**CNN** **DEEP** **LEARNING** **ASSISTANCE** **FOR** **DRIVER** **ATTENTION** **MONTIORY**

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in Computer Science and Engineering

By

**OM** **PRAKASH** **BISWAL** **(** **Reg.No** **-** **39110713** **)** **DEVA** **HARSHA** **NUTHAKKI** **(** **Reg.No** **–** **39110712** **)**

**DEPARTMENT** **OF** **COMPUTER** **SCIENCE** **AND** **ENGINEERING**

**SCHOOL** **OF** **COMPUTING**

**SATHYABAMA** **INSTITUTE** **OF** **SCIENCE** **AND** **TECHNOLOGY**

**(DEEMED** **TO** **BE** **UNIVERSITY)** **Accredited** **with** **Grade** **“A”** **by** **NAAC**

**JEPPIAAR** **NAGAR,** **RAJIV** **GANDHISALAI,** **CHENNAI** **-** **600119**

**NOVEMBER** **-** **2022**

**SATHYABAMA** INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED** **TO** **BE** **UNIVERSITY)** Accredited with ―A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119 [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in/)

**DEPARTMENT** **OF** **COMPUTER** **SCIENCE** **AND** **ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Omprakash** **Biswal(39110713)** **and** **Deva** **Harsha(39110712)** who carried out the Project Phase-1 entitled **“CNN** **DEEP** **LEANING** **ASSISTANCE** **FOR** **DRIVER** **ATTENTION** **MONITORY”** under my supervision from June 2022 to November 2022.

**Internal** **Guide**

**Ms.SREEKRISHNA** **M.E.,** **Ph.D**

**Head** **of** **the** **Department**

**Dr.** **L.** **LAKSHMANAN,** **M.E.,** **Ph.D.**

**Submitted** **for** **Viva** **voce** **Examination** **held** **on**

**InternalExaminer** **ExternalExaminer**

**DECLARATION**

I, **OM** **PRAKASH** **BISWAL(Reg.No-** **39110713),** hereby declare that the Project Phase-1 Report entitled **CNN** **DEEP** **LEARNING** **ASSISTANCE** **FOR** **DRIVER** **ATTENTION** **MONITORY”** done by me under the guidance of **Ms.SREEKRISHNA,** **M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer** **Science** **and** **Engineering**.

**DATE:**

**PLACE:Chennai**

OM PRAKASH BISWAL

**SIGNATURE** **OF** **THECANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board** **of** **Management** of

**SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.** **T.Sasikala** **M.E.,** **Ph.** **D**, **Dean**, School of Computing, **Dr.** **L.** **Lakshmanan** **M.E.,** **Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide

**MS.SREEKRISHNA** **M.E.,Ph.D,**for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks toall Teaching and Non-teaching staff members of the

**Department** **of** **Computer** **Science** **and** **Engineering** who were helpful in many ways

for the completion of the project.

**ABSTRACT**

The main idea behind this project is to develop a system which can detect fatigue of

anyhuman and can issue a timely warning. Drivers who do not take regular breaks during longdistances run a high risk of becoming drowsy, a state which they often fail to recognizeearly enough. Driving in drowsy state is dangerous because sleep deprivation can havesimilar effects on your body as drinking alcohol. This system will monitor the driver eyesusing a camera. By developing an algorithm we can detect symptoms of driver fatigue earlyenough to avoid the person from sleeping. So, this project will be helpful in detectingdriverfatigue in advance and will give warningoutputin formof alarm. Moreover, the warning willbe deactivated manually rather than automatically. For this purpose, a deactivation dialogwill be generated which will contain some simple mathematical operation which whenanswered correctly will dismiss the warning. Moreover, if driver feels drowsy there ispossibility of incorrect response to the dialog Input variables show a possibility of fatigue atone moment, then a warning signal is given in form of text and sound. This will directly givean indication of drowsiness/fatigue.

i

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapter** **No** | **TITLE** | | | **Page** **No.** |
|  | **ABSTRACT** | | |  |
|  | **LIST** **OF** **FIGURES** | | |  |
| 1 | **INTRODUCTION** | | | 1 |
| 2 | **LITERATURE** **SURVEY** | | | 2 |
| 3 | **REQUIREMENT** **ANALYSIS** | | | 4 |
|  | 3.1 | Purpose | | 4 |
|  | 3.2 | Requirement Analysis | | 5 |
|  | 3.3 | System Requirement | | 6 |
|  | 3.4 | Software Description | | 6 |
| 4 | **WORKING** **OF** **CNN** **ALGORITHM** | | | 9 |
| 5 | **SYSTEM** **ANALYSIS** | | | 11 |
|  | 5.1 | Existing System | | 11 |
|  | 5.2 | Proposed System | | 11 |
|  | 5.3 | Feasibility Study | | 14 |
|  | 5.4 | Effort,Duration,and Cost Estimating using COCOMO Model | | 15 |
| 6 | **SYSTEM** **DESIGN** | | | 18 |
|  | 6.1 | System Architecture | | 18 |
|  | 6.2 | Modules Eye Detection | | 19 |
| 7 | **SYSTEM** **TESTING** | | | 25 |
|  | 7.1 | | Testing | 25 |
|  | 7.2 | | White Box Testing | 25 |
|  | 7.3 | | Black Box Testing | 27 |
| 8 | **CONCLUSION** | | | 30 |
|  | **REFERENCES** | | | 31 |

