

### SCHEDULING REAL TIME SECURITY AWARE TASKS IN FOG NETWORKS

A MAJOR PROJECT REPORT

*Report Submitted in partial fulfillment of the requirements of the award of degree of*

### BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE & ENGINEERING**

*Submitted by*

E. Tejasri (19A21A0512)

B. Tejasri (19A21A0504)

T.Arundhathi (19A21A0563)

L.Sujee (20A25A0506)

**Under the Esteemed Guidance of**

**Dr.P.Srinivasulu Phd**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SWARNANDHRA COLLEGE OF ENGINEERING & TECHNOLOGY**

**(AUTONOMOUS)**

**Accredited by National Board of Accreditation, AICTE, New Delhi (Approved by AICTE & Affiliated to JNTUK Kakinada)**

**An ISO 9001:2000 certified Institution**

**Seetharampuram, Narasapur-534280, West Godavari (Dt.), A.P 2022-2023**

**SWARNANDHRA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

###### (Accredited by NBA and NAAC with 'Grade (CGPA 3.32/4)) Approved by AICTE, Autonomous with JNTU Kakinada) Seetharampuram, Narsapur-534280

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the Major Project Report entitled "**SCHEDULING REAL TIME SECURITY AWARE TASKS IN FOG NETWORKS”,** is being submitted by **E.TEJASRI (19A21A0512), B.TEJASRI (19A21A0504), T.ARUNDHATHI (19A21A0563), L.SUJEE (20A25A0506)** Of B.Tech VIII Semester who carried out the work under my supervision and submitted in partial fulfillment for the award of the degree of "**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**" during the academic year 2022-2023 and it has been found worthy of acceptance according to the requirements of the autonomous.

###### Project Guide Head of the Department

**Dr P.SRINIVASULU Dr P.SRINIVASULU**

###### Professor Professor

EXTERNAL EXAMINER

### ACKNOWLEDGEMENT

Foremost ,We want to offer the endeavor to our GOD Almighty for the wisdom he bestowed upon us,the strength, peace of mind and good health in order to finish this Major Project.

We would like to express my sincere thanks to our parents for the encouragement for everything.

We are hugely indebted to **Dr. S. Suresh Kumar,** Principal for finding out time for us to share knowledge for being ever so kind to show interest in our work and for giving his precious and kind advice for the same.

We would like to express our grateful thanks to **Dr. P. Srinivasulu**, Professor & Head, Department of Computer Science and Engineering, for his valuable suggestions and guidance regarding the software analysis, Design and also for his continuous effort in the successful completion of the Project.

Our deepest appreciation to the Major Project Coordinator Mrs. **G. Archana**, Assistant Professor, Department of Computer Science and Engineering for her valuable guidance for efficacious completion of the Major Project.

We would like to express our sincere gratitude and heartfelt thanks to our Major project guide **,** or his continuous support, patience, motivation, enthusiasm, and immense knowledge helped us successfully carried out this assignment.

We also place on record, Our thanks to one and all who, directly or indirectly, have lent their helping hand in this venture.

**E.TEJASRI (19A21A0512)**

**B.TEJASRI (19A21A0504)**

**T.ARUNDHATHI (19A21A0563)**

**L.SUJEE (20A25A0506)**

#### DECLARATION

We certify that

1. The project work contained in the report is original and has been done by under the guidance of my supervisor.
2. The work has not been submitted to any other university for the award of any degree or diploma.
3. The guidelines of the university are followed in writing the report.

Date:

Place:

**NAMEOFTHESTUDENT REG.NO SIGNATURE E. TEJASRI 19A21A0512**

**B. TEJASRI 19A21A0504**

**T.ARUNDHATHI**

**L.SUJEE**

**19A21A0563**

**20A25A0506**

CHAPTER 1

# 1.INTRODUCTION

Internet of Things (IoT) enabled devices are now increasingly generating large amounts of data. In real-time systems, there is a requirement for this data to be processed within a specified deadline [9]. Although numbers may vary (across Gartner, Cisco and other market forecasts), it is estimated that we will have multi-billion Internet enabled devices by 2020 [26]. Hence, there is a strong need to construct distributed systems that can be successful in analyzing “big data” produced from these IoT devices. There has also been significant hardware innovation over recent years, with servers of the not so distant past being of comparable performance to current day mobile phones. Availability of such user-owned devices have enabled processing of data intensive applications in geographical proximity to users. A cloud data center (cdc) can be used to process the data that is generated by such IoT devices [15], and is the most dominant execution mode currently being used. The downside is the significant network latency involved between the cdc and the IoT devices. Real-time applications would however miss their processing deadlines by the time their data reaches the cdc [18]. Multi-user gaming, image/video rendering, audio/video content streaming, smart & autonomous cars, etc. are some examples of real-time applications that have such latency-sensitive processing requirements. By using the network edge to perform as much computation as possible [3], one can get around the latency that would be involved if application data was sent across the network to the cloud. Switches, routers, gateways are examples of some edge devices that can execute jobs that would have been scheduled to run at the cloud data center. Bittencourt, Lopes, Petri and Rana [15] and Dastjerdi, Gupta, Calheiros, Ghosh and Buyya [17] discuss the edge computing paradigm, where access points are used by applications to retrieve and transfer the data to a cdc. These access points may be enhanced to provide both storage and computation capacity at the edge of the network, and would be referred to as mdcs (also known as cloudlets). The communication between phones, mdcs and the cdc is as follows: smart phones ⇔ mdcs ⇔ cdc. Moreover, peerto-peer communication between the various mdcs is also possible. This communication is needed for the storage of the execution states of applications, and may be used to preempt applications from their local mdc, on the account of mobility, and to later resume applications on a new mdc. Additionally, based on their characteristics and QoS requirements, applications may broadly be classified into two categories: interactive and batch [29]. Interactive tasks are typically less compute intensive, and require real-time performance, i.e. the task should be finished by a specified deadline. Batch jobs, on the other hand, are more compute intensive, and may not have a real-time requirement. Intuitively, interactive tasks may be executed on the edge, i.e. on the mobile data centers (mdcs) – to limit the latency associated with sending data to a cloud data center. Two types of cdcs have been considered for the execution of batch jobs: private and public cdcs. Private cdcs are more secure than public cdcs, but this is provided at an extra cost [32]. Applications that require high security and can tolerate latency can be run on these secure cdcs [33], [34]. Privacy and security capabilities in Cloud computing depend on security controls offered by a provider. These can range from the types of encryption algorithms they support, facilities for data anonymisation to hosting locations of data centres employed as part of their deployment strategy. Rahulamathavan and others [37] investigated risk of a data breach (i.e. user data privacy concern) associated with data center hosting. The security tags identified in this work are based on the analysis undertaken in [37] – whereby data privacy is associated with the types of capabilities offered by a provider and included as a tag to support discovery of an mdc or cdc (depending on the audited security capability based on these security controls). A similar approach is adopted by the Cloud Security Alliance, which makes use of self-certification through a “Cloud Controls Matrix” security methodology for Cloud providers 1 . We consider 16 data privacy controls from [37] to support the inclusion of such security tags for both mdcs and cdcs. In this paper, we introduce a scheduling algorithm called RT-SANE, that addresses both the privacy/security and real-time performance requirement of application jobs amalgamating an mdc and a cdc. In RT-SANE, interactive applications that are private to a user are constrained to run only on their local mdc, while private batch applications are constrained to be run on the private cdc. Applications that are semi-private (i.e. those that involve use of a data set held at the cloud data center) are sent to the local cdc (either private or public). Finally, applications that belong to the public category, may be executed at a remote mdc or a cdc (either private or public). As interactive jobs have stringent deadline requirements, they are executed on the local or foreign mdcs, provided one with spare capacity is available. Batch jobs are assumed to have loose/flexible or no deadlines, and are executed on the cdcs. The rest of this paper is structured as follows. Related work covering combined use of fog/edge resources and a cloud system is discussed in section 2. Section 3 introduces a novel distributed orchestration architecture and protocol. Section 4 discusses the system model and provides a formulation of the research problem. The proposed algorithm RT-SANE is described in section 5. Section 6 talks about the results of various simulations that have been carried out. Finally, section 7 provides conclusions that can be drawn from this work.

