DETECTION OF FATIGUE IN DRIVERS USING CNN

A Major Project Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

1. SYKAM GEETHA BHAVYA SRI (18B01A0585)
2. KUKKADAPU VYSHNAVI TANUJA (18B01A0594)
3. NALADHA JEESHNA SARVANI (18B01A05A5)
4. PANJA TULASI LAKSHMI (18B01A05A8)
5. PULAVARTHI PRANATHI (18B01A05B5)

Under the esteemed guidance of
Mrs. T. Gayathri(Ph.d)

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)
(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202

2021 – 2022

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)
(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the Major Project entitled "**DRIVER'S FATIGUE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS**", is being submitted by **S. Geetha Bhavya Sri, K. Vyshnavi Tanuja, N. Jeeshna Sarvani, P. Tulasi Lakshmi, P. Pranathi** bearing the **Regd. No. 18B01A0585, 18B01A0594, 18B01A05A5, 18B01A05A8, 18B01A05B5** in partial fulfilment of the requirements for the award of the degree of "**Bachelor of Technology** in **Computer Science & Engineering**" is a record of bonafide work carried out by them under my guidance and supervision during the academic year **2021–2022** and it has been found worthy of acceptance according to the requirements of the university.*

Internal Guide **Head of the Department**

ACKNOWLEDGMENTS

The satisfaction that accompanies the successful completion of any task would be incomplete without mention of the people who made it possible and those who constant encouragement and guidance has been a source of inspiration throughout the course of this project. We take this opportunity to express our gratitude to all those who helped us in this project.

We wish to place our deep sense of gratitude to **Sri. K. V. Vishnu Raju, Chairman of SVES** for his constant support on each and every progressive work of us.

Our deep sense of gratitude and sincere thanks to **Dr. G. Srinivasa Rao, Principal of SVECW** for being a source of an inspirational constant encouragement

Our deep sense of gratitude and sincere thanks to **Dr. P. Srinavasa Raju, Vice-Principal of SVECW** for being a source of an inspirational constant encouragement

We deeply indebted and sincere thanks to **Dr. P. Kiran Sree, Head of the department** for his valuable advices in completing this project successfully

Our deep sense of gratitude and sincere thanks to **Mrs.T.Gayathri(Ph.D.), Assistant Professor** for his unflinching devotion and valuable suggestion throughout this project.

Project Associates

S.Geetha Bhavya Sri (18B01A0585)
K. Vyshnavi Tanuja(18B01A0594)
N. Jeeshna Sarvani (18B01A5A5)
P. Tulasi Lakshmi (18B01A05A8)
P. Pranathi (18B01A05B5)

ABSTRACT

Many of the accidents occur due to drowsiness of drivers. It is one of the critical causes of roadways accidents now-a-days. Latest statistics say that many of the accidents were caused because of drowsiness of drivers. Vehicle accidents due to drowsiness in drivers are causing death to thousands of lives. More than 30% accidents occur due to drowsiness. For the prevention of this, a system is required which detects the drowsiness and alerts the driver which saves the life. In this project, we present a scheme for driver drowsiness detection. In this, the driver is continuously monitored through webcam. This model uses image processing techniques which mainly focuses on face and eyes of the driver. The model extract the drivers face and predicts the blinking of eye from eye region. We use an algorithm to track and analyze drivers face and eyes. If the blinking rate is high then the system alerts the driver with a sound.

Keywords — Drowsiness, Distraction, Eye detection, Eye Tracking, Face Detection.

CONTENTS

1. INTRODUCTION

Drowsiness of the drivers is one of the key issues for majority of road accidents. Drowsiness threatens the road safety and causes severe injuries sometimes, resulting in fatality of the victim and economical losses. Drowsiness implies feeling lethargic, lack of concentration, tired eyes of the drivers while driving vehicles. Most of the accidents happen in India due to the lack of concentration of the driver. Performance of the driver gradually deteriorates owing to drowsiness. To avoid this anomaly, we developed a system that is able to detect the drowsiness nature of the driver and alert him immediately. This system captures images as a video stream through a camera, detects the face and localizes the eyes. The eyes are then analysed for drowsiness detection using CNN algorithm. Based on the result, the driver is alerted for drowsiness through an alarm system.

1.1. DIFFERENT APPROACHES TO DETECTING DROWSINESS:

There are different approaches to identify drowsiness state of the driver. They can be categorised into the following three main categories:

1.1.1 BEHAVIOURAL PARAMETERS-BASED TECHNIQUES: Measuring the driver’s fatigue without using non-invasive instruments comes under this category. Analysing the behaviour of the driver based on his/her eye closure ratio, blink frequency, yawning, position of the head and facial expressions. The current parameter used in thissystem isthe eye-closure ratio of the driver.

1.1.2 VEHICULAR PARAMETERS-BASED TECHNIQUES: Measuring the fatigue nature of the driver through vehicle driving patterns comes under this category. These parameters include lane changing patterns, steering wheel angle, steering wheel grip force, vehicle speed variability and many more.

1.1.3 PHYSIOLOGICAL PARAMETERS-BASED TECHNIQUES:

Measuring the drowsiness of the driver based on the physical conditions of the driver fall under this category. Such parameters may be respiration rate, heart-beat rate, body temperature and many more. Among other various approaches, these physiological parameters provide the most accurate results since they are based on the biological nature of the driver. All the above approaches have their own advantages and disadvantages. Based on the desired result accuracy, any approach can be used. Physiological approach includes wearing of the equipment on the driver’s body.

This equipment includes electrodes to detect the pulse rate of the driver which might make the driver uncomfortable while driving. This also can’t be assured that the driver always wears such equipment while driving which may result in inefficient results. Hence there is a hindrance using the physiological approach. Vehicular- based approach is always based on the efficiency of the driver and his condition. There are also constraints like the road condition and the type of vehicle which may change regularly. Hence it is best to follow the behavioural based approach through visual assessment of the driver from a camera. There shall be no equipment attached to the driver. Hence this technique is always the best approach and can be implemented in any vehicle without any modifications.

1.1.4 DIGITAL IMAGE PROCESSING

The term digital image processing generally refers to processing of a two dimensional picture by a digital computer. In a broader context, it implies digital processing of any two- dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision.

Pixel:

Pixel is the smallest element of an image. Each pixel corresponds to any one value. In an 8-bit gray scale image, the value of the pixel between 0 and 255. The values of a pixel at any point correspond to the intensity of the light photons striking at that point. Each pixel stores a value proportional to the light intensity at that particular location.

Digital image:

A digital image is nothing more than data numbers indicating variations of red, green, and blue at a particular location on a grid of pixels.

Gray level:

The value of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level. Generally to convert an image to gray scale, the equation that was used previously is : Grayscale = (Red + Green + Blue / 3).But as red has more wavelength we use the equation:

Grayscale = ((0.3 * R) + (0.59 * G) + (0.11 * B)). -- Eq: (1)

1.2 MOTIVATION FOR THE WORK

Now-a-days, there is huge increase in private transportation day by day in this modernize world. It will be tedious and bored for driving when it is for long time distance.

One of the main causes behind the driver’s lack of alertness is due to long time travelling without sleep and rest. Tired driver can get drowsy while driving. Every fraction of seconds drowsiness can turn into dangerous and life-threatening accidents may lead to death also. To prevent this type of incidents, it is required to monitor driver’s alertness continuously and when it detects drowsiness, the driver should be alerted. Through this we can reduce significant number of accidents and can save lives of people.

1.3 PROBLEM STATEMENT

Many of the road accidents will occur due to drowsiness of the driver. Drowsiness can be detected by monitoring the driver through continuous video stream with a mobile or camera. The general objective is to create a model that will indicate whether a person is feeling drowsy or not. The model takes image for every second and check for eye blinking and calculate the time for eye closed by perclos algorithm. If the blinking is high and eye is closed for certain amount of time then it will indicate driver through a sound.

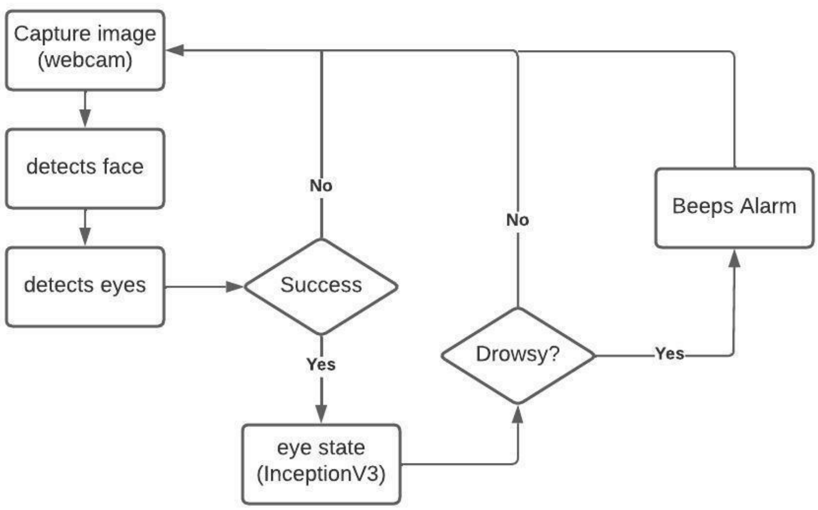
2. SYSTEM ANALYSIS

2.1 Existing System

The current drowsiness detection systems include the usage of the devices that detect the respiration rate, heart rate, blood pressure, etc. These devices can cause the driver to be uncomfortable for driving. Cannot be assured that the drivers wear these devices all the time while driving. May get lost or improper functioning which may lead to low accuracy in the result. The existing system does not produce good results in low light conditions. If the light conditions are dark or too low it is unable to detect the face and eyes of the driver which results in lower accuracy.

2.2 Proposed System

2.2.1 Architecture:



This is the architecture for detecting the drowsiness of the driver. First of all the system captures images through the webcam and after capturing it detects the face through haar cascade algorithm. It uses haar features which can detect the face. If the system founds it as face the it will proceed for next phase i.e eye detection. The eye is also detected using haar cascade features and it is used for blink frequency. The state of eye will be detected using perclos algorithm. Through this algorithm we can find the percentage of time the eye lids remains closed.

If it found eyes in closed state then it detects driver in drowsy state and alerts him by an alarm. In some cases distraction can be measured by continuous gazing. The drivers face is analysed continuously to detect any distraction. If found then alarm is activated by the system.

2.3 Feasibility Study

Generally the feasibility study is used for determining the resource cost, benefits and whether the proposed system is feasible with respect to the organization. The proposed system feasibility could be as follows. There are six types of feasibility which are equally important are:

2.3.1 Technical feasibility
2.3.2 Economic feasibility
2.3.3 Behavioural feasibility

2.3.1 Technical Feasibility

Technical feasibility deals with the existing technology, software and hardware requirements for the proposed system. The proposed system "Hand Gesture Detection" is planned to run on python. Thus, the project is considered technically feasible for the development. The work for the project can be done with current equipment, existing software technology and available personnel. Hence the proposed system is technically feasible.

2.3.2 Economic Feasibility

This method is most frequently used for evaluating the effectiveness of a python. It is also called as benefit analysis. In this project "Hand Gesture Detection" is developed on current equipment, existing software technology Since the required hardware and software for developing the system is already available in the organization, it does not cost must developing the proposed system.

2.3.3 Behavioural Feasibility

This project has been implemented by python and it satisfies all conditions and norms of the organization and the users. This proposed system "Hand Gesture Detection " Application has much behavioral feasibility because users are provided with a better facility.

3 SYSTEM REQUIREMENTS SPECIFICATION

3.1 Software Requirements

These are the software requirements for running this project.

- **Operating System:** Windows 10/8/7 (incl. 64-bit), Mac OS, Linux
- **Language:** Python 3
- **IDE:** Jupyter Notebook

3.2 Hardware Requirements

- **Processor:** 64 bit, quad-core, 2.5 GHz minimum per core
- **RAM:** 4 GB or more.
- **HDD:** 20 GB of available space or more.
- **Display:** Dual XGA (1024 x 768) or higher resolution monitors.
- **Camera:** A detachable webcam.
- **Keyboard:** A standard keyboard.