ii

**LIST** **OF** **FIGURES**

**FIGURE** **FIGURE** **NAME**

5.1 Drownisness Detection

6.1 System Architecture of Drownisness

6.2 Shape predictor 68 face landmarks

6.3 Eye Aspected Ratio

6.4 Use Case Diagram

6.5 Activity Diagram of prediction algorithm

**PAGE** **NO.**

13

19

20

21

22

24

iii

iv

**CHAPTER** **1**

**INTRODUCTION**

Real Time Drowsiness behaviors which are related to fatigue are in the form of eye

closing, headnoddingetc. Hence,we caneither measurechange inphysiological signals, such as brain waves, heart rate and eye blinking to monitor drowsiness or consider physical changes suchas saggingposture, leaningofdriver’s headand open/closedstate of eyes. The former technique, while more accurate, is not realistic since highly sensitive electrodes would have to be attached directly on the driver’s body and hence which can be annoying and distracting to the driver. In addition, long time working would result in perspiration on the sensors, diminishing their ability to monitor accurately. The second technique is to measure physical changes (i.e. open/closed eyes to detect fatigue) is well suited for real world conditions since it is nonintrusive by using a video camera to detect changes. In addition, micro sleeps that are short period of sleeps lasting 2 to 3 minutes are good indicators of fatigue. Thus, by continuously the eyes of the driver one can detect the sleepy state of driver and a timely warning is iissued.

**Image** **Capture**:

Utilizing a web camera introduced inside the automobile we can get the picture of the

driver. Despite the fact that the camera creates a video clip, we have to apply the developed algorithm on each edge of the video stream. This paper is only focused on the applying the proposed mechanism only on single frame. The used camera is a low cost web camera with a frame rate of 30 fps in VGA mode.

1

**CHAPTER** **2**

**LITERATURE** **SURVEY**

In this section, we have discussed various methodologies that have been proposed for drowsiness detection and blink detection during the recent years. Ashish Kumar in 2018 has proposed based on visual behavior and machine learning. Here, visual behavior features like eye aspect ratio, mouth opening ratio and nose length ratio are computed from the streaming video, captured by a webcam. An adaptive thresholding technique has beendevelopedtodetect driver drowsiness in real time. Thedevelopedsystemworks accurately with the generated synthetic data. Subsequently, the feature values are stored and machine learning algorithms have been used for classification. Also, the system will be implemented in hardware to make it portable for car system and pilot study on drivers will be carried out to validate the developed system. Luigi Celona, Lorenzo Mammana, SimoneBianco, RaimondoSchettini in 2018, has proposedaMulti-Task Driver Monitoring Framework (MT-DMF), which is able to simultaneously estimate the status of the driver’s eyes, mouth, head and drowsiness. The framework involves the use of a specifically designed Multi-task CNN. Experimental results show that the proposed framework outperforms not only other methods in the state-of-the-art, but also human-based visual assessment. M. Tayab Khan, H. Anwar, F. Ullah, A. Ur Rehman, R. Ullah, A. Iqbal, B.-H. Lee and K. S. Kwak in 2019, A method for image-based drowsiness detection in real time driving surveillance videos is proposed. It is a four step method that first detects the face of the driver in the image from among several detected faces. Secondly, it extracts the eyes from the detected faces. In the third step, the curvature of the eyelids is detected using a modified Sobel operator. Finally, the eyes are classified as closed or open based on the curvature of the eyelids. Te proposed method achieved an average classification accuracyof 95% on a simple image dataset with homogeneous backgrounds, an average 7 classification accuracy of 70% on a complex benchmark image dataset, and greater than 95% classification accuracy on two real-time driving surveillance videos. However, the proposed method works only in the day time; its adaptation to night time will be explored in future work with more statof-the-art face and eye detection algorithms. Similarly, more challenging face images where subjects might have glasses or phones

2

will be used to evaluate the proposed method. Rateb Jabbar, Khalifa Al-Khalifa, Mohamed Kharbeche, Wael Alhajyaseen, MohsenJafari , Shan Jianga in 2018, drowsiness detection system based on multilayers perceptron classifiers. It is specifically designed for embedded systems such as Android mobile. The role of the system is to detect facial landmark from images and deliver the obtained data to the trained model to identify the driver's state. The purpose of the method is to reduce the model's size considering that current applications cannot be used in embedded systems due to their limited calculation and storage capacity. According to the experimental results, the size of the used model is small while having the accuracy rate of 81%. Hence, it can be integrated into advanced driver-assistance systems, the Driver drowsiness detection system, and mobile applications. However, there is still space for the performance improvement. The further work will focus on detecting the distraction and yawning of the driver.