**CHAPTER 2 LITECTURE SURVEY**

1. Driver drowsiness in commercial truck drivers is a major concern and is responsible for thousands of accidents and fatalities every year. In a 1994 report (Knipling 1994), the Office of Crash Avoidance Research (OCAR) of the National Highway Traffic Safety Administration (NHTSA) identified driver drowsiness as one of the leading causes of single and multiple car accidents. NHTSA estimates that 100,000 crashes annually involve driver fatigue resulting in more than 40,000 injuries.
2. Driver's drowsiness can be measured by two classes of phenomena: Physical and physiological and Vehicle state variables. Physical and physiological measurements include the measurement of brain wave or Electroencephalogram (EEG) (Akerstedt and Gillberg 1990; Huang, Kuo et al. 1996), eye activity (Skipper, Wierwille et al. 1984; Dingus, Hardee et al. 1985; Ueno, Kaneda et al. 1994; Ogawa and Shimotani 1997). PERCLOS (PERcent eyelid CLOSure) is one of the most widely accepted measures in scientific literature for measurement and detection of drowsiness (Dinges, Mallis et al. 1998; Grace, Byrne et al. 1998).
3. Drowsiness detection systems have been developed which work based on measurement of Physical and physiological features, and can provide very good detection accuracy. However, they have some shortcomings. The problem with an EEG is that it requires the use of electrodes to be attached to the scalp and that makes it very impractical to use. Eye closure activity can also provide good detection accuracy, but capturing eye image unobtrusively can be expensive and challenging under certain conditions.
4. Fatigue has been estimated in 15% of single vehicle fatal truck crashes (Wang and Knipling 1994) and is the most frequent contributor to crashes in which a truck driver is fatally injured (NTSB 1990). Based on NHTSA General Estimates System (GES) statistics (Knipling and Wierwille 1994), although the frequency of drowsiness related crashes involving passenger vehicles is greater than that of combination-unit trucks, the number of involvements per vehicle life cycle for trucks is about 4 times greater due to their very high exposure level, as well as the greater likelihood of night driving.
5. With respect to Vehicle State Variables Measurement, other approaches for detecting driver drowsiness are based on monitoring driver inputs or vehicle output variables during driving. These methods have the advantage of being non-intrusive to the drivers. Wierwille et al. (1992) discussed the performance measures as indicator of driver drowsiness in detail.
6. Robert Gabriel Lupu has discussed that in the previous year’s many algorithms for eye pupil/iris detection have been developed. Depending upon the source light point of view there are two

approaches namely based on ambient or infrared light. All of them search for characteristics of the eye. There are some algorithms that search for features like blackest pixels in the image, pixels that correspond to pupil or iris and are known as feature based algorithms.

1. Dongheng Li, Derick J. Parkhrust has discussed that Starburst algorithm is a robust eye-tracking algorithm that combines feature-based and model-based approaches to achieve a good trade-off between run-time performance and accuracy for dark-pupil infrared imagery.
2. V. Starovoitov and D. Samal have discussed, Geometric feature learning methods extract distinctive geometric features from images. Geometric features are features of objects constructed by a set of geometric elements like points, lines, curves or surfaces. These features can be corner features, edge features, Blobs, Ridges, salient points image texture and so on, which can be detected by feature detection methods

#### Introduction to Literature Survey What is Drowsiness

Drowsiness is defined as a decreased level of awareness portrayed by sleepiness and trouble in staying alarm but the person awakes with simple excitement by stimuli. It might be caused by an absence of rest, medicine, substance misuse, or a cerebral issue. It is mostly the result of fatigue which can be both mental and physical. Physical fatigue, or muscle weariness, is the temporary physical failure of a muscle to perform ideally. Mental fatigue is a temporary failure to keep up ideal psychological execution. The onset of mental exhaustion amid any intellectual action is progressive, and relies on an individual's psychological capacity, furthermore upon different elements, for example, lack of sleep and general well- being. Mental exhaustion has additionally been appeared to diminish physical performance. It can show as sleepiness, dormancy, or coordinated consideration weakness. In the past years according to available data driver sleepiness has gotten to be one of the real reasons for street mishaps prompting demise and extreme physical injuries and loss of economy. A driver who falls asleep is in an edge of losing control over the vehicle prompting crash with other vehicle or stationary bodies. Keeping in mind to stop or reduce the number of accidents to a great extent the condition of sleepiness of the driver should be observed continuously.

The term “drowsy” is substitutable with sleepy, that merely means that an inclination to fall asleep. The stage of sleep is often classified as awake, non-rapid eye movement sleep (NREM), and rapid eye movement sleep(REM). The second stage, NREM, is often divided into the subsequent 3 stages.

* + - Stage I : Transition from awake to asleep(drowsy)
    - Stage II : Light- Weight sleep
    - Stage III: Deep sleep

#### Features, which Drowsiness Depends On

One of the challenges in developing an economical drowsiness detection system is a way to acquire proper drowsiness information. Because of safety reasons, drowsiness cannot be manipulated during a real environment, Therefore the drowsiness detection system needs to be developed and tested in

a laboratory setting. However, in a laboratory setting, the foremost reliable and informative information that pertains to driver drowsiness depends only on the approach in which the driver falls into the drowsy state. Driver drowsiness principally depends on the quality of the last sleep, the biological time (time of day) and the rise within the period of the driving task. In some analysis experiments, the subjects were totally deprived of sleep, whereas they were only part deprived of sleep in others. Additionally, some researchers recruited night shift staff as their subjects, in this case, the subjects were entirely deprived of sleep as results of the experiments were conducted within the morning. Conducted an experiment during which they monitored the participants for twenty four before the experiment began to make sure that they were utterly sleep deprived.

In certain experiments, researches partly deprived the subjects of sleep by permitting them to sleep for less than a half dozen. Peters, et al. Studies an equivalent subject throughout four consecutive days and regarded the results of no sleep deprivation, partial sleep deprivation and total sleep deprivation on their drowsiness level [2]. They discovered that, even within the case of partial sleep deprivation, the subjects tend to urge drowsy after a while. Hence, the standard of the last sleep is a crucial criterion that influences drowsiness. Otamani, et al. Found that sleep deprivation alone doesn’t directly influence the brain signals that control, drowsiness, whereas the period of the task includes a strong influence [3]. Researchers have additionally inferred that prolonged driving on a boring setting stimulates drowsiness. In fact, it has been discovered that the subjects will become drowsy at intervals twenty to twenty five min of diving.

