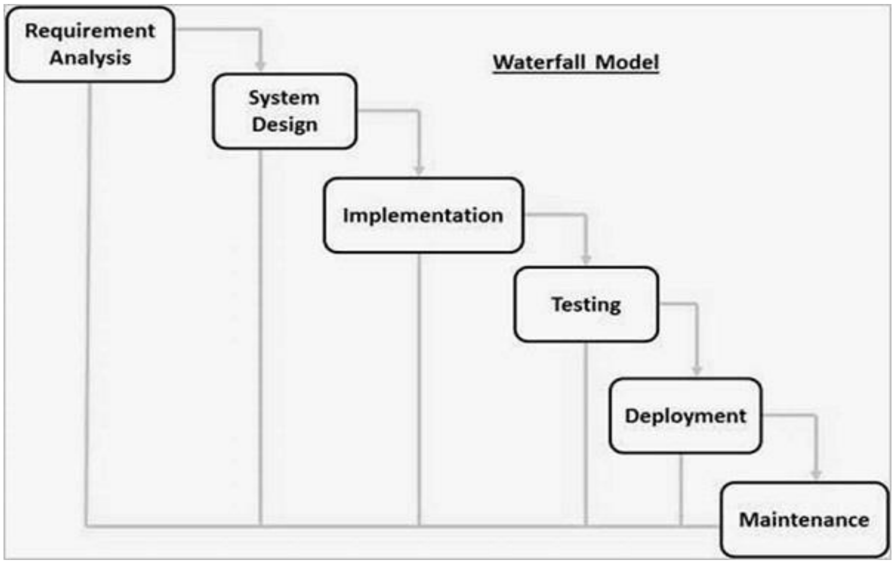


*Figure 1: Water Fall Model*

## The Iterative Methodology

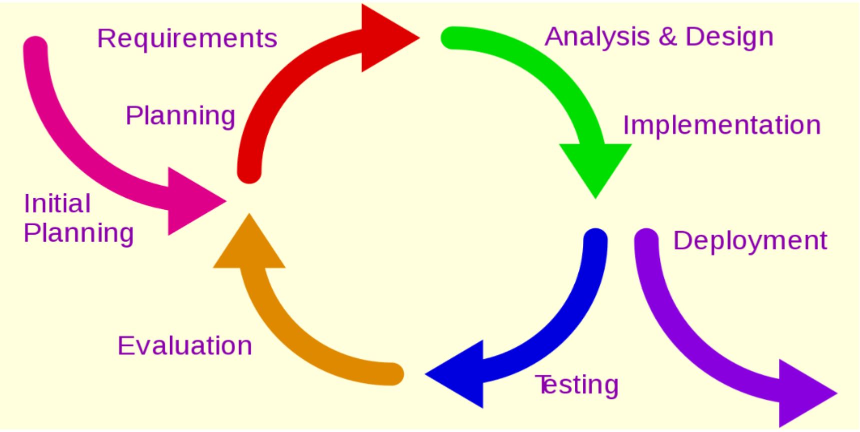


*Figure 2: An illustration of the Iterative Methodology*

This would be a suitable approach because:

1. Most requirements are clear and well understood
2. Product definition is stable
3. The project is constrained by cost and time

# CHAPTER 4: ANALYSIS AND DESIGN

## PROJECT ANALYSIS

Eliciting the system requirements during the requirements analysis stage was done mainly through researching the internet on previously built systems that attempted to solve the same problem. It was decided that using a machine learning model, specifically applying computer vision methods would be the cheapest method to solve the problem both in terms of time and resources needed.

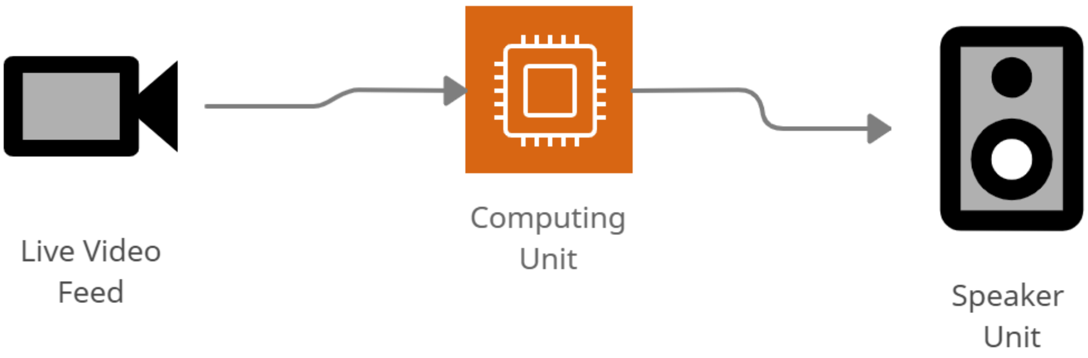
### Data collection

The dataset used to create the classifier model was found from online specifically Kaggle and consisted of images of people yawning. The aspect we were interested in was the eyes of people when they yawned.