3

**CHAPTER** **3** **REQUIREMENT** **ANALYSIS**

Software Engineering by James F Peters &WitoldPedrycz. Head First Java by Ka A Software Requirements Specification (SRS) is a description of particular software product, program or set of programs that performs a set of functions in a target environment (IEEE Std. 830- 1993).

**3.1.1** **Purpose**

The purpose of software requirements specification specifies the intentions and intended

audience of the SRS.

**3.1.2** **Scope**

The scope of the SRS identifies the software product to be produced,

thecapabilities,application,relevant objects etc. We are proposed to implement Passive AggressiveAlgorithm which takes the test andtrained data set from the

**3.1.3** **Definitions,** **Acronyms** **and** **Abbreviations** **Software** **Requirements**

**Specification:**

It’s a description of a particular software product, program or set of programs that performs a set of function in target environment.

**3.1.4** **References:**

IEEE Std. 830-1993, IEEE Recommended Practice for Software RequirementsSpecifications the Sierra and Bert Bates.

**3.1.5** **Overview:**

The SRS contains the details of process, DFD’s, functions of the product, user

characteristics. The non-functional requirements if any are also specified **3.1.6** **Overall** **Description:**

The main functions associated with the product are described in this section of SRS. The

characteristics of a user of this product are indicated. The assumptions in this section result from interaction with the project stakeholders.

4

**3.2** **Requirement** **Analysis** **:**

We’ll need the SciPy package so we can compute the Euclidean distance between facial landmarks points in the eye aspect ratio calculation (not strictly a requirement, but you should have SciPy installed if you intend on doing any work in the computer vision, image processing, or machine learning space). We’ll also need the imutils package, my series of computer vision and image processing functions to make working with OpenCV easier. If you don’t already have imutils installed on your system, you can install/upgrade imutils via: $ pip install --upgrade imutils We’ll also import the Thread class so we can play our alarm in a separate thread from the main thread to ensure our script doesn’t pause execution while the alarm sounds .In order to actually play our WAV/MP3 alarm, we need the playsound library, a pure Python, crossplatform implementation for playing simple sounds. The playsound library is conveniently installable via pip : $ pip install playsound

**Dlib** **:-**

The dlib library only has four primary prerequisites: ... Boost. Python: As the name of this

library suggests, Boost. Python provides interoperability between the C++ and Python programming language

**Cmake:**

CMake is an open-source, cross-platform set of tools used to build, test, and packagesoftware **Install** **process:** Step 1: Create Conda Environment Open Anaconda command prompt from Windows start menu. First of all update anaconda packages by issuing the following commands. Step 2: conda update anaconda Now, we will create a new conda environment. conda create -n env\_dlib python=3.6 Step 3: Activate new environment We need to activate the newly created conda environment by using the following command: conda activate env\_dlib After issuing the above command, we will be migrated from conda ‘base’ to ‘env\_dlib’. Step 4: Install Dlib Just type in the following command to download and install the latest version of dlib from conda-forge repositories. conda install -c conda-forge dlib This will download all the required dependencies on its own. Just press ‘Y’ to proceed. It's a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points

5

on a person's face like image below. These points are identified from the pre-trained model where the iBUG300-W dataset was used.

**3.3** **System** **Requirements** **3.3.1** **H/W** **Requirements**

Processor: Any Processor above 500 MHz. Ram: 4 GB Hard Disk: 4 GB Input device: Standard Keyboard and Mouse. Output device: VGA and High Resolution Monitor.

**3.3.2** **S/W** **Requirements**

Operating System : Windows 7 or higher Programming : Python 3.6 and related libraries Software : Anaconda Navigator and Jupyter Notebook.

**3.4** **SOFTWARE** **DESCRIPTION:**

**3.4.1** **PYTHON**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophythat emphasizes codereadability, notablyusingsignificant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms ,including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems Cpython the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

**3.4.2** **Python** **Pandas:**

Pandas is an open-source Python Library providing high-performance data manipulation andanalysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and

6

preparation.It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data ,regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas used in a wide range of fields including academic and commercial domains including finance,economics,Statistics,analy-tics, etc.

**Key** **Features** **of** **Pandas:**

1. Fast and efficient Data Frame object with default and customized indexing. 2. Tools for loading data into in-memory data objects from different file formats. 3. Data alignment and integrated handling of missing data. 4. Reshaping and pivoting of date sets. 5. Label-based slicing, indexing and sub setting of large data sets. 6. Columns from a data structure can be deleted or inserted. 7. Group by data for aggregation and transformations. 8. High performance merging and joining of data. 9. Time Series functionality.

**3.4.3** **Numpy:**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. Sophisticated(broadcasting) functions • Tools for integrating C/C++ and Fortran code • Useful linear algebra, Fourier transform, and random number capabilities 24 • Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases. • It is the fundamental package for scientific computing with Python. It contains various features including these important ones • A powerful N-dimensional array object.