###### Driver Drowsiness Detection System and Techniques :

According to the experts it has been observed that when the drivers do not take break they tend to run a high risk of becoming drowsy. Study shows that accidents occur due to sleepy drivers in need of a rest, which means that road accidents occurs more due to drowsiness rather than drink-driving. Attention assist can warn of inattentiveness and drowsiness in an extended speed range and notify drivers of their current state of fatigue and the driving time since the last break, offers adjustable sensitivity and, if a warning is emitted, indicates nearby service areas in the COMAND navigation system.

###### Implementation of the Driver Drowsiness Detection System:

This paper is about making cars more intelligent and interactive which may notify or resist user under unacceptable conditions, they may provide critical information of real time situations to rescue or police or owner himself Driver fatigue resulting from sleep disorders is an important factor in the increasing number of accidents on today's roads. In this paper, we describe a real-time safety prototype that controls the vehicle speed under driver fatigue To advance a system to detect fatiguesymptoms in drivers and control the speed of vehicle to avoid accidents is the purpose of such a mode. In this paper, we propose a driver drowsiness detection system in which sensor like eye blink sensor are used for detecting drowsiness of driver .If the driver is found to have sleep, buzzer will start buzzing and then turns the vehicle ignition off.

###### Eye Tracking Based Driver Drowsiness Monitoring And Warning System:

This project represents a way of developing an interface to detect driver drowsiness based on continuously monitoring eyes and algorithms Micro sleeps are the short period of sleeps lasting 2 to 3 seconds, are good indicator of fatigue state. Thus by monitoring continuously the eyes of the driver by using camera one can detect the sleepy state of driver and timely warning is issued. Aim of the project is to develop the hardware which is very advanced product related to driver safety on the roads using controller and image processing This product detects driver drowsiness and gives warning in form of alarm and it also decreases the speed of vehicle.

###### Driver Drowsiness Detection System:

One of the major cause of traffic accident is Driver‘s drowsiness. It is a serious highway safety problem. If drivers could be warned before they became too drowsy to drive safely, some of these crashes could be prevented. In order to reliably detect the drowsiness, it depends on the presentation of timely warnings of drowsiness To date, the effectiveness of drowsiness detection methods has been limited by their failure to consider individual differences. Based on the type of data used, drowsiness detection can be conveniently separated into the two categories of intrusive and non-intrusive methods During the survey, non-intrusive methods detect drowsiness by measuring driving behaviour and sometimes eye features, through which camera based detection system is the best method and so are useful for real world driving situations

### TECHNIQUES

#### Techniques for Detecting Drowsy Drivers

Possible techniques for detecting drowsiness in drivers can be generally divided into the following categories: sensing of physiological characteristics, sensing of driver operation, sensing of vehicle response, monitoring the response of driver.

#### Monitoring Physiological Characteristics

Among these methods, the techniques that are best, based on accuracy are the ones based on human physiological phenomena. This technique is implemented in two ways: measuring changes in physiological signals, such as brain waves, heart rate, and eye blinking; and measuring physical changes such as sagging posture, leaning of the driver’s head and the open/closed states of the eyes. The first technique, while most accurate, is not realistic, since sensing electrodes would have to be attached directly onto the driver’s body, and hence be annoying and distracting to the driver. In addition, long time driving would result in perspiration on the sensors, diminishing their ability to monitor accurately. The second technique is well suited for real world driving conditions since it can be non-intrusive by using optical sensors of video cameras to detect changes.

#### Other Methods

Driver operation and vehicle behaviour can be implemented by monitoring the steering wheel movement, accelerator or brake patterns, vehicle speed, lateral acceleration, and lateral displacement. These too are non-intrusive ways of detecting drowsiness, but are limited to vehicle type and driver conditions. The final technique for detecting drowsiness is by 12 monitoring the response of the driver. This involves periodically requesting the driver to send a response to the system to indicate alertness.

#### Methods Focusing on Driver’s Performance

In order to detect drowsiness, studies on driver’s performance use lane tracking, distance between driver’s vehicle and the vehicle in front of it; place sensors on components of the vehicle such as steering wheel, gas pedal and analyze the data taken by these sensors. Pilutti and Ulsoy used vehicle lateral position as the input and steering wheel position as the output and they obtained a model which can be useful to detect drowsiness.

Some of the previous studies make use of driver steering wheel movements and steering grips as an indicator to detect drowsiness. Some car companies such as Nissan and Renault used this technology. Since these systems are too dependent on the characteristics of the road, they can only function well on motorways which make them work in limited situations.These systems are affected too much by the road quality and lighting. Another disadvantage of these systems is that they cannot detect drowsiness that has not affected vehicle’s situation yet. When a driver is drowsy and the vehicle is in the appropriate lines, these systems cannot detect drowsiness.

#### Methods Focusing on Driver’s State

The methods use physiological signals such as Electroencephalography(EEG), heart rate variability (HRV), pulse rate and breathing. The spectral analysis of heart rate variability shows that HRV has three frequency bands: high frequency band (0.15- 0.4 Hz), low frequency band (0.04-0.15 Hz) and very low frequency band (0.0033- 0.04Hz). Researchers have found out that LF/HF ratio decreases and HF power increases when a person goes from alert state to drowsy state. Power spectrum of EEG brain waves is used as an indicator to detect drowsiness; as drowsiness level increases, EEG power of the alpha and theta bands increases and beta band decreases[18][19]. EEG-based drowsiness detection methods are not easily implementable because they require the driver to wear an EEG cap during driving the vehicle. Devices being distractive is the main disadvantage of this group of methods.

#### Methods Using Computer vision

This group of methods are not offensive and does not make any disturbance to the driver, that’s why these methods are more pereferable. These methods are seperated into two groups: the ones using infrared illumination and the ones using day illumination. The former find the location of the eyes anddetect eye states by making use of retinal reflections of infrared waves.Matsuo and Khiat, who work for NISSAN, divided driver’s condition into 4categories: normal (alert), slightly sleepy, intensely sleepy and drowsy.They used eye closure rate (ECR), head sway and subsidiary behaviours todetect the driver’s condition. IR- based systems work well at night but do notwork well during daylight illumination since sunlight reflections make itimpossible to detect retinal reflections of infrared waves.

Methods using daylight illumination generally find the location of the face and eyes by making use of computer vision techniques. Driver State Sensor (DSS), developed by SeeingMachines, is a commercial product in this group. DSS uses face tracking and gets information about eyelid opening and percentage of eye closure in order to detect drowsiness. Viola Jones used the Haar-like features for face and eye detection and this method is used in many studies. Wu et al. uses adaboost classifier for face detection and use intensity image to detect the pupils of eyes. Then, they take radial-symmetry transform and use support vector machine (SVM) to find the location of eyes. After finding the location of eyes, using local binary patterns (LBP) and SVM classifier, they determine the eye state. Flowchart of their system isseen on Figure.