## PROJECT DESIGN

From this analysis a simple architectural design was conceived:

The video feed is the source of data that is used by the machine learning model to perform computations and the output is via the speaker unit used to produce sound.



### Model Architecture

The model we used is built with Keras using Convolutional Neural Networks. The CNN model architecture consists of the following layers:

- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 64 nodes, kernel size 3
- Fully connected layer; 128 nodes

The final layer is also a fully connected layer with 2 nodes. A Relu activation function is used in all the layers except the output layer in which we used Softmax.

### Algorithm design

Step 1 – Capture individual frames form the live video feed and convert them to greyscale.

Step 2 – Apply computer vision algorithms using python to detect the face in the image and thus identifying a region of interest.

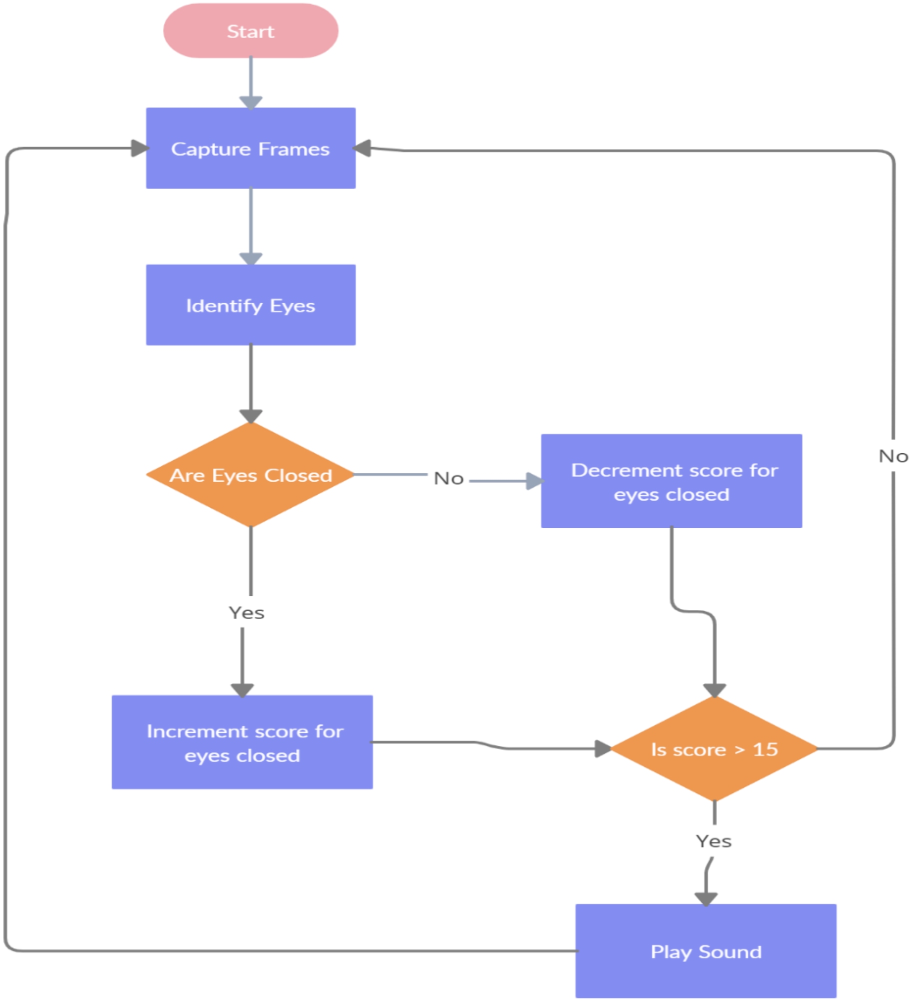
Step 3 – From this region of interest detect the eyes and feed only that portion of information to the trained classifier.

Step 4 – The classifier will then categorize whether eyes are open or closed.

Step 5 – Based on the cumulative results by the classifier from classifying the oncoming frames calculate the score to check whether the person is drowsy.

Step 6 – If the person is found to be drowsy play a sound from the speaker to alert them.

Step 7 – The steps 1 to 6 are repeated in an infinite loop



# CHAPTER 5: IMPLEMENTATION TESTING EVALUATION

## IMPLIMENTATION

The proposed system architecture was implemented using a single laptop that had an inbuilt webcam and speakers.

### Model Training

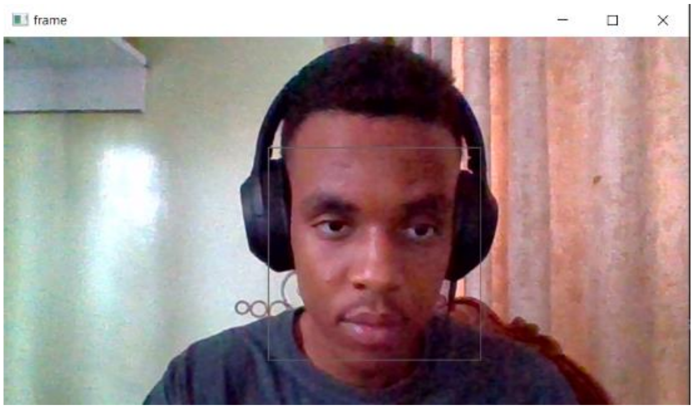
Using a separate script, we captured only the eyes from the photos in our dataset and sorted them manually according to eyes open or closed. Keras python library was used to build the convolutional neural network that was then trained.