**3.4.4** **DLIB:-**

The dlib library only has four primary prerequisites: Boost. Python: As the name of this library suggests, Boost. Python provides interoperability between the C++ and Python programming language.

**3.4.5** **OpenCV:**

OpenCV stands for Open Source Computer Vision. It's an Open Source BSDlicensed librarythat includes hundredsof advanced Computer Vision algorithms thatare optimized to use hardware acceleration. OpenCV is commonly used for machine learning, image

7

processing, image manipulation, and much more. OpenCV has a modular structure. There are shared and static libraries and a CV Name space. In short, OpenCV is used in our application to easily load bitmap files that contain landscaping pictures and perform a blend operation between two pictures so that onepicture can be seen in the background of another picture This image 13 manipulation is easily performed in a few lines of code using OpenCV versus other methods. OpenCV.org is a must if you want to explore and dive deeper into image processing and machine learning in general

8

**CHAPTER** **4**

**WORKING** **OF** **CNN** **ALGORITHM**

**Step** **1** **–** **Take** **Image** **as** **Input** **from** **a** **Camera**

**Step** **2** **–** **Detect** **Face** **in** **the** **Image** **and** **Create** **a** **Region** **of** **Interest** **(ROI)**

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don’t need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier **face** **=** **cv2.CascadeClassifier(‘** **path** **to** **our** **haar** **cascade** **xml** **file’)**.

Then we perform the detection using **faces** **=** **face.detectMultiScale(gray)**. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face. **for** (x,y,w,h) **in** faces: cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1 )

**Step** **3** **–** **Detect** **the** **eyes** **from** **ROI** **and** **feed** **it** **to** **the** **classifier**

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in **leye** and **reye** respectively then detect the eyes using **left\_eye** **=** **leye.detectMultiScale(gray)**. Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code. l\_eye = frame[ y : y+h, x : x+w ]**l\_eye** only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into **r\_eye**. **Step** **4** **–** **Classifier** **will** **Categorize** **whether** **Eyes** **are** **Open** **or** **Closed**

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using **r\_eye** **=** **cv2.cvtColor(r\_eye,** **cv2.COLOR\_BGR2GRAY)**. Then, we resize the image to 24\*24 pixels as our model was trained on 24\*24 pixel images **cv2.resize(r\_eye,** **(24,24))**. We normalize our data for better convergence **r\_eye** **=** **r\_eye/255** (All values will be between 0-1). Expand the dimensions tofeed into our classifier. Weloaded our model using **model** **=** **load\_model(‘models/cnnCat2.h5’)** . Now we predict each eye with our model**lpred** **=**

9

**model.predict\_classes(l\_eye)**. If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

**Step** **5** **–** **Calculate** **Score** **to** **Check** **whether** **Person** **is** **Drowsy**

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, wedecreasethescore. Wearedrawingtheresultonthescreenusingcv2.putText() function which will display real time status of the person. cv2.putText(frame, “Open”, (10, height-20), font, 1, (255,255,255), 1, cv2.LINE\_AA ) A threshold is defined for example if score becomes greater than 15 that means the person’s eyes are closed for a long period of time. This is when we beep the alarm using **sound.play()IMPLEMENTATION:** Here, you can find how each of the elements used for this article was obtained. • The system was implemented on python 3.5

• The extraction of frames from the webcam was achieved using OpenCV • The facial landmarks were extracted with the library dlib

• The model was constructed using Keras

• The front –end was deployed with the help of flask

**Steps** **for** **Implementation:**

1. Install the required packages for building the Drowsiness detection. 2. Load the libraries into the workspace from the packages.

3. Read the input data set.

10

**CHAPTER** **5** **SYSTEM** **ANALYSIS**

**5.1.** **Existing** **System:**

In the existingsystemwe are usingeye blinkingsensor. Aspectaclewith eye blink sensor is used to detect the driver drowsiness and alerts the driver with buzzer, if driver is affected by drowsiness. If the driver becomes drowsy the eye blink sensor’s frame vibrates attached to the vehicle and also the LCD displays the warning messages and it alerts the driver’s through alarm sound to avoid the road accidents.

**5.2.Proposed** **system:**

In proposed system the eye blinking frequency, head pose, etc. of a person is monitored through a camera and the person is alerted if any of these drowsiness symptoms are detected. This is a behavioral based approach and we use CNN (Convolutional Neural Network) for obtaining reliability and accuracy. Proposed algorithm is based on computer vision method. The main focus is on the detection of blinks by estimating the EAR(Eye aspect Ratio). This is achieved by monitoring the eyes of the driver throughout the entire video sequence. An IR camera will be used for capturing live video of driver eyes in all light conditions and frames will extracted for image processing scheme of video capturing by using pre trained Neural Network based Dlib functions. Dlib functions are trained by CNN algorithm.Drowsiness detection techniques are generally classified into three groups:

• Methods based on the condition of the driver,

• The method based on the performance of the driver, and • The combined of the previous method.

The method based on the condition of the driver is divided into two categories, namely: • A technique using physiological signals and image-based techniques.