4 SYSTEM DESIGN

4.1 Introduction

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

System Analysis is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blue print or plan for the solution, and is used later during implementation, testing and maintenance.

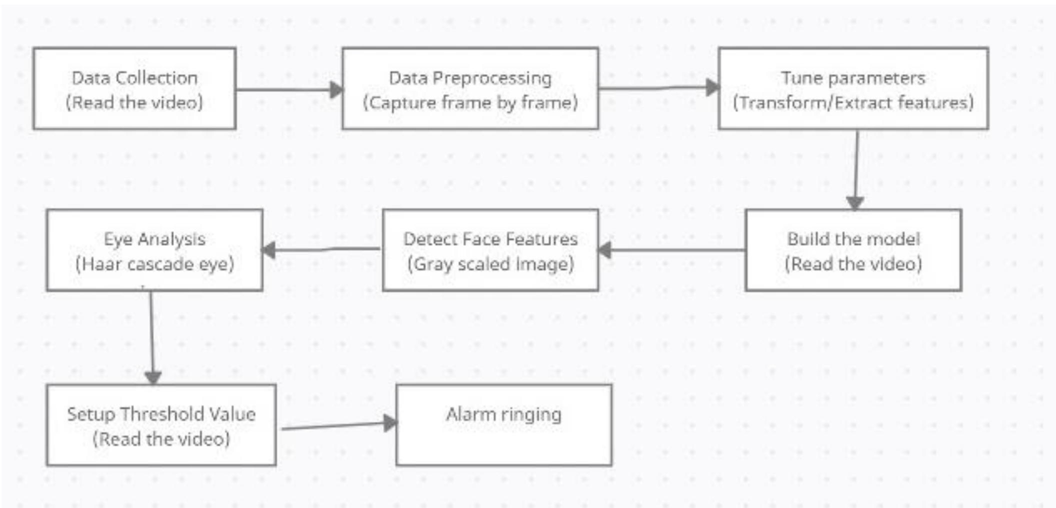
The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction. A large system cannot be handled as a whole, and so for design it’s partitioned into smaller systems. Abstraction is a concept related to problem partitioning. When partitioning is used during design, the design activity focuses on one part of the system at a time. Since the part being designed interacts with other parts of the system, a clear understanding of the interaction is essential for property designing the part.

4.1.1 System Model :

The framework is created utilizing the incremental model. The center model of the framework is first created and afterwards augmented in this way in the wake of testing at each turn. The underlying undertaking skeleton was refined into expanding levels of ability. At the following incremental level, it might incorporate new execution backing and improvement.

Step1: Capture the video frame by frame
Step2: Detect the face in the image and create a Region of Interest
Step3: Transform the image from one state to another state.
Step4: Detect the eyes and categorize whether eyes are open or closed.
Step5: Calculate score to check whether eyes open/close.
Step6: Alert by ringing alarm if eyes close and wrap the video and continues.



4.2. UML Diagrams

UML Diagrams is a rich visualizing model for representing the system architecture and design. These diagrams help us to know the flow of the system.

Some of them are:

- Use case diagram
- Class diagram
- Sequence diagram
- Activity diagram

USE CASE DIAGRAMS

A Use Case Diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Interaction among actors is not shown on the use case diagram. If this interaction is essential to a coherent description of the desired behavior, perhaps the system or use case boundaries should be re-examined. Alternatively, interaction among actors can be part of the assumptions used in the use case.

Use cases:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors:

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

System boundary boxes:

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

Four relationships among use cases are used often in practice.

Include:

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behaviour of the included use case is inserted into the behaviour of the including use case.

The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviours from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»".There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behaviour of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

Extend:

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behaviour of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". Modellers use the «extend» relationship to indicate use cases that are "optional" to the base use case.

Generalization:

In the third form of relationship among use cases, a generalization/specialization relationship exists. A given use case may have common behaviours, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, describing any differences in the specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation

Associations:

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

Identified Use Cases

The "user model view" encompasses a problem and solution from the preservative of those individuals whose problem the solution addresses. This view is composed of "use case diagrams". These diagrams describe the functionality provided by a system to external integrators. These diagrams contain actors, use cases, and their relationships.

Actors :

- ML Model
- Driver
- Camera
- Facial Processing

Use Cases for ML Model :

- Data Collection
- Preprocess the data
- Eye status analyzing
- Feature selection
- Eye Variable Count
- Drowsy status detection
- Alarm running

Use Case for Driver :

- Start camera
- Stop camera

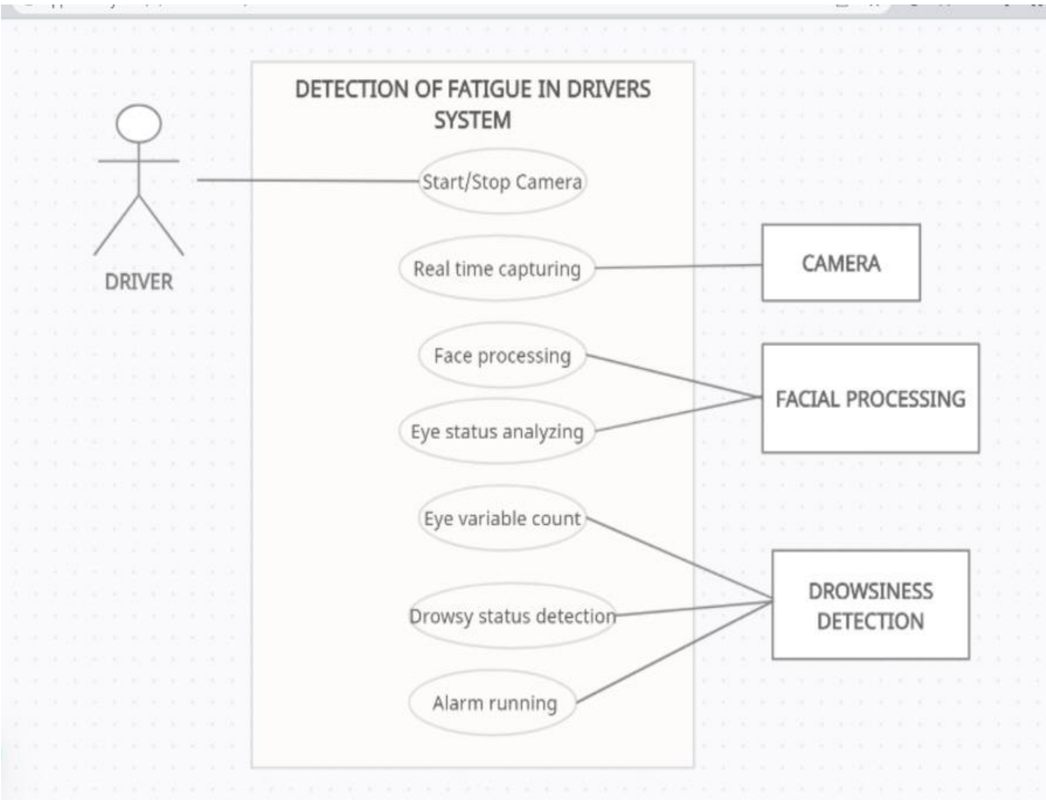
Use Case for Facial Processing :

- Face processing
- Eye status analyzing

Use Case for Camera :

- Real time capturing

USE CASE DIAGRAM FOR Detection Of Fatigue In Drivers:



CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects.

Purpose of Class Diagrams

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed by UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business

  A UML class diagram is made up of:
- A set of classes and
- A set of relationships between classes

CLASS NAMES: Driver,Vehicle,Detection_System,Analysis_System,Alarm,Decision_System.

ATTRIBUTES:

For Driver: DrivingLicenseID

For Vehicle: VehicleType

For Detection_System : WebCam

For Analysis_System : ImageProcessor

For Alarm : SoundSystem.

OPERATIONS:

For Driver: DrivingVehicle(),BlinkingEye()

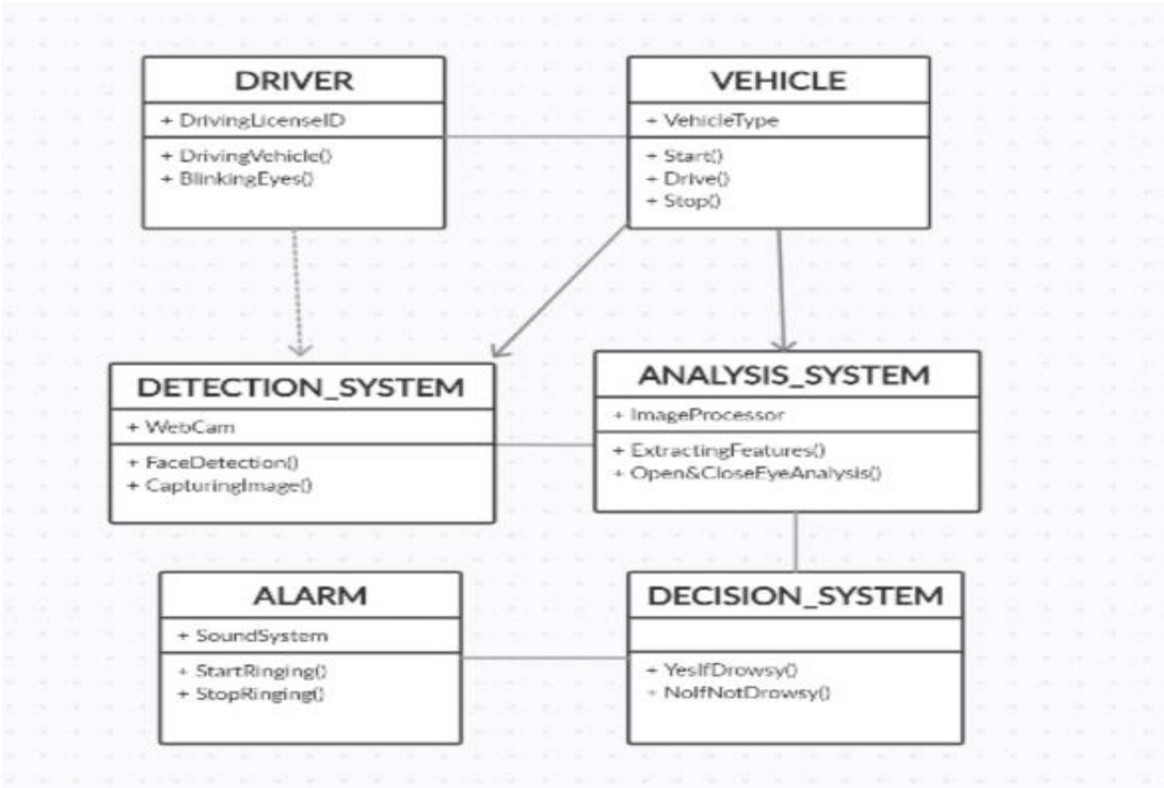
For Vehicle: Start(),Drive(),Stop()

For Detection_System : FaceDetection(),CapturingImage()

For Analysis_System : ExtractingFeature(),Open&CloseEyeAnalysis()

For Alarm : StartRinging(),StopRinging()

CLASS DIAGRAM FOR Detection Of Fatigue In Drivers:



SEQUENCE DIAGRAMS:

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagram captures:

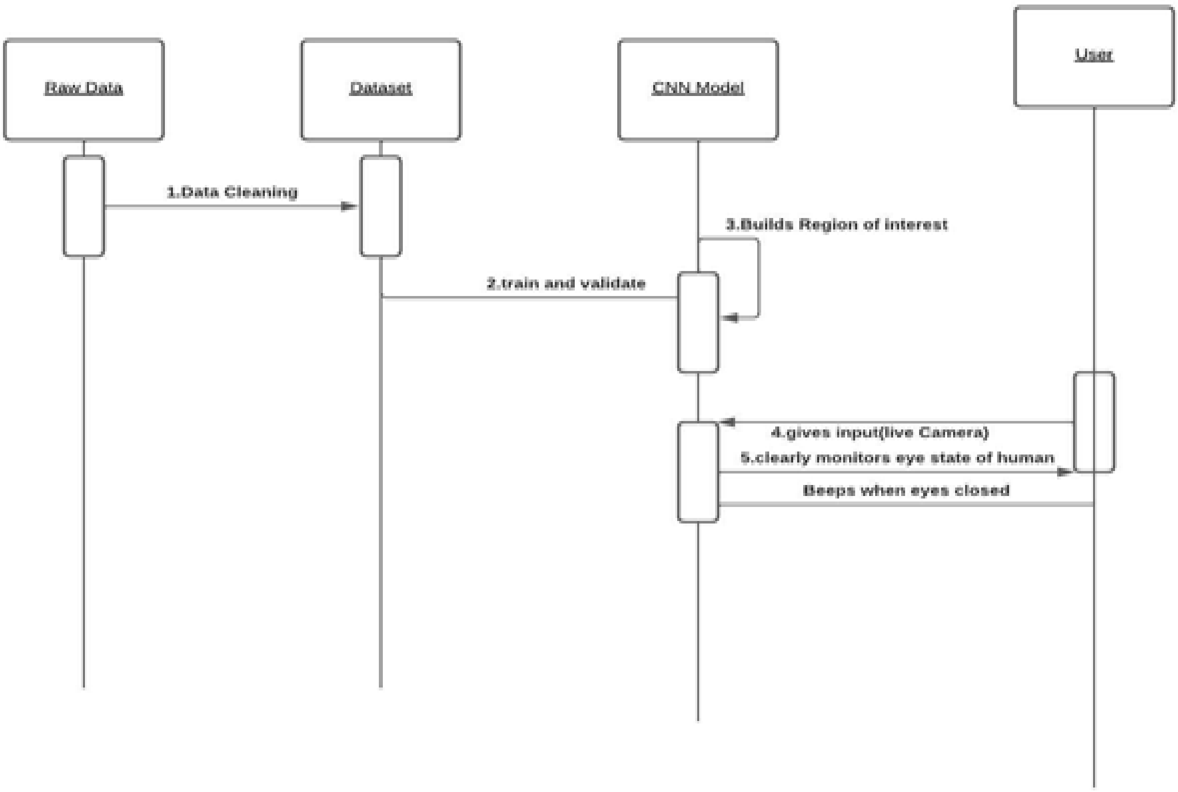
- the interaction that takes place in a collaboration that either realizes a use case or an operation.
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems.

Purpose of Sequence Diagram

- Model the high-level interaction between active objects in a system.
- Model the interaction between object instances within a collaboration that realizes a use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions or specific instances of interaction.

OBJECTS FOR LANE DETECTION SYSTEM: User, Data Set, Refined Data, ML Model.

SEQUENCE DIAGRAM FOR Detection Of Fatigue In Drivers:



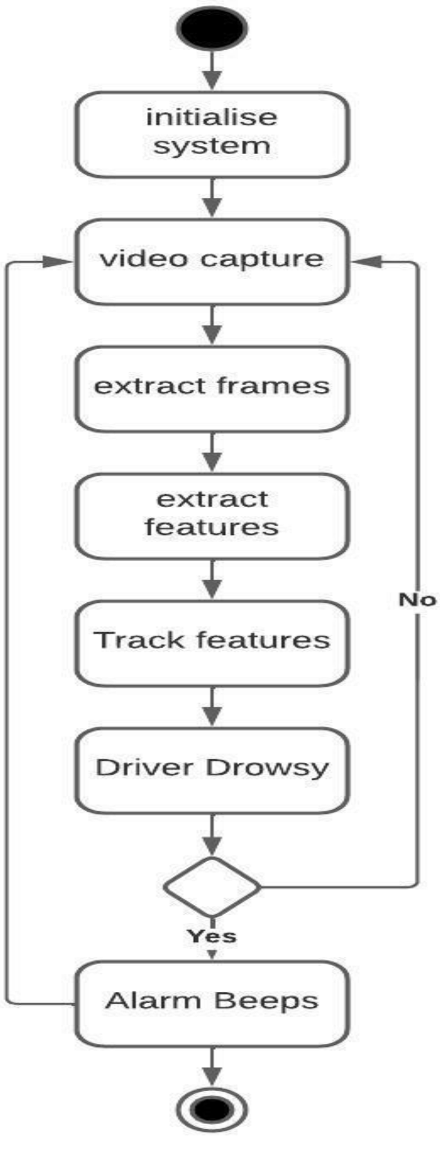
ACTIVITY DIAGRAMS:

Activity diagram is another important behavioral diagram in uml diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

Purpose of Activity Diagrams

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

ACTIVITY DIAGRAM FOR Detection Of Fatigue In Drivers:



5 Methodology

5.1 MODULAR DIVISION:

The entire architecture is divided into 6modules.

1. Face Detection
2. Eye Detection
3. Face Tracking
4. Eye Tracking
5. Drowsiness Detection
6. Distraction Detection

Face Detection:

This module takes input from the camera and tries to detect a face in the video input. The detection of the face is achieved through the Haar classifiers mainly, the Frontal face cascade classifier. The face is detected in a rectangle format and converted to grayscale image and stored in the memory which can be used for training the model.

Eye Detection:

Since the model works on building a detection system for drowsiness we need to focus on the eyes to detect drowsiness. The eyes are detected through the video input by implementing a haar classifier namely Haar Cascade Eye Classifier. The eyes are detected in rectangular formats

Face Tracking:

Due to the real-time nature of the project, we need to track the faces continuously for any form of distraction. Hence the faces are continuously detected during the entire time

Eye Tracking:

The input to this module is taken from the previous module. The eyes state is determined through CNN algorithm.

Drowsiness detection:

In the previous module the frequency is calculated and if it remains 0 for a longer period then the driver is alerted for the drowsiness through an alert from the system.

Distraction detection:

In the face tracking module the face of the driver is continuously monitored for any frequent movements or the long gaze of the eyes without any blinks which can be treated as lack of concentration of the driver and is alerted by the system for distraction.

5.1.1 Haar Cascade

Haar Cascade is based on the concept of features which are proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It can be used to detect objects from an image or a video.

This algorithm comprises of four stages:

i. Haar Feature Selection

ii. Creating Integral Images

iii. Adaboost Training

iv. Cascading Classifiers

Though Haar Cascade is used for detecting almost all objects, it is popular for detecting faces in images. Adaboost which both selects the best features and trains the classifiers that use them. This algorithm constructs a "strong" classifier as a linear combination of weighted simple "weak" classifiers.

A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the intensities of the pixels in each region and calculates the difference between these sums. During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because each Haar feature is only a "weak classifier" i.e. its detection quality is slightly better than random guessing and a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore they are organized into cascade classifiers to form a strong classifier.

Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector.

A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, with a human face, it is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, from the

fig 3.2, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object.

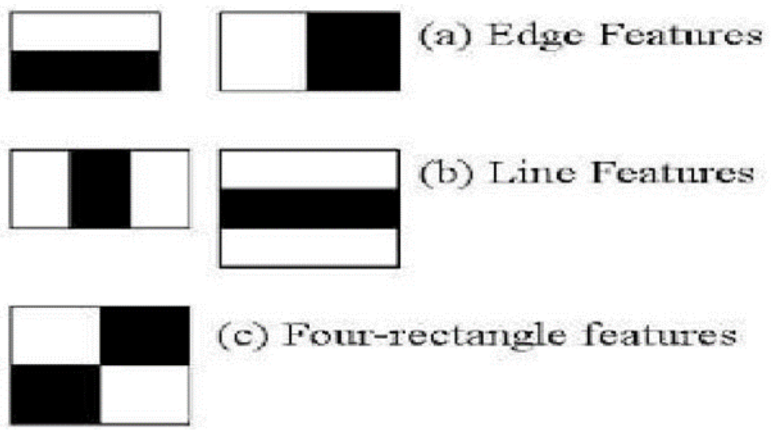


Fig 3.2 Different types in feature extraction

In the detection phase of the viola jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier a large number of Haar- like features are necessary to describe an object with sufficient accuracy. In the Viola–Jones object detection framework, the Haar-like features are therefore organized in something called a *classifier cascade* to form a strong learner or classifier. The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of integral images a Haar-like feature of any size can be calculated in constant time.

Integral image is a data structure and algorithm for generating the sum of values in a rectangular subset of a grid. The goal is reducing the number of computations needed to obtain the summations of pixel intensities within a window. The idea is transforming an input image into a summed-area table, where the value at any point (x, y) in that table is the sum of all the pixels above and to the left of (x, y), inclusive:

$$I(x,y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \qquad \text{----- Eq: (3)}$$

Where I(x,y) is the value of the integral image pixel in the position (x,y), while i(x,y) is the corresponding intensity in the original image. It is a recursive formula, hence, if we start from one corner of the input image, we will have the same result in the integral image.

1	3
7	9

0	0	0
0	1	4
0	8	20

Fig 3.3 Calculation of integral image from an actual image

We add one row and column of zeros, since we need one step backward in order to start the recursive formula. Hence, if actual image is w pixels wide and h pixels high, then the integral of this image will be w+1 pixels wide and h+1pixels high

Moving to the computations, let's start from the first pixel in the original image from fig. 3.3 with intensity 1: the integral image returns exactly the same value, since it is computing (1+0+0). Then, pixel '3' becomes '4', since it is 3+1+0+0. With the same procedure, we obtain an '8' (7+1+0) and a '20' (9+3+1+7).

Now, we have a new image. This image is useful in an unique property of the integral image. Indeed, it turned out that if you need to compute the summation within a window in the input image, hence that summation is equal to a linear combination of the

corresponding window's corner in the integral image, as follows:

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} i(x,y) = I(D) + I(A) - I(B) - I(C)$$

------ Eq: (4)

Where A, B, C and D are the corners of the corresponding window in the integral image of fig. 3.4.

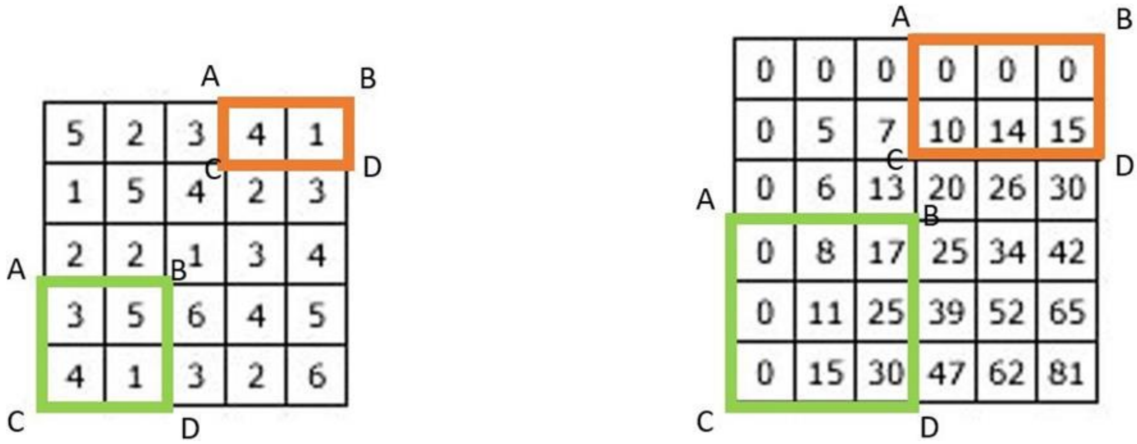


Fig 3.4 Converting a 5x5 image to a 6x6 integral image

This reduces the number of computations. To give you an idea, consider a 100×100 image with a 9×9 window. We want to compute the sum of the pixel intensities within that window, which requires 8 operations. If we repeat this procedure 100 times, we obtain 800 operations.

Now let's see the integral image approach. First, we compute the summed-area table, which requires 56 operations. Then, considering the same 9×9 window, to compute the sum of pixel intensity we just need the above formula, which is made of 3 operations. Hence, the total number of operations is 56+3*100=356. As you can see, it is less than a half.

This procedure is widely used in computer vision and Haar Cascade algorithm is based exactly on that.