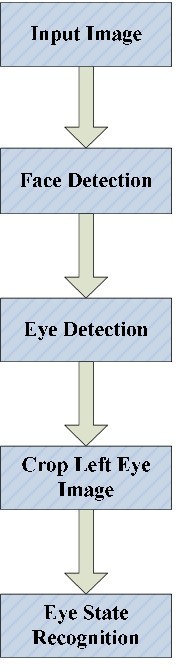


Fig 2.5: Flowchart of eye state recognition

Support vector machine is used by Flores et al. in order to identify eye state of a driver. They have used their drowsiness detection system as a new module for Advanced Driver Assistance System(ADAS). In, they assumed that eye region is detected successfully and then they use the eye region image to detect whether the eye is fully open, partially open or fully closed. They use singular value decomposition (SVD) and the term called eigen-eyes and decide which group the eye image belongs to. In addition to eye closure, yawning data is used as an indicator of drowsiness. They merged mouth and eye state data to detect drowsiness. However, Vural et al. states that yawning, which is assumed to be predictive of drowsiness, is a negative predictor of the 60-second window prior to crash. Drivers yawn less in the moments before falling asleep. They use facial movements in order to detect drowsiness. Facial movements are extracted by the help of a toolbox called Computer Expression Recognition Toolbox (CERT), which is a fully- automatic tool for facial expression recognition. They use facial action coding system (FACS) and find out which action units have postive or negative correlations with drowsiness and then by making use of this action unit data, they decide whether the driver is drowsy or alert.

#### Approach To The Solution

The method we propose belongs to the group which focuses on driver’s state by making use of computer vision. Eye closure rate, in other words,percentage of eye closure (PERCLOS) is a reliable measure to detect drowsiness [48]. This thesis makes use of PERCLOS to decide whether the driver in a video segment is drowsy or alert. For every frame in the video segment, eye state estimation is performed. There are 3 states of an eye: open eye, semi-closed eye and closed eye. The estimations for each frame in a video segment are combined and the driver’s state is estimated.

The video segment is extracted to its frames. After extraction of the frames of the video segment, the frames are input to the part called eye region extractor.Eye region extractor firstly finds the candidates for right and left eye regions and face by making use of extended version of Viola-Jones algorithm, which is available in Computer Vision System Toolbox of MATLAB [7] [8] [9]. Among the candidates of face, the wrong candidates are eliminated by some decision rules and the face region is detected. The detected face region is used to select the valid right and left eye region among the candidates found by extended version of Viola-Jones. After the detection of both eye regions, the eye images are converted to grayscale, resized to [12 18] and histogram equalized. After this process, every right and left eye image is input to neural networks separately which are trained with the subject’s eye region images. For each frame, the outputs of right and left eye neural networks are both digitized and merged in order to estimate the eye state of the subject. After eye state estimation for all of the frames of a video segment is completed, the mean of the estimated eye states is calculated by assigning “0” to open eyes, “0.5” to semi-closed eyes and “1” to closed eyes. The mean value obtained is called “eye closure point per frame” and an eye closure point per frame more than a threshold value means a drowsy driver, whereas an eye closure point per frame less than a threshold value means an alert driver.

Combining the estimations for right and left eyes increases the accuracies for both eye state and drowsiness detection. Since combination of the estimations for right and left eyes is not a common method used in the literature, increasing the accuracy with this method is a contribution of our proposed algorithm. Most of the studies assign eyes only two states: open and closed. As another contribution, this study reveals the fact that semi-closed state has an important role in detecting drowsiness and defining three states instead of two states increases the accuracy of the drowsiness detection method proposed.

#### MATERIAL AND METHOD

In this study, real-time camera image data of four volunteers were used to test the driver’s sleepiness. Camera images are taken from a driving simulation. The driver images taken were reduced to 320x240 spatial resolution and converted to gray scale with th In this Equation, Y represents pixel brightness, R is red, G is green and B is blue color brightness.



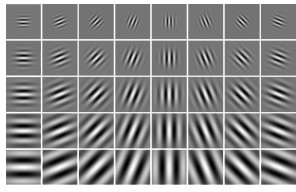
Since color information is not used in the image processing algorithms to be used, only the gray scale image is processed. The face and eye images of the driver are detected and cropped by the Viola-Jones detector

method. The Viola-Jones detector is an AdaBoost classifier that uses Haar-like features. AdaBoost classifiers train T amount of ht weak classifiers which usually consist of independent and single level binary decision trees.

An αt weight value is given for each classifier. As the input data set, xi feature vectors labeled with a yi binary tag are used, each of which is only -1 and +1. Finally, the class of the xi input is calculated by Equation



In this equation, H(x) is the class of the x sample, ht is the weak classifier, and αt is the classifier coefficient. The Sign function returns 1 for all positive values and -1 for all negative values. Zero values returned as zero. An example showing the application of Haar-like features to the facial image is shown in Figure 1. After finding the facial image, the left and right eye areas

were cropped using the geometric ratios of the face and

the left and right eye images were simultaneously

found in these areas using parallel tasks [2]. The eye

images were combined to obtain a minimum size

image containing both eyes. 40 Gabor filters were

used at eight different angles and five different

scales to extract the features of these images. Gabor

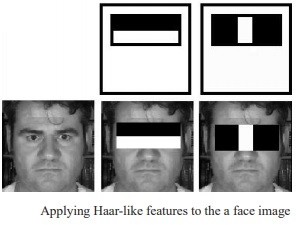
filters define sine and cosine functions within a

Gaussian window. Two dimensional Gabor wavelets

are obtained by using two dimensional forms of

these functions. The real (even) and imaginary (odd)

components of two-dimensional Gabor wavelets are denoted as Applying Haar-like feature to recognize facial structure

The total duration of the spontaneous closing and opening phases of the human eyelid is approximately 334 milliseconds The classification speed of the IBk algorithm is almost half the speed of the blink time. However, the J48 decision tree has more than enough speed to catch an eye blink easily. For this reason, we concluded that the J48 algorithm, which is 2.78% lower in accuracy, is suitable for real-time classification. A real-time software has been developed for detection of driver drowsiness using personal data. This software classifies the eyes as open or closed by cropping the

driver’s face and eye images in a real-time video image through the model trained beforehand by machine learning algorithms.

Recognizing as classifier

### CHAPTER 3

* 1. **Introduction**

### ANALYSIS

Analysis of the Existing system, helps in designing problem statement of Proposed system. In the following section based on the analysis of existing system, the requirements of proposed system has been defined.

#### Existing System

* Edge computing infrastructure can provide benefit for applications with stringent latency and response time requirements, such as gaming and stream processing, enabling some initial processing to be carried out closer to the user device/data generation source. Additionally, where the network connecting a user device to a cloud data center can fail or have a variable availability profile (i.e. network Quality of Service can change significantly over time, in unpredictable ways), edge resources can either: (i) support an approximate version of capability that would be carried out within a data center [30], or (ii) enable adaptation of a pre-generated model to be carried out [27], enabling subsequent re-synchronisation of this model with the cloud once the network connection is re-established. The use of edge resources also has a bearing on issues around data ownership and trust in a cloud data center provider, as data shared with a cloud provider can be directly viewed and searched.
* Scheduling across edge computing resources has also been explored by a number of authors. In [25], iFogStor and iFogStorZ are proposed to support scheduling – the first uses an Integer Linear Programming-based approach to find an optimal result, whereas the second uses a heuristic to create an approximate result at lower computational cost. However many of these approaches [24], [25] do not consider application deadlines. A mobility-aware scheduling algorithm (and a survey) is proposed in [12], but no support for deadline-centric tasks is provided. This aspect also aligns with focus on understanding how services can be mapped to edge resource, considering a group of possible edge nodes on which such resources can be hosted. Skarlat and others [19] show how this can be modelled as an optimization problem, focusing on the reduction of communication delay between different services within a workflow (using a genetic algorithm to find possible solutions to this problem) and realized through the use of a cloud-fog middleware.
* Some researchers have characterized and compared “real-life” workloads [28], [29]. In [29], Google’s data center workloads have been studied, including over 25 million tasks, spread over 12,500 hosts. The following characteristics were studied – job length, job submission frequency, job resource utilization (both CPU & memory). Two kinds of jobs were studied: short interactive jobs and long grid jobs. These workloads may be fed as input to algorithms that schedule jobs on fog networks.