• To develop a system which deals with driver drowsiness and fatigue detection based on visual information. The system uses driver’s face movements and eye locations to determine the state of driver’s eyes and if drowsy. This system will be able to work under low-lighting conditions with the help of a webcam installed on the dashboard. Instead of showing a warning sign, our system will sound a loud alarm which can not only alert the driver but the rest of the co-passengers as well. The former technique, while more

11

accurate, is not realistic since highly sensitiveelectrodes would have to be attached directlyon the driver’ bodyand hence which can be annoying and distracting to the driver. In addition, longtime working would result in perspiration on the sensors, diminishingtheir abilityto monitor accurately. So,here we use eye aspect ratio method inordertomeasure physical changes (i.e. open/closed eyes) by using a video camera to detect drowsiness. If the eye aspect ratio method indicates that the eyes have been closed for a sufficiently long enough of time i.e., more than 0.3 seconds, it will sound an alarm to wake the driver, thereby reducing the vehicle accidents. The process can be done by using CNN algorithm. Convolutional Neural Networks, like neural networks, are made up of neurons with learnable weights and biases. Eachneuron receives several inputs, takes a weighted sum over them,pass itthrough an activationfunction and respondswith an output. Neural networks, as its name suggests, is a machine learning technique which is modeled after the brain structure. It comprises of a network of learning units called neurons.

**Figure:5.1** Drowsiness Detection Architecture

**Convolution** **Of** **An** **Image:** Convolution has the nice property of being translational invariant. Intuitively, this means that each convolution filter represents a feature of interest and the Convolutional neural network algorithm learns which features comprise the resulting reference. We have 4 steps for convolution:**Relu** **layer:** Rectified Linear Unit

12

(Relu) transform function only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero, but when the input rises above a certain threshold, it has a linear relationship with the dependent variable

**Pooling:**

In this layer we shrink the image stack into a smaller size. Pooling is done after passing through the activation layer.

**Normalization:**

Manytypes of normalization layers have beenproposed for use in ConvNet architectures, sometimes with the intentions of implementing inhibition schemes observed in the biological brain. However, these layers have since fallen out of favor because in practice their contribution has been shown to be minimal, if any.

**Fully-connected** **layer:**

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset. See the neural network section of the notes for more information. In our project, CNN algorithm is used to retrieve the information which is available in the Dat file and acts as input to the Dlib function. Dat file consists of number of eye templates.When the driver’s eye matches the template it alerts the driver by sounding an alarm. CNN algorithm is a good conduction in identifying the images.def test(image\_path, model\_path): '''Test the given model by showing the detected landmarks. - Image\_path: the path of an image. Should contain a face. -model\_path: the path of a shape predictor model.''' image = cv2.imread(image\_path) face\_detector = dlib.get\_frontal\_face\_detector() dets=face\_detector(image,1) predictor = dlib.shape\_predictor(model\_path) for d in dets: cv2.rectangle(image, (d.left(), d.top()), (d.right(), d.bottom()), 255, 1) shape = predictor(image, d) for i in range(shape.num\_parts): p = shape.part(i) cv2.circle(image, (p.x, p.y), 2, 255, 1) cv2.putText(image, str(i), (p.x + 4, p.y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.25, (255, 255, 255) cv2.imshow("window", image) cv2.waitKey(0) cv2.destroyAllWindows()

**5.3** **FEASIBILITY** **STUDY:**

13

A Feasibility Study is a preliminary study undertaken before the real work of a project starts to ascertain the likely hood of the projects success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. **1.Economic** **Feasibility:**

It is defined as the process of assessing the benefits and costs associated with the development of project. A proposed system, which is both operationally and technically feasible, must be a good investment for the organization. With the proposed system the users are greatly benefited as the users can be able to detect the fake news from the real news and are aware of most real and most fake news published in the recent years. This proposed system does not need any additional software and high system configuration. Hence the proposed system is economically feasible

.**2.** **Technical** **Feasibility:**

The technical feasibility infers whether the proposed system can be developed considering the technical issues like availability of the necessary technology, technical capacity, adequate response and extensibility. The project is decided to build using Python. Jupyter Note Book is designed for use in distributed environment of the internet and for the professional programmer it is easy to learn and use effectively. As the developing organization has all the resources available to build the system therefore the proposed system is technically feasibility.

**3.Operational** **Feasibility:**

Feasibility is defined as the process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities. The system is self-explanatory and doesn’t need any extra sophisticated training. The system has Built in methods and classes which are required to produce the result. The application can be handled very easily with a novice user. The overall time that a user needs to get trained is less than one hour. As the software that is used for developing this application is very economical and is readily available in the market. Therefore the proposed system is operationally feasible.