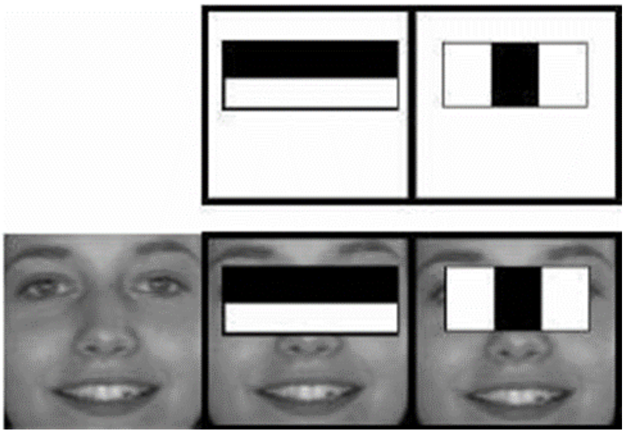


Fig 3.5 Extracting different features

The fig 3.5 is the cascade classifiers for the eyes region. In second picture eyes region is in black colour or both the eyes will have white colours. If the system founds the same pixel intensity at both the places then it detects them as the eyes. The detection is based on the intensity of the pixels around a particular object. The eyes region is white in colour and the region around the eye is black in colour. If there are white pixels found then it detects the region as eye.

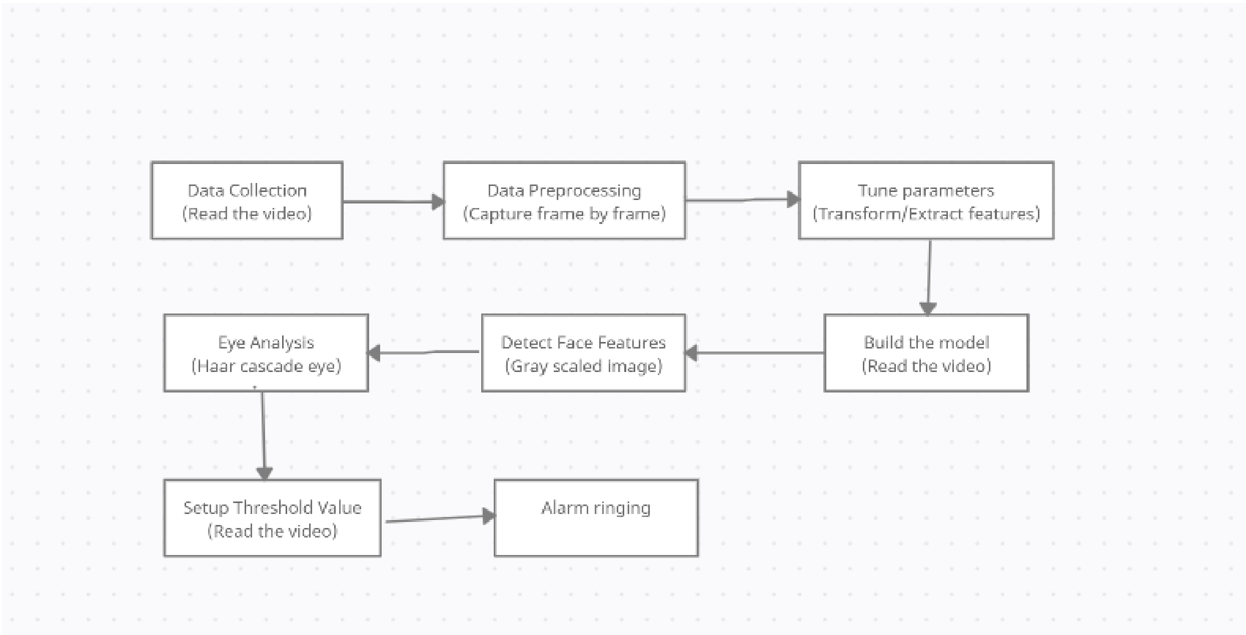
Selecting the most relevant features is performed through Adaboost technique which selects the best features and trains the classifiers that use them. This algorithm uses “Haar Cascade Frontal Face” classifier for detecting the faces since we need to detect only the frontal part of the face.

6 SYSTEM IMPLEMENTATION

6.1. Introduction

Fatigue detection system is regarded as an effective tool to reduce the number of road accidents. This project proposes a non-intrusive approach for detecting fatigue in drivers, using Computer Vision. This system tracks the eye for drowsiness.

Step1: Capture the video frame by frame
Step2: Detect the face in the image and create a Region of Interest
Step3: Transform the image from one state to another state.
Step4: Detect the eyes and categorize whether eyes are open or closed.
Step5: Calculate score to check whether eyes open/close.
Step6: Alert by ringing alarm if eyes close and wrap the video and continues.



6.2 Project Modules

Data Collection :

- Read the video.

Preprocessing :

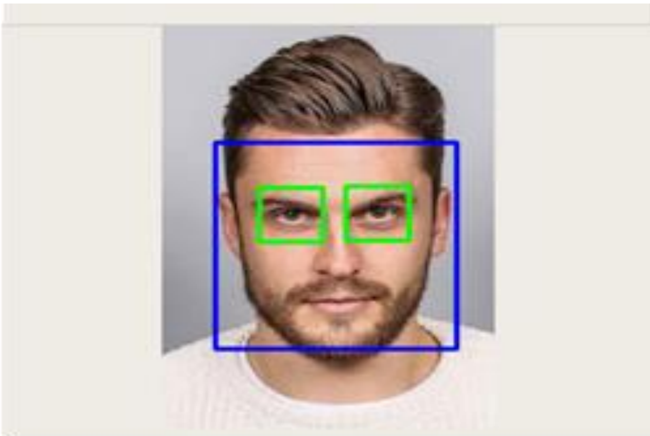
- Capture video frame by frame
- Face analyzation.

Haar cascade_eye Analysis:

- Eye Analysis.
- Grey scaling

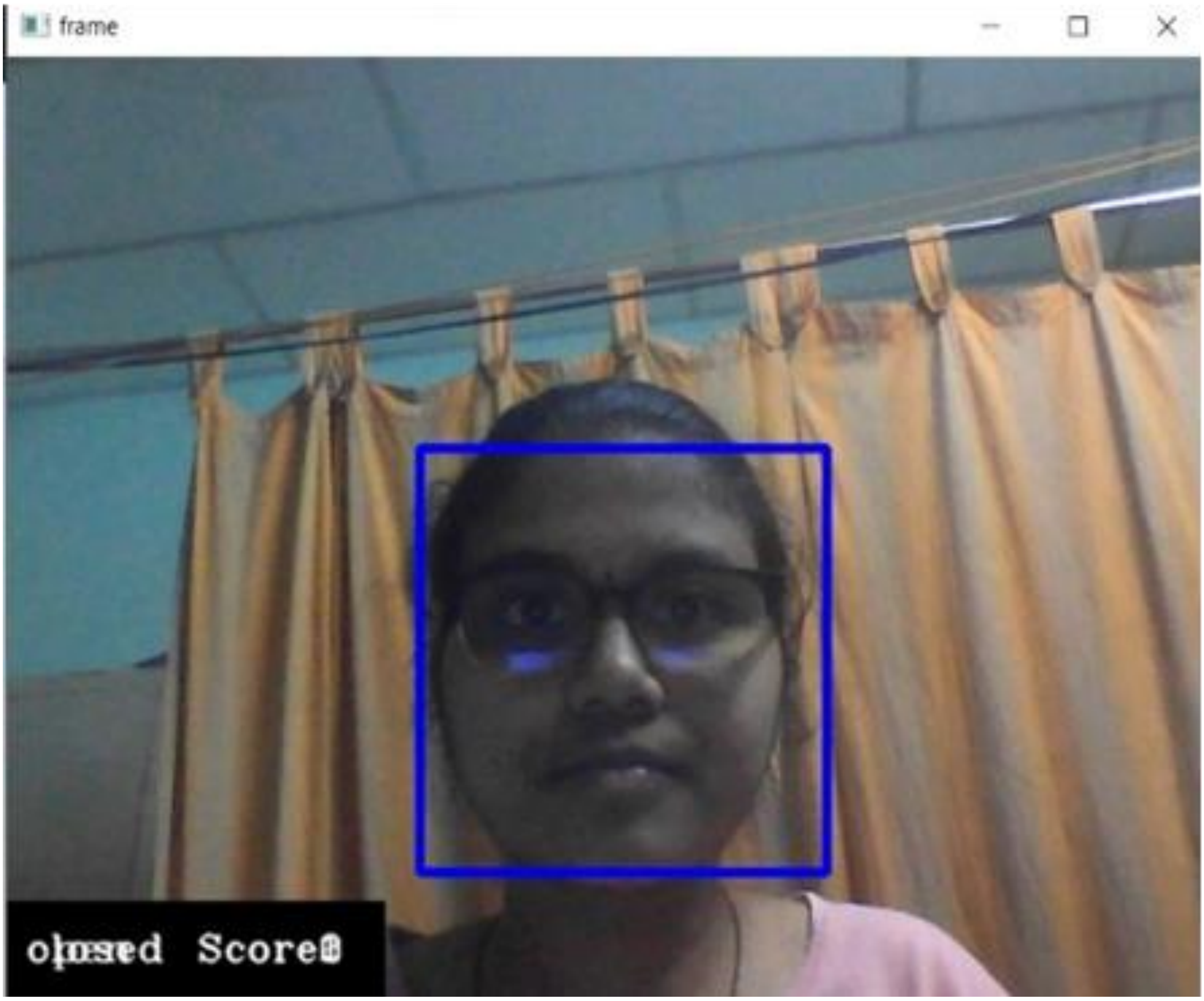
Model Training:

- Determine eyes open/close
- Alarm Ring

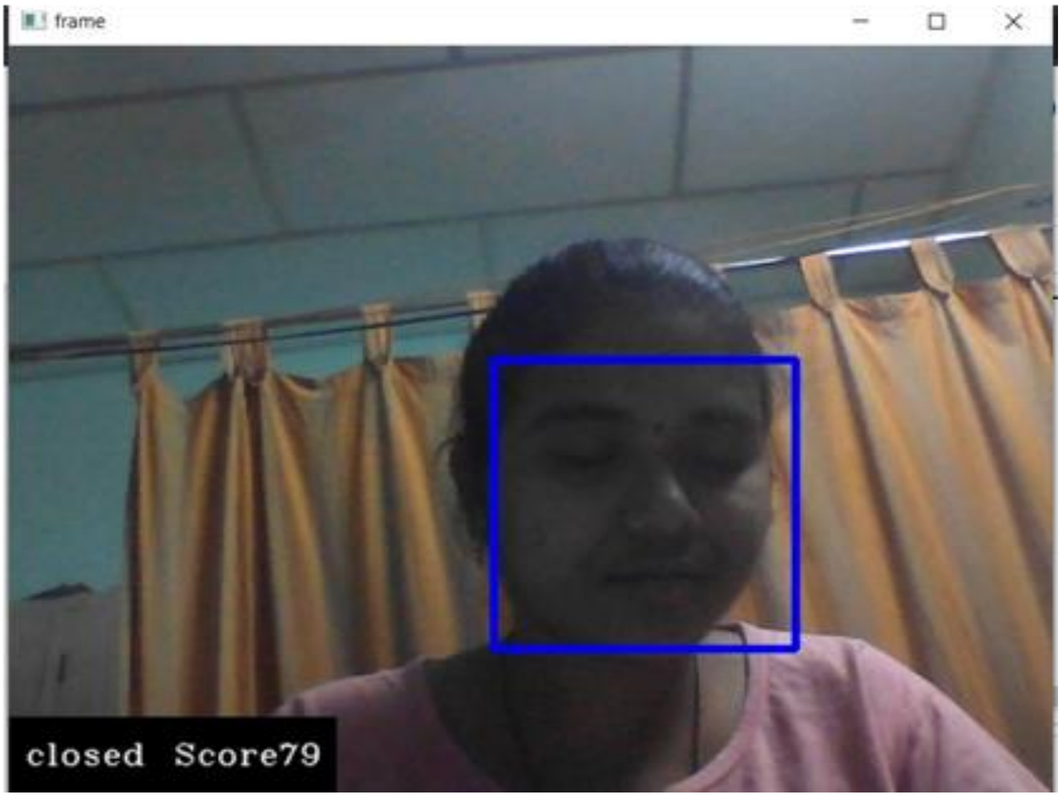


6.3. Screens

When the eyes are opened:

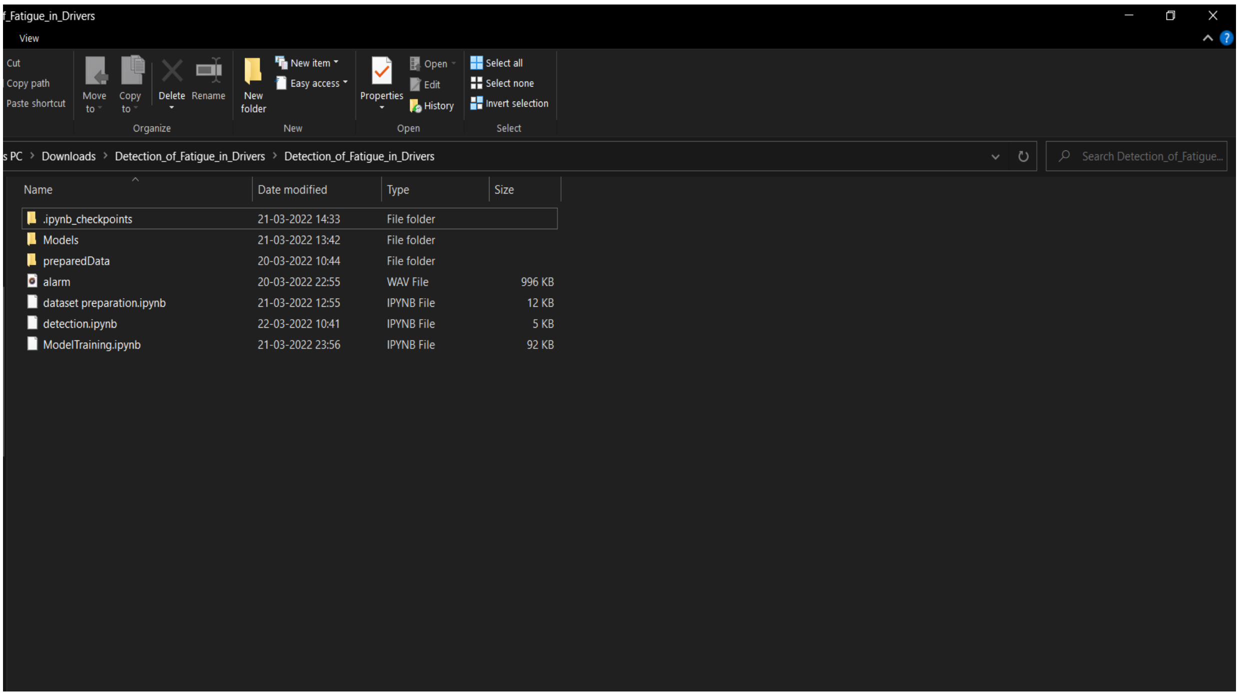


When the eyes are closed:

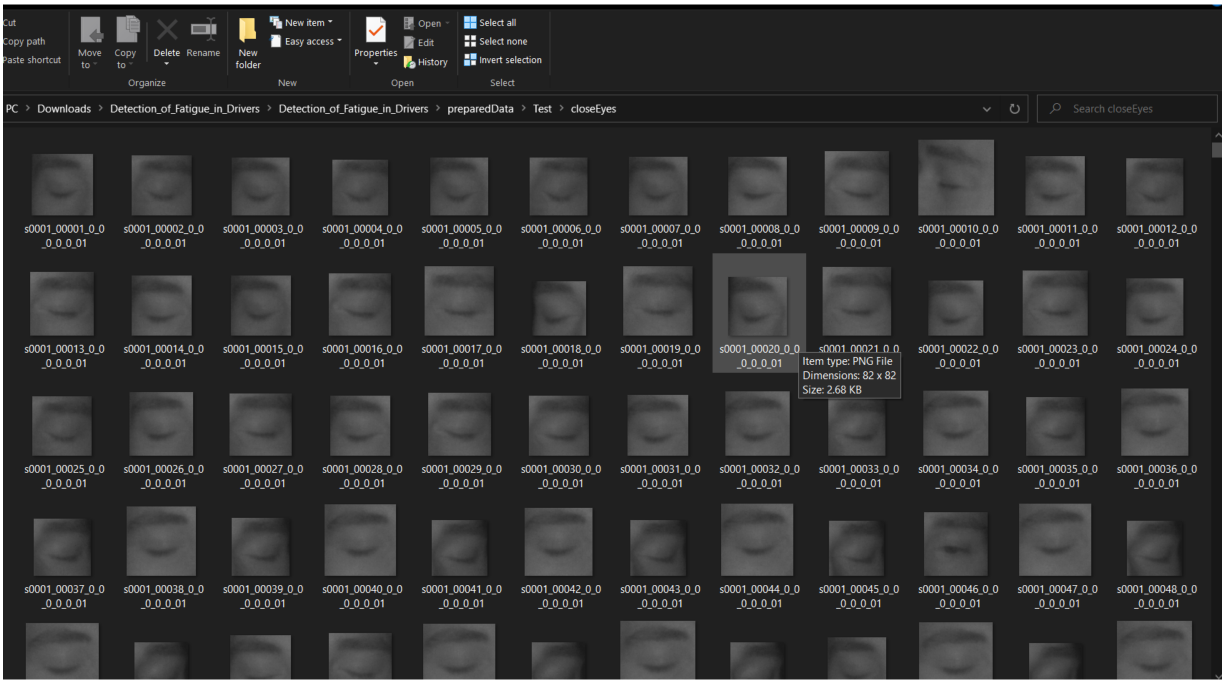


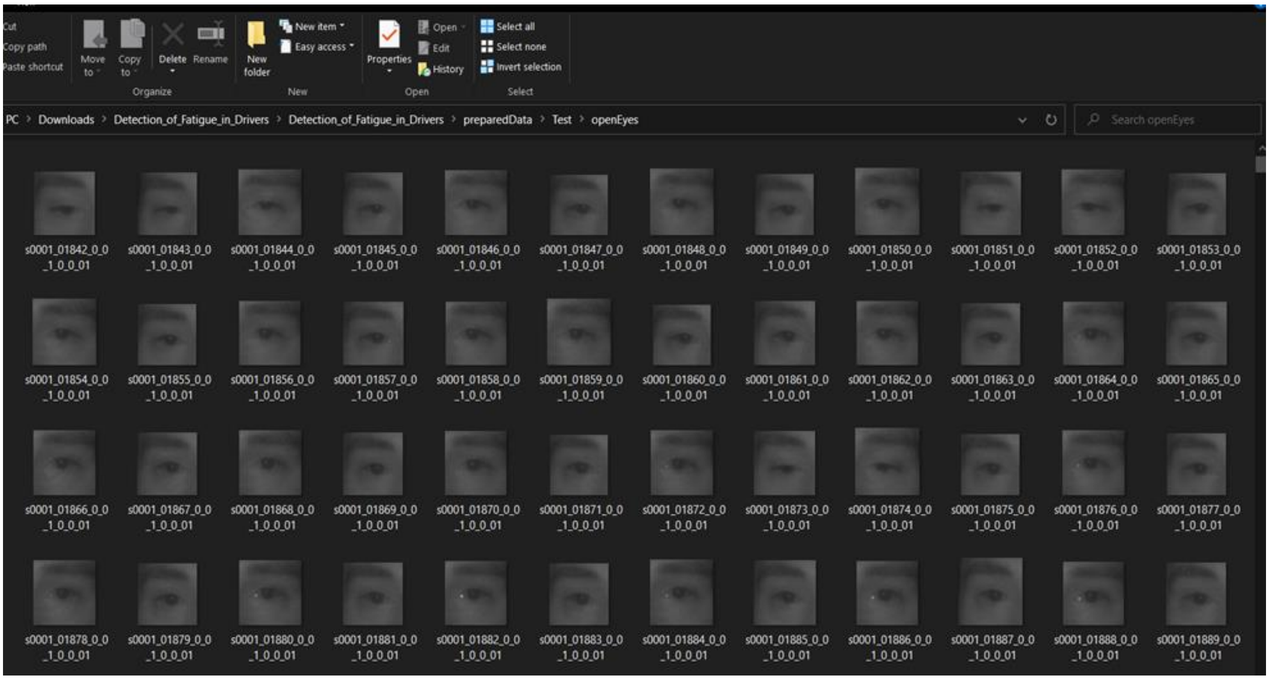
Folders Hierarchy:

- Dataset
- Detection.py
- Models – best Model will be downloaded with .h5 extension



Data Set Images:





7. SYSTEM TESTING

7.1. Introduction:

Software Testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for III planned through testing.

TESTING OBJECTIVES

These are several rules that can save as testing objectives:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

TEST LEVELS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or darkness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2. Testing Methods

Unit Testing

Unit testing involves the design of test cases that validate the internal program logic is functioning properly, and that program inputs procedure valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is structural testing that relies on knowledge of its construction and is invasive. Unit test perform basic test at component level and and test a specific business, application and/or system configuration Unit test ensures that each unique path of princess performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing in event driven and more concerened with the basic outcome of screens or fields. Integration test demonstrate that although the components were individually satisfaction, as shown successfully by unit testing the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination, of components.

Functional Testing

Functional test provide systematic demonstrations that function tests are available as specified by the business and technical requirement, system documentation and user manuals. Functional testing is centered on the following items: Valid input: identify classes of valid input must be accepted, Invalid Input: Identify classes of invalid inputs must be rejected, Functions: Identified be exercised identities function must be exercised, Output: identify classes of application outputs must be exercised, Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functions test is focused on requirements, key functions or special test cases. In addition, systematic coverage pertaining to identify business process flow; data fields, predefined process and successive processmust be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current test is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results An example on site testing is configuration oriented system integration test.

White box Testing

It is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or atleast its purpose. It is used to test areasthat cannot be reached from a black box level.

Black Box Testing

Black Box Testing a testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kind of tests must be written from a definitive source document, such as specification requirements document. It is a testing in which the software under test is treated as black box you cannot see into it. The test provides inputs and responds to outputs without considering how the software works.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

7.3. TEST PLAN:

A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning . It is a record of the test planning process. Follow the below steps to create a test plan as per IEEE 829.

7.3.1. Analyze the system:

A system/product can be analyzed only when the tester has any information about it i.e, how the system works, who the end users are, what software/hardware the system uses, what the system is for etc.

7.3.2. Design the Test Strategy:

Designing a test strategy for all different types of functioning, hardware by determining the efforts and costs incurred to achieve the objectives of the system.

For any project, the test strategy can be prepared by

- Defining the scope of the testing.
- Identifying the type of testing required
- Risks and issues
- Creating test logistics.

7.3.4. Define the Test Objectives:

Test objective is the overall goal and achievement of the test execution. Objectives are defined in such a way that the system is bug-free and is ready to use by the end-users. Test objective can be defined by identifying the software features that are needed to test and the goal of the test, these features need to achieve to be noted as successful.

7.3.5. Define Test Criteria:

Test Criteria is a standard or rule on which a test procedure or test judgment can be based. There are two such test criteria: Suspension criteria where if the specific number of test cases are failed, then the tester should suspend all the active test cycle till the criteria is resolved, Exit criteria which specifies the criteria that denote a successful completion of a test phase.

7.3.6. Resource Planning:

Resource plan is a detailed summary of all types of resources required to complete the project task. Resource could be human, equipment and materials needed to complete a project.

7.3.7. Plan Test Environment:

A testing environment is a setup of software and hardware on which the testing team is going to execute test cases.

7.3.8. Schedule & Estimation:

Preparing a schedule for different testing stages and estimating the time and man power needed to test the system is mandatory to mitigate the risk of completing the project within the deadline. It includes creating the test specification, test execution, test report, test delivery.