#### Proposed Systems

1. In the proposed system, the system introduces a scheduling algorithm called RT-SANE, that addresses both the privacy/security and real-time performance requirement of application jobs amalgamating an mdc and a cdc. In RT-SANE, interactive applications that are private to a user are constrained to run only on their local mdc, while private batch applications are constrained to be run on the private cdc. Applications that are semi-private (i.e. those that involve use of a data set held at the cloud data center) are sent to the local cdc (either private or public). Finally, applications that belong to the public category, may be executed at a remote mdc or a cdc (either private or public). As interactive jobs have stringent deadline requirements, they are executed on the local or foreign mdcs, provided one with spare capacity is available. Batch jobs are assumed to have loose/flexible or no deadlines, and are executed on the cdcs.

#### Algorithm Stages 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. Logitech Camera is used for this process is shown in figure



Fig3.3.2: Camera used for implementing drowsiness detection system

#### Dividing into Frames

We are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analyzing.

#### Object Detection

Object detection is commonly defined as method for discovering and identifying the existence of objects of a certain class. Also it can be considered as a method in image processing to find out an object from images. There are several ways to classify and find objects in a frame. Out of that one way can be based on color identification. But it is not an efficient method to detect the object as several different size object of same color may be present. Hence a more efficient way is Haar-like features, developed by Viola and Jones on the basis of the proposal by Papageorgiou et. al in 1998

. Haar-like features are digital image features used in object detection. Or we can say that these are rectangle shaped dark and light areas having similar kind of features like our face. The cascade classifier comprises of a number of stages, where each stage consists of many weak features. The system detects objects by moving a window over the entire image and by forming a strong classifier.

The output of each stage is labeled as either positive or negative– positive meaning that an object was found and negative means that the specified object was not found in the image.

#### Face Detection

In this stage we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame. By face detection we means that locating the face in a frame or in other words finding location of facial characters through a type of technology with the use of computer. The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

#### Eye Detection

After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it’s really quite complicated.

At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

Our blink detection blog post is divided into four parts.

In the first part we’ll discuss the *eye aspect ratio* and how it can be used to determine if a person is blinking or not in a given video frame.

From there, we’ll write Python, OpenCV, and dlib code to (1) perform facial landmark detection and

(2) detect blinks in video streams.

Based on this implementation we’ll apply our method to detecting blinks in example webcam streams along with video files

#### State of eye

In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts.

We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye.

#### Advantages

The various advantages of the implemented system are mentioned below

1. Detection of drowsiness
2. Decreasing road accidents
3. No need of monitoring cameras or other devicesare attached or aimed at the driver.
4. This method is practically applicable.

#### APPLICATIONS

The drowsiness detection system can be used for different applications. One of them is heavy vehicles for example trucks, since the drivers of trucks have long driving periods. It can also be used for commercial vehicles. Many people use public transport facility for travelling. For their safety this system can be used in public vehicles. Heavy things are lifted by using cranes and transporting them to other places. So for overloaded cranes and mobile cranes this system can be used to avoid accidents related to drowsiness.

#### System used

**3.4.1 System Configuration Background and Ambient Light**

Because the eye tracking system is based on intensity changes on the face, it is crucial that the background does not contain any object with strong intensity changes. Highly reflective object behind the driver, can be picked up by the camera, and be consequently mistaken as the eyes. Since this design is a prototype, a controlled lighting area was set up for testing. Low surrounding light

(ambient light) is also important, since the only significant light illuminating the face should come from the drowsy driver system. If there is a lot of ambient light, the effect of the light source diminishes. The testing area included a black background, and low ambient light (in this case, the ceiling light was physically high, and hence had low illumination). This setup is somewhat realistic since inside a vehicle, there is no direct light, and the background is fairly uniform.

#### Camera

The drowsy driver detection system consists of a CCD camera that takes images of the driver’s face. This type of drowsiness detection system is based on the use of image processing technology that will be able to accommodate individual driver differences. The camera is placed in front of the driver, approximately 30 cm away from the face.

The camera must be positioned such that the following criteria are met:

1. The driver’s face takes up the majority of the image.
2. The driver’s face is approximately in the centre of the image.

The facial image data is in 480x640 pixel format and is stored as an array through the predefined Picolo driver functions (as described in a later section).

#### Light Source

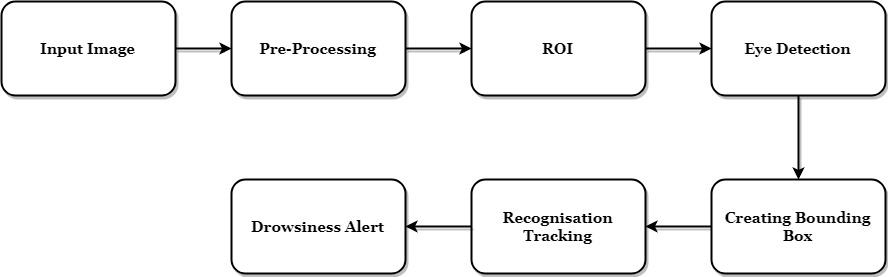
For conditions when ambient light is poor (night time), a light source must be present to compensate. Initially, the construction of an infrared light source using infrared LED was going to be implemented.

It was later found that at least 50 LEDs would be needed so create a source that would be able to illuminate the entire face. To cut down cost, a simple desk light was used. Using the desk light alone could not work, since the bright light is blinding if looked at directly, and could not be used to illuminate the face.

However, light from light bulbs and even daylight all contain infrared light; using this fact, it was decided that if an infrared filter was placed over the desk lamp, this would protect the eyes from a strong and distracting light and provide strong enough light to illuminate the face. A wideband infrared filter was placed over the desk lamp, and provides an excellent method of illuminating the face.

* 1. **Architecture Diagram**

### CHAPTER 4 DESIGN



###### Fig 4.1: Block diagram for driver drowsiness detection system

In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will classify whether the person’s eyes are ‘Open’ or ‘Closed’.

The approach we will be using for this Python project is as follows:

* + - Take Image as Input from a Camera.
    - Detect Face in the Image and Create a Region of Interest (ROI).
    - Detect the eyes from ROI and feed it to the classifier.
    - Classifier will Categorize whether Eyes are Open or Closed.
    - Calculate Score to Check whether Person is Drowsy.
  1. **Introduction to System Design**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.The purpose of the System Design is to supplement the system architecture providing information and data useful and necessary for implementation of the system elements.

#### Inputdesign

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

#### Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer- based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

#### Outputdesign

A quality output is one, which meets the requirements of the end user and presents theinformation clearly. In any system results of processing are communicated to the usersand to other system through outputs. In output design it is determined how theinformation is to be displaced for immediate need and also the hard copy output. It isthe most important and direct source information to the user. Efficient and intelligentoutput design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought outmanner; the right output must be developed while ensuring that each output element isdesigned so that people will find the system can use easily and effectively. Whenanalysis design computer output, they should Identify the specific output that is neededto meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by thesystem.