14

**5.4.** **Effort,** **Duration,** **and** **Cost** **Estimation** **using** **COCOMO** **Model:**

The COCOMO (Constructive Cost Model) model is the most complete and thoroughly documented model used in effort estimation. The model provides detailed formulas for determining the development time schedule, overall development effort, and effort breakdown by phase and activity as well as maintenance effort. COCOMO estimates the effort in person months of direct labor. The primary effort factor is the number of source lines of code (SLOC) expressed in thousands of delivered source instructions (KDSI). The model is developed in three versions of different level of detail basic, intermediate, and detailed. The overall modeling process takes into account three classes of systems. Embedded: This class of system is characterized by tight constraints, changing environment, and unfamiliar surroundings. Projects of the embedded type are model to the company and usually exhibit temporal constraints. Organic: This category encompasses all systems that are small relative to project size and team size, and have a stable environment, familiar surroundings and relaxed interfaces. These are simple business systems, data processing systems,and small software libraries. Semidetached: The software systems falling under this category are a mix of those of organic and embedded in nature. Some examples of software of this class are operating systems, database management system, and inventory management systems.For Basic COCOMO Effort=a\*(KLOC)bType=c\*(effort)d For Intermediate and Detailed COCOMO

Effort = a \* (KLOC) b\* EAF (EAF = product of cost drivers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Type Of Product A | B C | D |  |
|  | 2.4 | 1.02 2.5 | 0.38 |  |
|  | 3.0 | 1.12 2.5 | 0.35 |  |
|  | 3.6 | 1.20 2.5 | 0.32 |  |

The above table gives the values a, b, c, d for Organic, Semidetached and Embedded

systems.

**5.4.1** **Product** **Attributes:**

15

Required reliability (RELY): It is used to express an effect of software faults ranging from slight inconvenience (VL) to loss of life (VH). The nominal value (NM) denotes moderate recoverable losses. Data bytes per DSI (DATA): The lower rating comes with lower size of a database. Complexity (CPLX): The attribute expresses code complexity again ranging from straight batch code (VL)to real time code with multiple resourcesscheduling (XH).

**5.4.2** **Computer** **Attributes:**

Execution time (TIME) and memory (STOR) constraints: This attribute identifies the percentage of computer resources used by the system. NM states that less than 50% is used; 95% is indicated by XH. Virtual machine volatility (VIRT): It is used to indicate the frequency of changes made to the hardware, operating system, and overall software environment. More frequent and significant changes are indicated by higher ratings. Development turnaround time (TURN): This is a time from when a job is submitted until output becomes received. LO indicated a highly interactive environment, VH quantifies a situation when this time is longer than 12 hours.

**5.4.3** **Personal** **attributes:**

Analyst capability (ACAP) and programmer capability (PCAP): describe skills of the developing team. The higher the skills the higher the rating. Application experience (AEXP), language experience (LEXP), and virtual machine experience (VEXP): These are used to quantify the number of experience in each area by the development team; more experience, higher rating.

**5.4.4** **Project** **attributes:**

Modern development practices (MODP): dealswith theamount of useofmodernsoftware practices such as structural programming and object oriented approach. Use of software tools (TOOL): is used to measure a level of sophistication of automated tools used in software development and a degree of integration among the tools being used. Higher rating describes levels in both aspects.Our project is an organic system and for intermediate COCOMO Effort = a \* (KLOC) b \*EAF KLOC = 115 For organic system a = 2.4 b = 1.02 EAF = product of cost drivers Effort = 2.4 \* (0.115)^ 1.02 \* 1.30 = 1.034

16

programmer months Time for development = C \* (Effort) d = 2.5 \* (1.034)^0.38 = 2.71 months = 1.034 \* 20000 = 20680 Project cost = 20000 + 20680 = 406

17

**CHAPTER** **6** **SYSTEM** **DESIGN**

**6.1** **SYSTEM** **ARCHITECTURE** **:**

The webcam is always streaming video, but we analyze a single frame every 0.1 seconds until we reach 10 samples, the equivalent of 1 second, to extract the facial landmarks and keep onlythe points corresponding to both eyes. We group the points with an overlapping of 7 units, which means that we group the points from frame one to ten, the next group is formed from the point of the frame four to the frame thirteen. Once we have a group of eyes’ points (x, y coordinates) we pass them to our neural model to get a classification, whose result can be [1, 0] which represents “awake”, or [0, 1] that represents “drowsy”.

**Figure:** **6.1** System Architecture of Drowsiness detection

18

**6.2** **MODULES:Eye** **detection:**

In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face, such as:

**Figure:** **6.2** shape\_predictor\_68\_face\_landmarks

**Recognition** **of** **Eye's** **State:**

The eye area can be estimated from optical flow, by sparse tracking or by frame-to-frame intensity differencing and adaptive thresholding. And Finally, a decision is made whether the eyes are or are not covered by eyelids. A different approach is to infer the state of the eye opening from a single image, as e.g. by correlation matching with open and closed eye templates, a heuristic horizontal or vertical image intensity projection over the eye region, a parametric model fitting to find the eyelids , or active shape models. A major drawback of the previous approaches is that they usually implicitly impose too strong requirementson the setup ,in the sense of a relative face-camerapose (head orientation), image resolution, illumination ,motiondynamics, etc. Especiallytheheuristic methodsthat use raw image intensity are likely to be very sensitive despite their real-time performance .Therefore, we propose a simple but efficient algorithm to detect eye blinks by using a

19

recent facial landmark detector. A single scalar quantity that reflects a level of the eye opening is derived from the landmarks. Finally, having a per-frame sequence of the eye-opening estimates ,the eye blinks are found by an SVM classifier that is trained on examples of blinking and non- blinking patterns.