7.3.9. Determine Test Deliverables:

Deliverables are the documents, tools and other components that has to be developed and maintained in support of the testing effort. Test deliverables are provided before, during and after the testing phase.

Test Plan and different scenarios to be considered for this project are as follows:

1.This project is developed at an aim to detect the drowsiness and the distraction of the driver while driving a non 2-wheeler vehicle.

2.Various modules of the project like the face detection, eye detection, face and eye tracking, drowsiness detection and distraction detection are all tested under unit and functional testing. After detecting the face, check if the system is able to extract required features of the face or not through functional testing. Since this is project doesn't have any form of api, API testing and database testing are out of scope for test plan. Nonfunctional testing such as stress, performance or logical database currently will not be tested. At last, the entire system is tested through system testing for the alert while the driver is drowsy/distracted.

3.To mitigate the risks of any team member not able to understand the testing, every team member in the project group is made familiar to the testing process. To make the testing faster, two members of this project group have been handled the responsibility to test this system. G.Vamsi Krishna and P.R.Krishna Priya have performed the functional and system testing of this system respectively.

4.The suspension criteria for this project is considered as 50% i.e, if whenever the 50% of test cases failed then the testing cycle is suspended and the development team comprising of D. Saiesh and J. Likith have made the required improvement in the code. The exit criteria is that the 95% of the test cases should be successful. We have achieved this success under low light conditions and good light conditions.

5.The resource planning for testing comprised of two members from the project team namely, G. Vamsi Krishna who identified different scenarios for testing amd can be considered as the Developer in test cum Test Administrator, P.R. Krishna Priya who executed the given tests, logged the results and reported the defects to the project team.

6.The system resources and the test environment comprised of the Windows 10 PC run under an i5 6^{th} Gen processor and 8GB RAM with a 1TB HDD. The entire project is run and tested under PyCharm IDE with an attached WebCam.

7.The schedule for this project is as follows:

- The test specification for this project is created by G. Vamsi Krishna which comprised of testing the different modules of the project under different conditions and excluding certain scenarios.
- The test execution is performed by P.R. Krishna Priya which included executing all various test cases provided by G. Vamsi Krishna. Logging the results and reporting the outcomes along with defects is performed by her carefully.
- The test reports are generated accordingly by both the members of the testing team. Defects are dealt accordingly by the development team which included D. Saiesh and J. Likith. Later again these test cases have been tested and are proved successful in achieving the correct results.

7.4. TEST REPORT

The testing of the system is performed for various test cases under different conditions considering most of the possible scenarios. To list a few, some of the test reports have been listed below.

Table 7.4.1 (Unit testing)

Test case:	UTC-1
Name of the Test:	Face Detection
Item Tested:	Face
Sample Input:	Video of the person with face included
Expected output:	Face Detection in the frame
Actual output:	Same as expected output
Remarks:	Successful

Table 7.4.2 (Unit testing)

Test case:	UTC-2
Name of the Test:	Eye Detection
Item Tested:	Eye
Sample Input:	Driver’s face
Expected output:	Eye Detection
Actual output:	Same as expected output
Remarks:	Successful

Table 7.4.3 (Functional Testing)

Test case:	FTC-1
Name of the Test:	Drowsiness Detection
Item Tested:	Drivers eyes
Sample Input:	Face and eyes of the driver
Expected output:	Drowsiness Detection
Actual output:	Same as expected output
Remarks:	Successful

Table 7.4.4 (Integration testing)

Test case:	ITC-1
Name of the Test:	System Alert
Item Tested:	Alarm
Sample Input:	Drivers face in drowsy condition
Expected output:	Alert in form of alarm
Actual output:	Same as expected output
Remarks:	Successful

8. Literature Survey

In computer science, image processing is the use of computer algorithms to perform image processing on images. As a subcategory or field of digital signal processing, image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the buildup of noise and signal distortion during processing. Since images are defined over two dimensions digital image processing may be modelled in the form of multidimensional system.

8.1. DROWSINESS DETECTION THROUGH REGION OF INTEREST:

Region of interest (ROI) can detect a driver’s face. As can be seen in the blue rectangle is the region of interest. The way to create an ROI area is to first obtain the green rectangle area from the Haar Cascade Classifier in the first frame, which includes height, width. Then, the rectangle is scaled up to create region of interest. There are several steps to calculate the ROI area and we have to calculate ROI for each and every region of interest

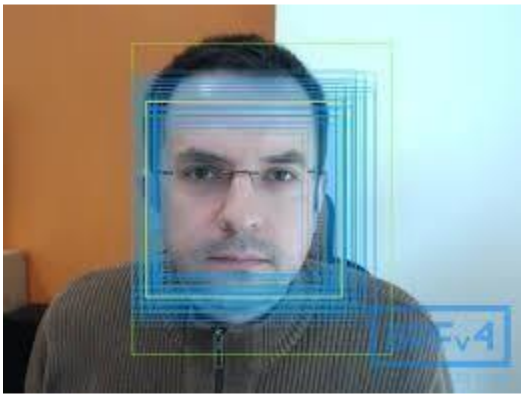


Fig 2.1 Region of Interest

DISADVANTAGES OF REGION OF INTEREST

1. It is uses extra frames or squares to detect face detection.
2. It can't find in low light.
3. Why to use again region of interest while Haar cascade classifier can do the same process?
4. It can't detect while using glasses in driving.

8.2. DETECTION OF DROWSINESS THROUGH LBPH:

In this algorithm the faces are detected by using local binary patterns histograms (LBPH). The first computational step in lbph is to create an intermediate images that describes the original image in a binary format. The image is converted into matrix form and we need to take a central value of the matrix to be used as and threshold value. This value is used to define neighbouring values which can be set to to either 0 or 1. The values which are 1 in the matrix form are to be considered and the remaining values are discarded. The values represent each pixel. Through this the region of face can be detected.

DISADVANTAGES OF LBHP

- It produces less urate results
- The computational time is high.
- This will work only if the data samples are less.

8.3. BEHAVIOURAL BASED TECHNIQUES

The different techniques used in behavioural based parameters are:

8.3.1. EYETRACKINGANDDYNAMIC TEMPLATE MATCHING

To avoid road accidents, real time driver fatigue detection system based on vision is proposed. Firstly, system detects the face of driver from the input images using HSI color model. Secondly, Sobel edge operator is used to locate the eyes positions and gets the images of eye as the dynamic template for the tracking of eye. Then the obtained images are converted to HSI colour model to decide that whether the eyes are close or open to judge the drowsiness of driver. The experiments use four test videos for the tracking of eyes and face detection. The proposed system is compared with the labelled data which is annotated by the experts. The average correct rate of proposed system reaches up to 99.01 % and the precision to 88.90 %.

8.3.2. MOUTH AND YAWNING ANALYSIS

Fatigue is the major reason for road accidents. To avoid the issue, Sarada Devi and Bajaj proposed the driver fatigue detection system based on mouth and yawning analysis. Firstly, the system locates and tracks the mouth of a driver using cascade of classifier training and mouth detection from the input images. Then, the images of mouth and yawning are trained using SVM. In the end, SVM is used to classify the regions of mouth to detects the yawning and alerts the fatigue. For experiment, authors collect some videos and select 20 yawning images and more than 100 normal videos as dataset. The results show that the proposed system gives better results as compared to the system using geometric features. The proposed system detects yawning, alerts the fatigue earlier and facilitates to make the driver safe.

8.3.3. FACIAL EXPRESSIONS METHOD

Laboratory condition using Finite Element Analysis is used by the researchers which is a complex system that contains the database of facial expression as a template and detect the drowsiness on the basis of results from database. Similarly, Assari and Rahmati present the hardware-based Driver Drowsiness Detection system based on facial expressions. The hardware system uses infrared light as it has giving many benefits like ease of use, independent of lightning conditions of environment. The system firstly uses the technique of background subtraction to determines the face region from the input images. Then using horizontal projection and template matching, facial expressions are obtained. After that in the tracking phase, elements found earlier are followed up using template matching and then investigates the incidence of sleepiness using the determination of facial states from the changes of the facial components. Changing in the three main elements such as eye brow rising, yawning and eye closure for the certain period are taken as the initial indications for drowsiness and the system generates the alert. The experiment is performed in the real driving scenario. For testing, images are acquired by the webcam under different conditions of lighting and from different people. The results investigate that the system produces appropriate response in the presence of beard or glasses and mustache on the face of driver.

8.3.4. YAWNING EXTRACTIONMETHOD

Fatigue or drowsiness is the major reason for road accidents. To prevent the issue, Alioua proposed the efficient system for monitoring the driver fatigue using Yawning extraction. Firstly, face region is obtained from the images using Support Vector Machine (SVM) technique to reduce the edge required cost. The proposed method is used to localize the mouth, detection technique is used to detects facial edges, then compute vertical projection on the lower half face to detect the right and left region boundaries and then compute the horizontal projection on the resulting region to detect the upper and lower limit of mouth and mouth localized region is obtained. Finally, to detect the yawning, Circular Hough Transform (CHT) is executed on the images of mouth region to identify the wide-open mouth. If the system finds notable number of continuous frames where the mouth is widely open, system generates the alert. The results are compared with the other edge detectors like Sobel, Prewitt, Roberts, Canny. The experiment uses 6 videos representing real driving conditions and results are presented in the form of confusion matrix. The proposed method achieves 98% accuracy and outperforms all other edge detection techniques.

8.3.5. EYE CLOSURE AND HEAD POSTURES METHOD

Teyeb proposed the Drowsy Driver Detection using Eye Closure and Head postures. Firstly, video is captured using webcam and for each frame of video, following operations are performed. To detect the ROI (face and eyes), viola-jones method is used. The face is partitioned in to three areas and the top one presenting the aye area is browsed by the Haar classifier. Then to detect the eye state, Wavelet Network based on neural network is used to train the images then the coefficients learning images is compared with the coefficients of the testing images and tells which class it belongs. When the closed eye is identified in the frames then the eye closure duration is calculated, if the value exceeds the pre- defined time then the drowsiness state is detected. Then the developed system estimates the head movements which are: left, right, forward, backward inclination and left or right rotation. The captured video is segmented into frames and extract the images of head and determines the coordinates of image. Then the images are compared to determine the inclined state of head and same case with other head postures. Finally, the system combines the eye closure duration and head posture estimation to measure the drowsiness. To evaluate the system, experiment is performed on 10 volunteers in various situations. And results show that the systems achieve the accuracy of 80%.

8.3.6. REAL TIME ANALYSIS USING EYE AND YAWNING

Kumar proposed the real time analysis of Driver Fatigue Detection using behavioral measures and gestures like eye blink, head movement and yawning to identify the drivers' state.

The basic purpose of the proposed method is to detect the close eye and open mouth simultaneously and generates an alarm on positive detection. The system firstly captures the real time video using the camera mounted in front of the driver. Then the frames of captured video are used to detect the face and eyes by applying the viola-jones method, with the training set of face and eyes provided in OpenCV. Small rectangle is drawn around the center of eye and matrix is created that shows that the Region of Interest (ROI) that is eyes used in the next step. Since the both eyes blink at the same time that's why only the right eye is examined to detect the close eye state. If the eye is closed for certain amount of time it will be considered as closed eye. To determine the eye state, firstly the eye ball color is acquired by sampling the RGB components on the center of eye pixel. Then the absolute thresholding is done on the eye ROI based on eye ball color and intensity map is obtained on Y-axis that show the distribution of pixels on y-axis which gives the height of eye ball and compared that value with threshold value which is 4 to distinguish the open and close eye. After that, if the eye blink is detected in each frame it will be considered as 1 and stored in the buffer and after the 100 frames, eye blinking rate is calculated. Then to detect the yawning motion of the mouth, contour finding algorithm is used to measure the size of mouth. If the height is greater than the certain threshold. It means person is taking yawning. To evaluate the performance of the proposed system, system has been measured under different conditions like persons with glasses, without glasses, with moustache and without moustache for 20 days in different timings. The system performs best when the drivers are without glasses.