The output form of an information system should accomplish one or more of thefollowing objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.
  1. **Data Flow Diagram**

Captured frame

Next frame

Face

false

## Eye

true

## false



Eye state determination

Determine pixel intensity of center and surrounding

Detect the eye center

Fig 4.3 Data flow Diagram for Driver Drowsiness Detection System

**Driver drowsiness**

## false

* The processing will target to detect the drivers face from the video stream, once the face is detected, the region of interest that is the eyes will then be located from the facial features.

Drowsiness alert

* The state of t value.

he eye will then be computed using the pixel i

ntensity difference and a threshold

* With eye found in the face region, detection of the face is important to finding the eyes.



* They are computed from the pixel intensity value of the image.
* Once the eye region is obtained, it is cropped and moved to the next level where the state (closed/open) of the eye is determined.
* The two default eye states are open and closed, from the real world set up an eye is open when one can see, however from the computer perspective an eye is considered open when the iris and the white component of the eye are visible.
* To be able to detect this states color information of different parts both the eye and the surrounding is very important.
* In order to use color information a color space needs to be selected.
* Color intensity space is most appropriate for segmenting the eye region.
* The RGB can also be used but the eye has no enough features of the Red, Green and Blue colors for complete segmentation.
* The drowsiness level in the driver a percentage of the number of the frames with drowsy eyes is obtain against the total number of detected frames, the percentage is obtained in loops of 200 frames, which represent blocks of 20 sec.
* A total frame counter is used to calculate the total number of frames received, another counter drowsy counter establishes the total number of drowsy frames that have been detected and then a counter normal record the number of frames with eye that are alert.
* Once the driver drowsiness levels have been established, the system continually monitor the levels and in the event they hit a certain level currently set at 50% the system will trigger a warning on the display.
* If the driver notices and takes corrective action, the percentage drops and the warning disappears.
* If the driver continues to be drowsy and the percentage continues to rise for a period above 30 seconds the system treats this as a micro-sleep and an audio alert is generated by the system to warn the driver he is asleep.

#### Experimental Protocol

The system proposed is built on Linux Operating system and the detection mechanism is carried out with the help of OpenCV Library. Windows is an interface between computer/server hardware, and the programs which run on it. The most obvious advantage of using Linux is that it is free to obtain. A windows distribution can be installed on any number of computers free of cost. In line with the costs, the security aspect of windows is much stronger than that of other OS. Hence, no extra money has to be spent on virus protection software. Open source Computer Vision Library (OpenCV) is a library of programming functions that is exclusively used for applications based on computer vision. It has python interfaces and supports windows easily. OpenCV is quick when it comes to speed of execution. OpenCV programs only require around 70MB of RAM to run in realtime. Any device that can run C, can, in all probability, run OpenCV. The detection algorithm is

implemented using features of the Haar Classifiers for object detection. The core basis for Haar classifier object detection is the Haar-like features. These features use the change in contrast values between adjacent rectangular groups of pixels instead of the intensity values of a pixel. The entire system for detecting drowsiness is implemented using these Haar Classifiers.

The different phases of the algorithm are driven by:

* Face Detection
* Eye Detection
* Eye Closure Characterization
* Yawn Detection

#### ALGORITHM

The different phases involved in detecting drowsiness are as described below.

#### Smoothening the input image

The live feed is continuously fed into the camera placed in front of the driver. The video that is captured is then converted into frames. The illumination changes in the input should be corrected.

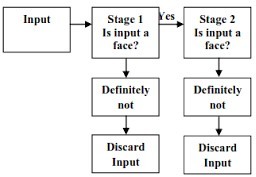
This is done by smoothening the image. An average filter is used for the same.

It takes the average of all neighboring pixels and sum is computed. This is then divided by the number of pixels in the window. One such window can be seen in Fig.

Where, (p) is the filtered pixel. The average filter thus smoothens out the input images. The filter used is of 3\*3 size. The average filter lessens the changes in illumination from the images drawn.

#### Face Detection

The face is detected using Viola Jones based Haar Classifiers. The classifier is trained for a set of positive and negative images. The training algorithm includes the region of interest, here; the face as positive samples of images and the negative images become the images that do not comprise face. A sample of pixel values is generated by an input of numerous face images. Thus, the face is detected herewith. OpenCV thus acts as the trainer and the associated pixel values characterize an input feature as a face or a nonface. If it is detected as a face, it is carried forward to detect the eyes, otherwise it is discarded from any further detection mechanism. This identification is done with the help of cascading a number of classifiers.



**Fig4.5.2. Cascade Classifiers to detect face**

#### Eye Detection

Once the cascade classifiers detect and pass an image detected as face, the eyes are searched for. Using the Haar Classifiers, feature points are analyzed and pixel values are determined from the passed image. A sample of Haar feature types are passed and the filter gradient calculates the pixel values of the region of interest which are the eyes here. Some of the Haar types used in searching and locating the eyes are shown in Fig.

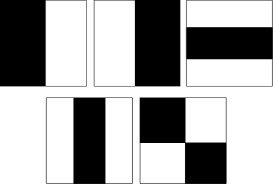


Fig4.5.3. Haar Types used to detect eyes

#### Eye Closure Evaluation

The eye related parameters used to detect drowsiness in the proposed system are namely eyelid closure, gaze detection and blink rate monitoring. A square wave has been used to recognize the different states of the eye. A logical 1 and 0 are respectively used to signify input of frames and otherwise. The following block diagram briefly depicts the algorithm involved.

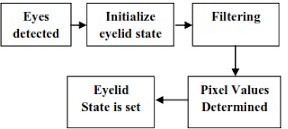


Fig 4.5.4. Eyelid State Detection

#### UML Diagrams

Unified Modelling Language (UML) is a generalmodelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programminglanguage,it is rather a visual language. We use UML diagrams to portray the **behavior and structure** of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

#### Usecase Diagram

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, associations between actors and users. In general, it shows a set of use cases and actors and their relationships. The creation of a use case model is an excellent vehicle for elicitation of functional requirements.Existing user(driver) and New user(driver) are acts as a Actors in the above A usecase diagram. Detect Drowsiness state, Detect facial expression, Detect physiological parameters and Audio Alert Message are act as usecases, Which are represented in the ellipse shape in the above UML diagram. The Existing user and New user has communicated with all these usecases.