**Eye** **Aspected** **Ratio** **Calculation:**

For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed. EAR =(||p2 − p6|| + ||p3 − p5||) /2(||p1 − p4||) where p1, . . ., p6 are the 2D landmark locations, depicted in Fig. 1. The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals, and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously,

**Figure:** **6.3** Eye Aspected Ratio

**Eye** **State** **Determination:**

20

Finally, the decision for the eye state is made based on EAR calculated in the previous step. If the distance is zero or is close to zero, the eye state is classified as “closed” otherwise the eye state is identified as “open”.

**Drowsiness** **Detection:**

The last step of the algorithm is to determine the person’s condition based on a preset condition for drowsiness. The average blink duration of a person is 100-400 milliseconds (i.e. 0.1-0.4 of a second). Hence if a person is drowsy his eye closure must be beyond this interval. We set a time frame of 5 seconds. If the eyes remain closed for five or more seconds, drowsiness is detected and alert pop regarding this is triggered.

**6.4** **UML** **DIAGRAMS** **USE** **CSE** **DIAGRAM:**

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is just a special kind of diagram and shares the same common properties as do all other diagrams a name and graphical contents that are a

21

projection into a model. What distinguishes a use case diagram from all other kinds of d

**Figure** **:** **6.4** Use case diagram of drowsiness detection

22

c

**Figure:6.5** Activity diagram of prediction of algorithms

**CLASS** **DIAGRAM:**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

**Sequence** **Diagram:**

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. We use sequence diagrams to illustrate the dynamic view .

23

**CHAPTER** **7** **SYSTEM** **TESTING**

**7.1** **Testing:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the 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** **White** **Box** **Testing:**

White Box testing is testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers. It is one of two parts of the Box Testing approach to software testing. Its counterpart, Black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing. The term "White Box" was used because of the see-through box concept. The clear box or White Box name symbolizes the ability to see through the software's outer shell (or "box")into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

**How** **do** **you** **perform** **White** **Box** **Testing?**

To give you a simplified explanation of white box testing, we have divided it into two basic steps. This is what testers do when testing an application using the white box testing technique:

24

**step** **1:** **understand** **the** **source** **code**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**step** **2:** **Create** **test** **cases** **and** **execute**

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include Manual Testing, trial, anderror testing.Thetester will develop little tests for eachprocess or series ofprocesses in the application. This method requires that the tester use of testing tools as we will explain further on in this article.

**White** **Box** **Testing** **Techniques:**

A major White boxtesting technique is Code Coverage analysis. Code Coverage analysis eliminates gaps inaTest Casesuite. Itidentifies areas of aprogram thatarenot exercised by a set of test cases. Once gaps are identified, you create test cases to verify untested parts of the code, thereby increasing the quality of the software product There are automated tools available to perform Code coverage analysis. Below are few coverage analysis techniques Statement Coverage:- This technique requires every possible statement in the code to be tested at least once during the testing process of software engineering. Branch Coverage - This technique checks every possible path (if-else and other conditional loops) of a software application. Apart from above, there are numerous coverage types such as Condition Coverage, Multiple Condition Coverage, Path Coverage, Function Coverage etc. Each technique has its own merits and attempts to

25

test (cover) all parts of software code. Using Statement and Branch coverage you generally attain 80-90% code coverage which is sufficient.

**Types** **of** **White** **Box** **Testing:**

White box testing encompasses several testing types used to evaluate the usability of an application, block of code or specific software package.

There are listed below **Unit** **Testing:**

It is often the first type of testing done on an application. Unit testing is performed on each unit or block of code as it is developed. Unit Testing is essentially done by the programmer. As a software developer, you develop a few lines of code, a single function or an object and test it to make sure it works before continuing Unit Testing helps identify a majority of bugs, early in the software development lifecycle. Bugs identified in this stage are cheaper and easy to fix.

**Testing** **for** **Memory** **Leaks**:

Memory leaks are leading causes of slower running applications. A QA specialist who is experienced at detecting memory leaks is essential in cases where you have a slow running software application. Apart from above, a few testing types are part of both black box and white box testing.

**7.3** **Black** **Box** **Testing:**

BLACK BOX TESTING is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications. In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications byjust focusing on the inputs and outputs without knowing their internal code implementation

26

**Types** **of** **Black** **Box** **Testing**

There are many types of Black Box Testing but the following are the prominent ones Functional testing . This black box testing type is related to the functional requirements of a system, it is done by software testers. • Non-functional testing - This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability. • Regression testing - Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

**Tools** **used** **for** **Black** **Box** **Testing:**

• Tools used for Black box testing largely depends on the type of black box testing • You are doing.