8.3.7. EYE BLINK DETECTIONMETHOD

Ahmad and Borolie proposed the Driver Drowsiness System based on non-intrusive machine-based concepts. The system consists of a web camera which is placed in front of the driver. Online videos as well as saved videos for simulation purposed are considered. Firstly, camera records the facial expressions and head movements of the driver. Then the video is converted into frames and each frame is processed one by one. Face is detected from frames using Viola-jones algorithm. Then the required features like eyes, mouth and head from face are extracted using cascade classifier. Region of interest on face is indicated by rectangles. Here the main attribute of detecting drowsiness is eyes blinking, varies from 12 to19 per minute normally and indicates the drowsiness if the frequency is less than the normal range. Instead of calculating eye blinking, average drowsiness is calculated. The detected eye is equivalent to zero (closed eye) and non-zero values are indicated as partially or fully open eyes. The equation (2) is used to calculate the average.

%d= no of closed eyes found / no of frames ---eq(2)

8.3.8. EYE CLOSENESS DETECTION METHOD

Khunpisuth creates an experiment the calculates the drowsiness level of driver using Raspberry Pi camera and Raspberry Pi 3 model

B. Firstly Pi camera captures video and to detect face regions in the images, Haar cascade classifier from Viola-Jones method is used. Several images are trained in different lights. The percentage of 83.09 % is achieved based on the case study with 10 volunteers. Blue rectangle shows the Region of Interest (ROI) that is face. Again, Haar cascade classifier is applied on the last obtained frame which reduces the size of ROI. After the face detection, drowsiness level is calculated using eye blink rate. Eye region is detected using template matching on the face and authors uses three templates to check the eye blink and aye area. CV_TM_CCOEFF_NORMED from OpenCV is considered as it gives improved results than other methods of template matching. The integration of eyes and face detection permits the checking of an eye blinking and closeness rate. If the eyes are closed, then the value of closed eye is higher than the open the eyes and opposite case if eyes are open. Authors assumed that Haar cascade classifier will work if the face is front facing position. That why authors proposed the method to rotate the tilted face back in to the front-facing position. Firstly, determines whether the head is tilt or not then calculates the degrees of rotation (angle). After the accurate detection of face and eyes, drowsiness level of driver is determined. If the drivers blink eyes too frequently, he system indicates he drowsiness. When the level reaches to one hundred, a loud sound will be generated to alert the driver.

The proposed method is compared with Haar cascade and results shows that the proposed method achieves the accuracy of 99.59 %. It works in all lighting conditions and able to detect the face wearing glasses.

8.4. VEHICULAR PARAMETER-BASED TECHNIQUES

8.4.1. REAL TIME LANE DETECTION SYSTEM

Road accidents have become common in the present era, causing the severe damage to the property and also to the lives of people travelling. There are many reasons of road accidents like: rash driving, inexperience, ignoring signboards, jumping signal etc. To address the issues, Katyal *et al.* proposed the Drivers' Drowsiness Detection system. The system works in two phases: firstly, detects lane based on Hough transform. Secondly, detects the drivers' eyes to detect the drowsiness. For eye detection, firstly use viola jones method to detect face, then do the image segmentation, after that otsu thresholding is done and canny edge detection is applied. The obtained results is integrated with the circle detection hough transform method to detect eyes to detect the fatigue level. It will also work in low lightning conditions. Result shows that the proposed system is useful for the drivers travelling on lengthy routes, driving late night, drivers who drink and drive.

8.4.2 TIMESERIESANALYSISOFSTEERIN GWHEEL ANGULARVELOCITY

To avoid the road accidents, Zhenhai proposed the Driver Drowsiness Detection method using time series analysis of steering wheel angular velocity. The method firstly analyses the behavior of steering below the fatigue, then temporal detection window is used as the detection feature to determine the angular velocity of steering wheel during time-series. In the temporal window, if the detection feature satisfies the variability constraints and extent constraints, then the state of drowsiness is detected accordingly. The experiment based on real testers is performed, and results shows that the proposed method outperforms the previous methods and useful in the real world.

8.4.3 STEERINGWHEEL ANGLEFOR REAL DRIVING CONDITIONS FOR DDT

To avoid road accidents, Li proposed the online detection of Drowsiness Detection System to monitor the fatigue level of drivers under real conditions using Steering Wheel Angles (SWA). The data of SWA is collected from the sensors attached on the steering lever. The system firstly extracts the features of Approximate Entropy (ApEn) from fixed sliding windows on time series of real time steering wheel angles, then the system linearizes the features of ApEn using the deviation of adaptive linear piecewise fitting method. After that the system calculates the warping distance between the series of linear features of sample data. Finally, system determine the drowsiness state of drivers using warping distance according to the designed decision classifier. The empirical analysis uses the data collected in 14.68 hrs. driving under real road conditions and evaluated on two fatigue levels: drowsy and awake. Results show that the proposed sys- tem is capable for working online with an accuracy of 78.01 % and useful for the prevention of road accidents caused by drivers' fatigue.

8.4.4. AUTOMATIC DETECTION OF DRIVER FATIGUE

To address the issue of drivers' fatigue, an online detection of drivers' fatigue using the Steering Wheel Angles (SWA) and Yaw Angles (YA) information in the real driving conditions is proposed. The system firstly investigates the operation features of SWA and YA in the different states of fatigue, after that calculates the ApEn features on time series of shot sliding window, then using the dynamic time series of non-linear feature construction theory and taking features of fatigue as input, designs a 2-6-6-3 multi-level Back Propagation (BP) neural network classifier to deter- mine the fatigue detection. For empirical analysis, 15 hours long experiment is performed in real road conditions. The experts evaluated the retrieved data and categorized in three levels of fatigue: drowsy, very drowsy, and awake. And the experiment achieves the average accuracy of 88.02% in fatigue detection and valuable for the engineering applications.

8.5 DROWSINESS DETECTION THROUGH PHYSIOLOGICAL APPROACH

Physiological measures have much of the time been utilized for drowsiness discovery as they can give an immediate and objective measure. Conceivable measures are EEG, eyelid closure, movements of eye, heart rate, size of pupil, skin conductance and creation of the cortical. Among these procedures, the systems that are best, in light of precision are the ones in view of physiological experience of human. There are two ways for implementing this procedure. Measurement of changes in physiological signs for example, waves of human brain, blinking of eyes and heart rate; and physical changes measurement for example, drooping stance, leaning of the head of driver and the open/close conditions of the eyes.

8.5.1. EEG-BASED DRIVER FATIGUE DETECTION

EEG is a technique for measuring the electrical action created by the nerve cells of the human brain, basically the cortical movement. The EEG- action is available all the time and recording show both arbitrary and periodic behaviour. The fundamental inception of the EEG is the neuronal action in the cerebral cortex; however some action like wise starts from the thalamus and from subcritical parts of the human brain. The EEG speaks to the summation of excitatory and inhibitory postsynaptic possibilities in the nerve cells. The musical movement is because of the synchronous actuation of the nerve cells. The drivers' fatigue detection system using Electroencephalogram (EEG) signals is proposed to avoid the road accidents usually caused due to drivers' fatigue. The proposed method firstly finds the index related to different drowsiness levels. The system takes EEG signal as input which is calculated by a cheap single electrode neuro signal acquisition device. To evaluate the proposed method, data set for simulated car driver under the different levels of drowsiness is collected locally. And result shows that the proposed system can detect all subjects of tiredness.

Disadvantages:

1. High cost
2. Sensors Required
3. Can’t see behind objects
4. Takes longer time

8.5.2 WAVELET ANALYSIS OF HEARTRATEVARIABILITY & SVM CLASSIFIER

Li and Chung proposed the driver drowsiness detection that uses wavelet analysis of Heart Rate Variability (HRV) and Support Vector Machine (SVM) classifier. The basic purpose is to categorize the alert and drowsy drivers using the wavelet transform of HRV signals over short durations. The system firstly takes Photo PlethysmoGraphy (PPG) signal as input and divide it into 1- minute intervals and then verify two driving events using average percentage of eyelid closure over pupil over time (PERCLOS) measurement over the interval. Secondly, the system performs the feature extraction of HRV time series based on Fast Fourier Trans- form (FFT) and wavelet. A Receiver Operation Curve (ROC) and SVM classifier is used for feature extraction and classification respectively. The analysis of ROC shows that the wavelet-based method gives improved results than the FFT-based method. Finally, the real time requirements for drowsiness detection, FFT and wavelet features are used to train the SVM classifier extracted from the HRV signals. The performance of classification using the wavelet-based features achieve the accuracy of 95%, sensitivity to 95% and specificity to 95%. The FFT-based results achieve the accuracy of 68.85. The results show that wavelet-based methods perform better than the FFT-based methods.

8.5.3 PULSE SENSOR METHOD

Mostly, previous studies focus on the physical conditions of drivers to detect drowsiness. That's why Rahim detects the drowsy drivers using infrared heart-rate sensors or pulse sensors. The pulse sensor measures the heart pulse rate from drivers' finger or hand. The sensor is attached with the finger or hand, detects the amount of blood flowing through the finger. Then amount of the blood's oxygen is shown in the finger, which causes the infrared light to reflect off and to the transmitter. The sensor picks up the fluctuation of oxygen that are connected to the Arduino as microcontroller. Then, the heart pulse rate is visualizing by the software processing of HRV frequency domain. Experimental results show that LF/HF (Low to high frequency) ratio decreases as drivers go from the state of being awake to the drowsy and many road accidents can be avoided if an alert is sent ontime.

8.5.4 WEARABLE DRIVER DROWSINESS DETECTION SYSTEM

Mobile based applications have been developed to detect the drowsiness of drivers. But mobile phones distract the drivers' attention and may cause accident. To address the issue, Leng proposed the wearable-type drowsiness detection system. The system uses self-designed wrist band consists of PPG signal and galvanic skin response sensor. The data collected from the sensors are delivered to the mobile device which acts as the main evaluating unit. The collected data are examined with the motion sensors that are built-in in the mobiles. Then five features are extracted from the data: heart rate, respiratory rate, stress level, pulse rate variability, and adjustment counter. The features are moreover used as the computation parameters to the SVM classifier to determine the drowsiness state. The experimental results show that the accuracy of the proposed system reaches up to 98.02 %. Mobile phone generates graphical and vibrational alarm to alert the driver.

8.5.5 WIRELESS WEARABLES METHOD

To avoid the disastrous road accidents, Warwick proposed the idea for drowsiness detection system using wearable Bio sensor called Bio- harness. The system has two phases. In the first phase, the physiological data of driver is collected using bio-harness and then analyzes the data to find the key parameters like ECG, heart rate, posture and others related to the drowsiness. In the second phase, drowsiness detection algorithm will be designed and develop a mobile app to alert the drowsy drivers.

8.5.6 DRIVER FATIGUE DETECTION SYSTEM

Chellappa presents the Driver fatigue detection system. The basic of the system is to detect the drowsiness when the vehicle is in the motion. The system has three components: external hardware (sensors and camera), data processing module and alert unit. Hardware unit communicates over the USB port with the rest of the system. Physio- logical and physical factors like pulse rate, yawning, closed eyes, blink duration and others are continuously monitored using somatic sensor. The processing module uses the combi- nation of the factors to detect drowsiness. In the end, alert unit alerts the driver at multiple stages according to the severity of the symptoms.

8.5.7 HYBRID APPROACH UTILIZING PHYSIOLOGICAL FEATURES

To improve the performance of detection, Awais proposed the hybrid method which integrates the features of ECG and EEG. The method firstly extracts the time and frequency domain features like time domain statistical descriptors, complexity measures and power spectral measures from EEG. Then using ECG, features like heart rate, HRV, low frequency, high frequency and LF/HF ratio.

After that, subjective sleepiness is measured to study its relationship with drowsiness. To select only statistically significant features, t-tests is used that can differentiate between the drowsy and alert. The features extracted from ECG and EEG are integrated to study the improvements in the performance using SVM. The other main contribution is to study the channel reduction and its impact on the performance of detection. The method measures the differences between the drowsy and alert state from physiological data collected from the driving simulated-based study. Monotonous driving environment is used to induce the drowsiness in the participants. The pro- posed method demonstrated that combining ECG and EEG improves the performance of system in differentiating the drowsy and alert states, instead of using them alone. The analysis of channel reduction confirms that the accuracy level reaches to 80% by just combining the ECG and EEG. The performance of the system indicates that the proposed system is feasible for practical drowsiness detection system.