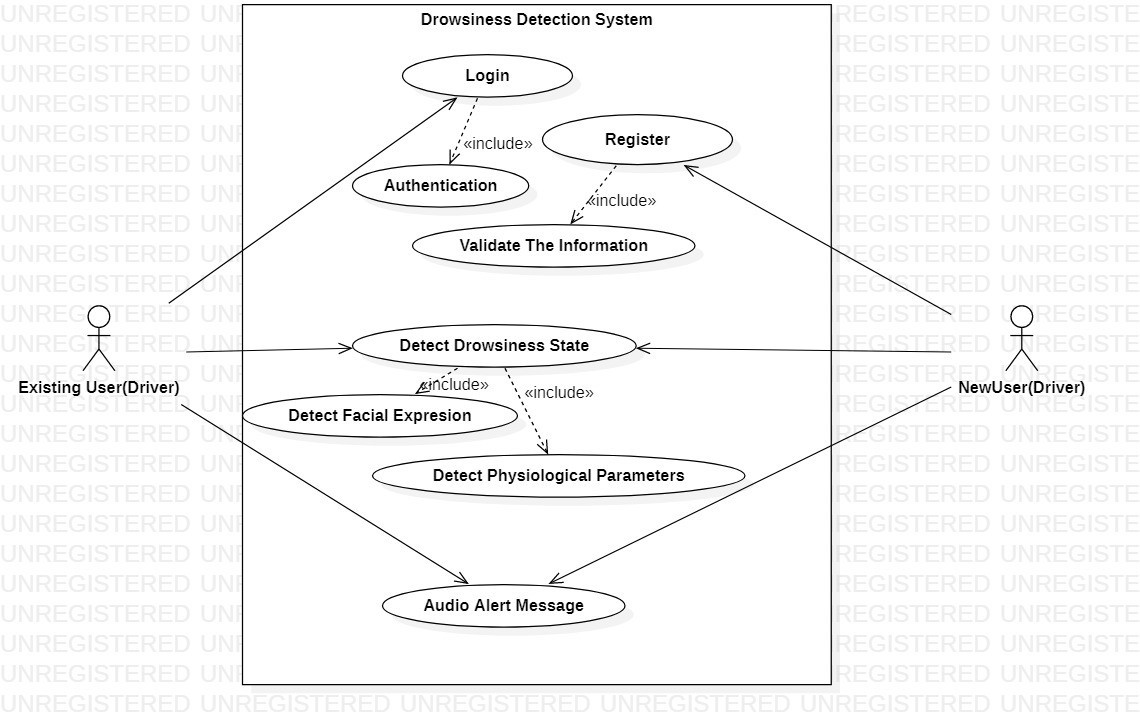


Fig 4.6.1: Usecase diagram for driver drowsiness detection system

#### Sequence Diagram

A sequence diagram in unified modelling language(UML) is a kind of interaction diagram that shows how processes with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. For instance, the UML diagram on the right describes the sequences of messages of a Driver Drowsiness Detection System.Sequence diagram are an easy and intuitive way of describing the behavior of a system by the interaction between the system and its environment. A sequence diagram has two components are vertical dimension represents time, the horizontal dimension represents different object. The vertical line is called object’s lifeline.

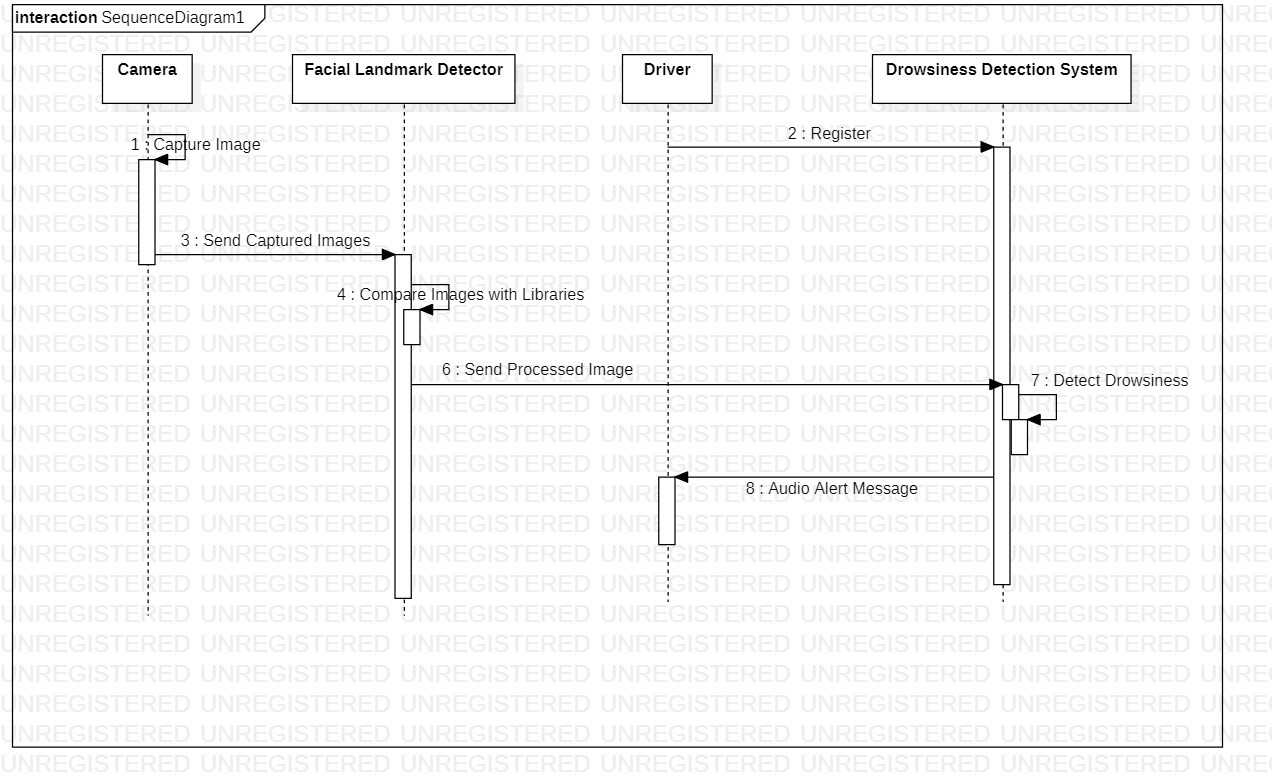


Fig 4.6.2: Sequence diagram for driver drowsiness detection system

In the above sequence diagram, camera, Facial Landmark Detector, Driver, Drowsiness Detection System are the objects and their functionalities are represented through the stimulus from the selected object to target object.

#### Activity Diagram

The purpose of activity diagram is to provide a view of flows and what is going on inside ause case or among several classes. An activity is shown as around box containing the nameof the operation.In the activity diagram. Take the input video and separate the frames for Detection of Eye.Camera capture the blinking of eyes then Drowsiness Alert Message in audio.Based upon the conditions of activities are connected.We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

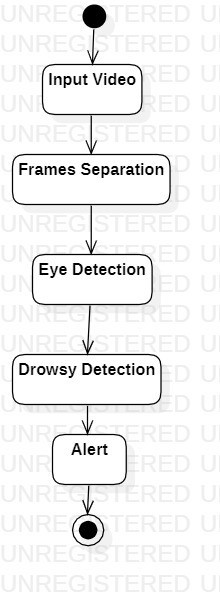


Fig 4.6.3: Activity Diagram for Driver Drowsiness Detection System

#### Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships

between those elements. Several vendors offer software for creating and editing collaboration diagrams.