• For Functional/ Regression Tests you can use - QTP, Selenium • For Non-Functional Tests, you can use - LoadRunner, Jmeter

**Black** **Box** **Testing** **Techniques**

• Following are the prominent Test Strategy amongst the many used in Black box Testing • Equivalence Class Testing:

It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.

• BoundaryValue Testing: Boundaryvalue testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.

• Decision Table Testing: A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

**Black** **Box** **Testing** **and** **Software** **Development** **Life** **Cycle** **(SDLC)**

Black box testing has its own life cycle called Software Testing Life Cycle (STLC) and it is relative to every stage of Software Development Life Cycle of Software Engineering. **Requirement:**

This is the initial stage of SDLC and in this stage, a requirement is gathered. Software testers also take part in this stage.

27

**Test** **Planning** **&** **Analysis:**

Testing Types applicable to the project are determined. A Test Plan is created which determines possible project risks and their mitigation.

**Design:**

In this stage Test cases/scripts are created on the basis of software requirement documents

**Test** **Execution:**

In this stage Test Cases prepared are executed. Bugs if any are fixed and re- tested.

28

**CHAPTER** **8** **CONCLUSION**

The drowsiness detection and correction system developed is capable of detecting drowsiness in a rapid manner. The system can differentiate normal eye blink and drowsiness by which we can prevent the driver from entering the state of sleepiness while driving.

**Limitations:**

Use of spectacles: In case the user uses spectacle then it is difficult to detect the state of the eye. As it hugely depends on light hence reflection of spectacles may give the output for a closed eye as opened eye. Hence for this purpose the closeness of eye to the camera is required to avoid light.

**Multiple** **face** **problem:**

If multiple face arises in the window then the camera may detect more number of faces undesired output may appear. Because of different condition of different faces. So, we need to make sure that only the driver face come within the range of the camera. Also, the speed of detection reduces because of operation on multiple faces.

29

**REFERENCES**

1 . B Sangle, R Rathore, A Rathod, A Yadav, ; V Yadav, A Varghese, S Shenoy, K P Ks, ; R Remya, Patra, Driver drowsiness monitoring system using visual behaviour and machine learningISCAIE 2018 -2018 IEEE Symposium on Computer Applications and Industrial Electronics, volume 2018, p. 339 – 344.

2 . S Junawane, P Jagatap, L Aeshpande, K Soni: R Jabbar, M AI-Khalifa, W Kharbeche, M Alhajyaseen ,S Jafari, Jiang, Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques. Procedia Computer Science, volume 6, issue 11, p. 400 – 407.

3 . A Rosebrock (2017) facial landmarks with dlib, opencv and python.

4 . L. Celona, L. Mammana, S. Bianco, and R. Schettini, “A multi-task CNN framework for driver face monitoring,” IEEE International Conference on Consumer Electronics – Berlin, ICCE- Berlin, vol. 2018- Septe, pp. 1–4, 2018.

5 . M. Tayab Khan, H. Anwar, F. Ullah, A. Ur Rehman, R. Ullah, A. Iqbal, B.-H. Lee, and K. S. Kwak, “Smart Real-Time Video Surveillance Platform for Drowsiness Detection Based on Eyelid Closure,” Wireless Communications and Mobile Computing, vol. 2019, pp. 1–9, 2019.

6 . Y. Xie, K. Chen, and Y. L. Murphey, “Real-time and Robust Driver Yawning Detection with Deep Neural Networks,” Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018, pp. 532–538, 2019.

7 . R. Jabbar, K. Al-Khalifa, M. Kharbeche, W.Alhajyaseen, M. Jafari, and S. Jiang, “Real-time driver drowsiness detection for android application using deep neural networks techniques,” Procedia computer science, vol. 130, pp. 400–407, 2018.

8 . P. Peyrard, “Personal system for the detection of a risky situation and alert,” Feb. 28 2019, uS Patent App. 16/178,365.

9 . L. Celona, L. Mammana, S. Bianco, and R. Schettini, “A multi-task CNN framework for driver face monitoring,” IEEE International Conference on Consumer Electronics – Berlin, ICCE- Berlin, vol. 2018- Septe, pp. 1–4, 2018.

10 . N. C. for Statistics and Analysis, “Crash Stats: Drowsy Driving 2015,” October 2017. 11 . Zhao Y, Kong X, Taubman D. Image and Graphics. 9th International Conference on Image and Graphics (ICIG) Springer 2017.

30

12 .S. Tateno, X. Guan, R. Cao, and Z. Qu, “Development of Drowsiness Detection System Based on Respiration Changes Using Heart Rate Monitoring,” in 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2018, 2018, pp. 1664–1669.

13. C. S. Wei, Y. T. Wang, C. T.Lin, and T. P. Jung, “Toward Drowsiness Detection Using Non-hair-Bearing EEG-Based Brain-Computer Interfaces,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2018. Computers and Electronics in Agriculture, vol. 147, pp. 70 – 90, 2018 .

31