There are three classifiers built

## FACE AND EYE DETECTION

Face detection is necessary in order to identify the eyes of a person and feed it to the classifier which will determine whether the eyes are open or closed. There are also three cascade classifiers that are used; first to detect the face, then to detect the left and right eye from the face.

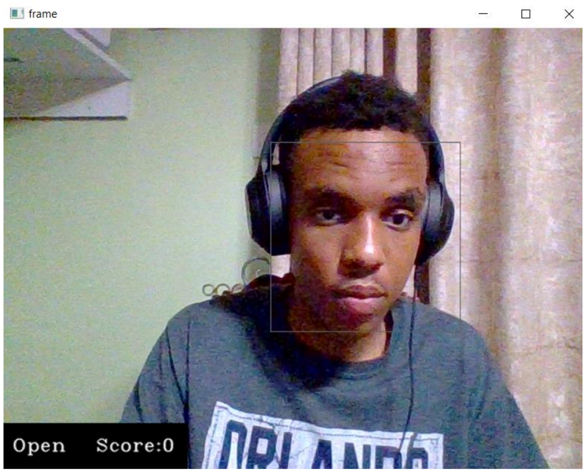
The process starts by capturing an image from a live video camera feed.



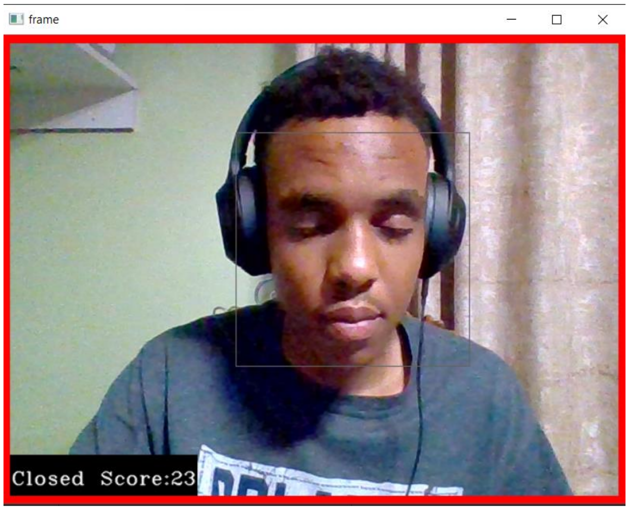
To detect the face in the image, it is first converted into grayscale because the OpenCV algorithm for object detection takes in gray images as input. This image is fed into the first cascade classifier for detecting only the face and returns a set of coordinates that are used to draw a grey square around the face. This is then passed to the left and right eye cascade classifiers respectively which return only the coordinates of both eyes respectively. The image data of both eyes are then sent to the convolutional neural network that was trained to detect whether an eye is open or closed.

## Output

Based on whether the eyes are open or closed and the duration of eyes being closed a score is calculated and if the score is greater than 15 the alarm sound is played.



Open Score:0



Closed Score:23

## Hardware Resources used

1. A computer with the following specifications:

- 40 GB free disk space
- 16 GB RAM
- 2.4 GHz intel i7 processor
- NVIDIA GeForce GTX 1650
- Inbuilt camera and speakers

## Software Resources

- A 64bit Windows 10 Operating system
- PyCharm IDE
- Python – The following python modules were used
  - OpenCv
  - TesnsorFlow
  - Keras
  - Pygame

# TESTING AND EVALUATION

During testing we found that the system did a pretty good at detecting whether a person was drowsy under normal lighting conditions.

The system slightly struggled under lower light conditions but was still impressively accurate.

It was also found that due to the implementation strategy of classifying each eye respectively, the sytem was not confused when a person closed one eye and left the other open. That was an impressive discovery.

Overall, the system worked quite well taking into account the time and resource constraints during its development

# CHAPTER 6: CONCLUSION

## Achievements of the project

We were able to successfully create a system that detected whether a person was drowsy by calculating the duration in which their eyes were closed and once confirmed alert them using sound.

The program runs smoothly without any bugs.

## Limitations of the Project

Due to the time constraints we and lack of resources, we were not able to implement the system in a vehicle and test it under various driving conditions.

## Recommendation for Future Work

It was found that there is no need for specialized equipment when developing a system that does a similar task. Development should be focused on optimization of machine learning algorithms used to reduce the overall time complexities in running them on cheap processors that will most likely be more practical to install on vehicles. It will be both cheaper in terms of time and resources if such an approach is taken.