8.6. **Other Methodologies**:

Every year a large number of deaths occur due to fatigue related road accidents. According to study around 20% accidents are occuring yearly with an average of 90 deaths per day due to drowsiness. Drivers who drive continuously will have a chance of getting tiredness. Hence detection of drivers drowsiness and its indication can significantly decrease number of accidents. To decrease these type of accidents some image processing techniques like viola jones, Adaboost, haar cascade, gobar features, facial land mark detection . The following are some of the methodologies for detecting the drowsiness

M.A. Assari & M. Rahmati [1] proposed a system in which the drowsiness of the driver is detected by detecting the face through horizontal projection on the image and tracking the face components via template matching technique which comprised of eyebrows and eyes along with mouth. The proposed method has been implemented in simulation environment of MATLAB (Simulink). Addition of the IR lighting as sources of light helped in better detection of faces in this system.

Tianyi Hong[2] presented a system which used face-detection method basing on the cascade of classifiers trained through Adaboost technique. Optimization in this system is performed by applying the integral image of the original image to develop a canny filter for cascade processing and improve the performance. Integrated performance primitives(IPP) have been used for better and faster computational results. This system is validated in GENE-8310 embedded platform.

B. Warwick[3] proposed a system that is based on physiological approach in which the driver wears a wireless biosensor called BioHarness, a wearable device capable of collecting the physiological data and then transmitting to a smartphone. This data is then analysed through Fast Fourier Transform (FFT) and Power Spectral Density (PSD) which provide the desired vectored inputs that can be fed into a Neural Network. This system is run on a drowsiness detection mobile app by the researchers.

K. Dwivedi[4] developed a system which identifies drowsiness of the driver using representational learning. A Haar-like face detector feeds the images to a 2-layer convolutional neural network for extracted features which are then used to train a softmax layer classifier for detecting whether a driver is drowsy or not drowsy. This system was able to yield a satisfactory result of 78% accuracy in detecting the drowsiness and alerting the driver.

J.J. Yan[5] developed a system in which the images captured are converted into grayscale using the Sobel operator for edge detection. The position of the eyes are calculated using template matching. To determine the states of the eyes, the binarization and quick sort techniques are used which also confirm the distribution of the black pixels in the grayscale image.

In this study, P80 is taken as the important criterion of the driver's physical state. If the amount of black pixels is lower than this threshold value then it is considered as the driver in drowsy state.

9. CONCLUSION AND FUTURE WORK

9.1 CONCLUSION:

The current study developed an automated system for detecting drowsiness of the driver. The continuous video stream is read from the system and is used for detecting the drowsiness. It is detected by using haar cascade algorithm. The haar cascade algorithm uses haar features to detect face and eyes. Haar features are predefined are used for detecting different things. The haar features are applied on the image and blink frequency is calculated using perclos algorithm. If the value remains 0 for some amount of time then it detects as sleepy and alerts driver by activating an alarm. If the value remains constant for longer periods then the driver is said to be distracted then also an alarm is activated.

9.2 FUTURE WORK:

The work can be extended by extracting the features of mouth where the driver can be detected as drowsy through yawning. If the driver yawns repeatedly for more number of times then we can say that he is in sleepy mode. If the number exceeds a limit then we can alert the driver. This work can also be extended by implementing in full night light using IR web cam. It is camera which uses infrared radiations to detect whether the person is drowsy or not.

While this is a research project, there is scope when this completely turns out to be developed into an application which can be run by the end users on their own for their own purposes on their own systems.

10 Bibliography

For Book References:

1. Joel Murach and Michael Urban, "Murach's Java Servlets and JSP", 3rd Edition, Published June 2014.

2. K. Santosh Kumar, "JDBC, Servlets, and JSP Black Book", New Edition, Publisher: Dreamtech Press India Pvt. Ltd, Published: May 13 2008.

For Website References

Java-web, Servlet and JSP Tutorial

edu4Java JDBC Tutorial - Tutorialspoint

11 APPENDIX

11.1 Introduction to Python

Python is an open-source, high-level programming language developed by Guido van Rossum in the late 1980s and presently administered by Python Software Foundation. It came from the ABC language that he helped create early on in his career. Python is a powerful language that you can use to create games, write GUIs, and develop web applications.

It is a high-level language. Reading and writing codes in Python is much like reading and writing regular English statements. Because they are not written in the machine-readable language, Python programs need to be processed before machines can run them. Python is an interpreted language.

Readability: Python programs use clear, simple, and concise instructions that are easy to read even by those who have no substantial programming background. Programs written in Python are, therefore, easier to maintain, debug, or enhance.

Higher productivity: Codes used in Python are considerably shorter, simpler, and less verbose than other high-level programming languages such as Java and C++. In addition, it has well-designed built-in features and a standard library as well as access to third party modules and source libraries.

Less learning time: Python is relatively easy to learn. Many find Python a good first language for learning programming because it uses simple syntax and shorter codes. Python works on Windows, Linux/UNIX, Mac OS X, other operating systems and small form devices. It also runs on microcontrollers used in appliances, toys, remote controls, embedded devices, and other similar devices.

Installing Python in Windows: To install Python, you must first download the installation package from this link: https://www.python.org/downloads/ On this page, you will be asked to choose between the two latest versions for Python 2 and 3: Python 3.5.1 and Python 2.7.11. . You would normally opt to download the latest version, which is Python3.5.1. This was released on December 7, 2015. However, you may opt for the latest version of Python 2, 2.7.11. Your preferences will usually depend on which version will be most useful for your project. While Python 3 is the present and future of the language, issues such as third-party utility or compatibility may require you to do2wnload Python 2.

11.2 Introduction to OPENCV

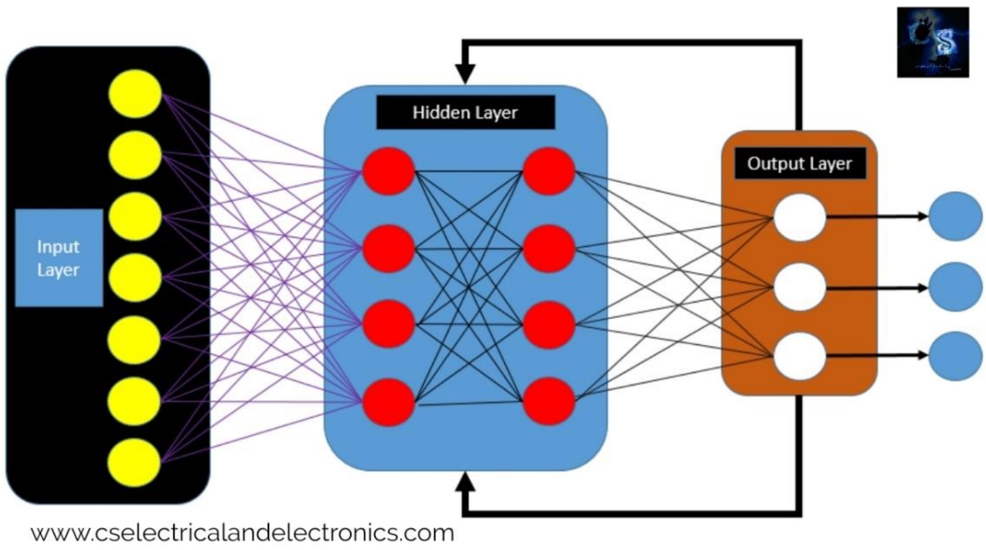
OpenCV (Open-Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality.

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by willow garage then Itseez (which was later acquired by Intel). The library is cross platform and free for use under the open source BSD license. OpenCV supports some models from deep learning frameworks 54 | P a g e like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers. It promotes Open Vision Capsules which is a portable format, compatible with all other formats.

11.3 Introduction to CNN

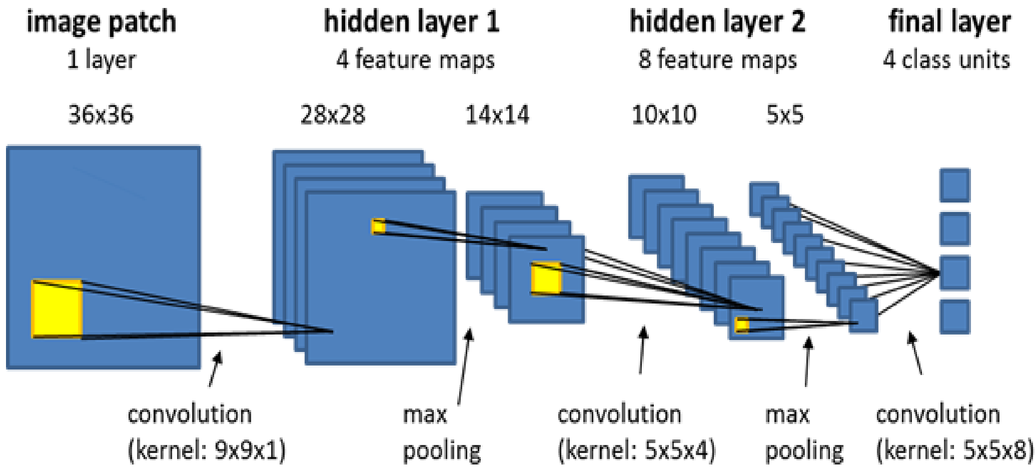
A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

Next, we will apply a Pooling layer to our Convolutional layer, so that from every feature map we create a Pooled feature map as the main purpose of the pooling layer is to make sure that we have spatial invariance in our images. It also helps to reduce the size of our images as well as avoid any kind of overfitting of our data. After that, we will flatten all of our pooled images into one long vector or column of all of these values, followed by inputting these values into our artificial neural network. Lastly, we will feed it into the locally connected layer
To achieve the final output.



1. **Convolution Layer:**

In convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration we slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process we will create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour.

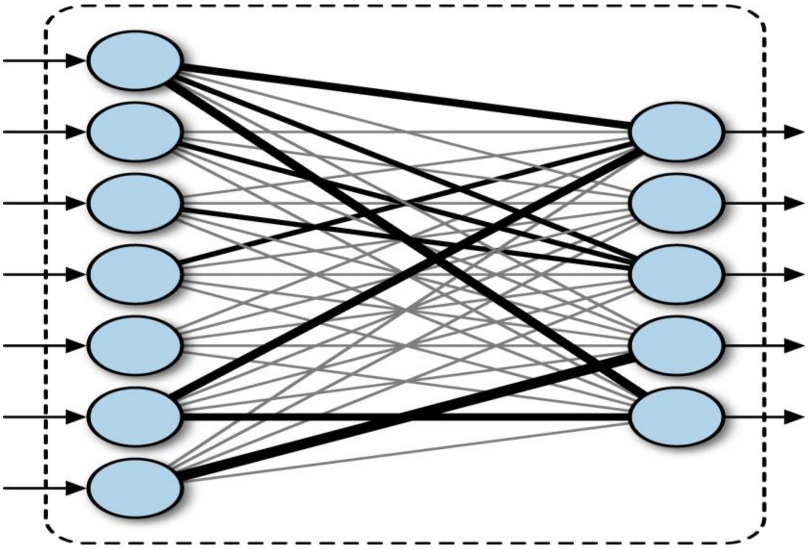


2. Pooling Layer:

We use pooling layer to decrease the size of activation matrix and ultimately reduce the learnable parameters. There are two types of pooling: a. Max Pooling: In max pooling we take a window size [for example window of size 2*2], and only take the maximum of 4 values. Well lid this window and continue this process, so well finally get an activation matrix half of its original Size. b. Average Pooling: In average pooling, we take advantage of of all Values in a window.

3. Fully Connected Layer:

In convolution layer, neurons are connected only to a local region, while in a fully connected region, we will connect all the inputs to neurons.



4.Final Output Layer:

After getting values from fully connected layer, we will connect them to the final layer of neurons [having count equal to total number of classes], that will predict the probability of each image to be in different classes.