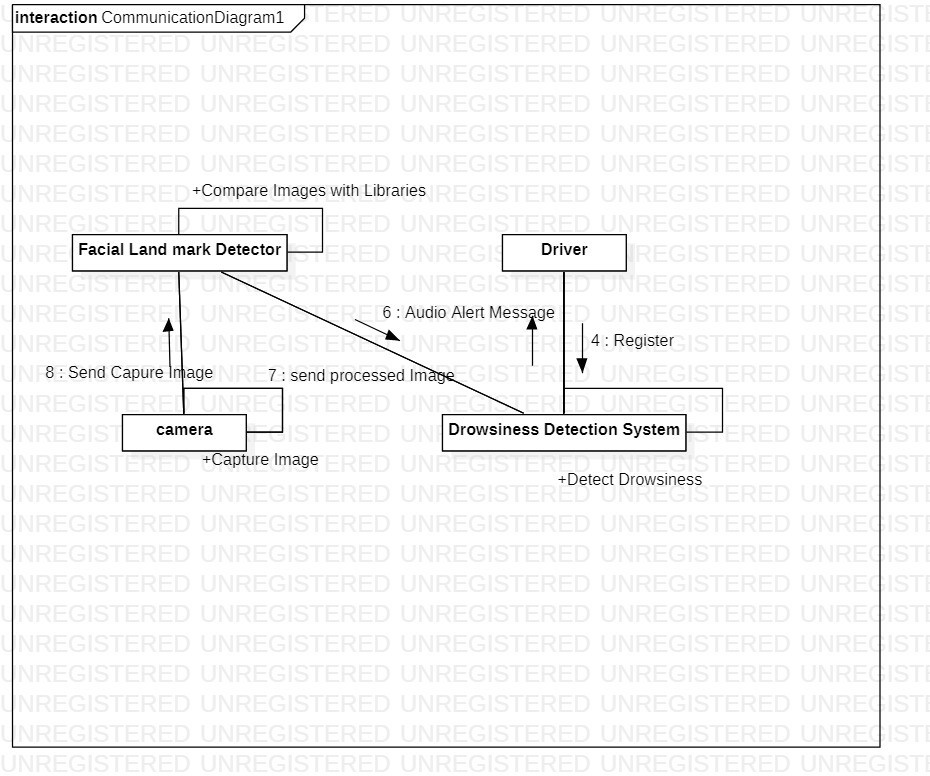


Fig 4.6.4: Collaboration Diagram for Driver Drowsiness Detection System

In the above collaboration diagram. Actions are represented by the Nodes. Camera, Facial Landmark Detector, Driver, Drowsiness Detection System are the nodes which are connected to one another based on their functionalities.

#### Classification Diagram

**CHAPTER 5 CLASSIFICATION**

The function of the system can be broadly divided into eye detection function, comprising the first half of the preprocessing routine, and a drowsiness detection function, comprising the second half.

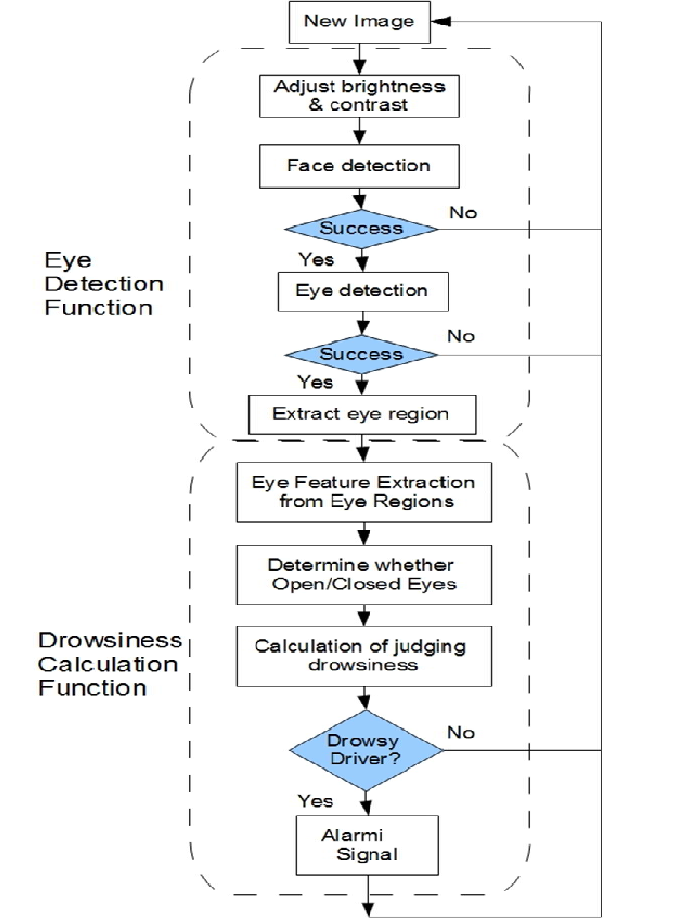


Fig 5.1: Classification Diagram For DDDS

After inputting a facial image, preprocessing is performed to binarize the image and remove noise, which makes it possible for the image to be accepted by the image processor. The maximum width of the face is then detected so that the right and left edges of the face can be identified. After that the vertical position of each eye is detected independently within an area defined by the center line of the face width and lines running through the outermost points of the face. On that basis, the area in which each eye is present is determined. Once the areas of eye presence have been defined, they can be updated by tracking the movement of the eyes. The degree of eye openness is output simultaneously with the establishment or updating of the areas of eye presence. That value is used in judging whether the eyes are open or closed and also in judging whether the eyes have been detected correctly or not. If the system judges that the eyes have not been detected correctly, the routine returns to the detection of the entire face.

The following explains the eye detection procedure in the order of the processing operations.

#### Preprocessing

The preprocessing operations include the binarization of a facial image to increase the processing speed and conserve memory capacity and noise removal. The image processor developed for this drowsiness warning system performs the expansion and contraction operation on the white pixels and processing for noise removal is performed on the small black pixels of the facial images. After the binarization, the noise removal procedure involves an expansion processing method combined with the use of a median filter. These preprocessing operations are sufficient to support detection of the vertical positions of the eyes. However, following identification of the eye positions, the size of the eyes must be converted back to the original image format at he time the degree of eye openness is output. To facilitate that, data contraction is performed in the latter stage of preprocessing.

#### Face width detection

The maximum width of the driver’s face must be detected in order to determine the lateral positions of the areas in which the eyes are present. Face width is detected by judging the continuity of white pixels and the pattern of change in pixel number. On that basis, the outer edges of the face are recognized and determined.

#### Detection of vertical eye positions

Each vertical eye position is detected independently within an area demarcated by the center line of the face, which is found from the face width, and straight lines running through the right and left outer edges of the face. In a binary image, the eye becomes collection of black pixels, along with the eyebrows, nostrils, mouth and other facial features. Thesecollections of black pixels are recognized on the basis of a labeling operation, and the position of each eye is extracted by judging the area of each label along with its aspect ratio and relative coordinate positions in the facial image. Through this process of detecting each vertical eye position, the central coordinates of each eye are recognized. The coordinates serve as references for defining the areas of eye presence.

#### Eye tracking

A function for tracking the positions of the eye is an important capability for achieving high-speed processing because it eliminates the need to process every frame in order to detect each eye position from the entire facial image. This function consists of a subordinate for updating the areas of eye presence and recognizing when tracking becomes impossible. The basic concept of eye tracking is to update the area of eye presence, in which an eye search is made in the following frame, according to the central coordinates of the eye in the previous frame. The updating process involves defining an arc of eye presence on the basis of the coordinates (xk, yk) at the point of intersection of center lines running through the Feret’s diameter of the detected eye. The area thus becomes the area of eye presence in which the system searches for the eye in the image data of the next frame.

This process of using information on eye position to define the eye position for obtaining the next facial image data makes it possible to track the position of the eye. As it is clear from this description, the size of the area of eye position changes. If the eyes are tracked correctly, their degree of openness will always vary within certain specified range for each individual driver. Consequently, if the value found by the system falls outside the range, it judges that the eyes are not being tracked correctly. The process of detecting the position of each eye from the entire facial image is then executed once more.

#### Judgment whether the eye are open/closed

We constructed a template consisting of two circles, one inside the other. A good match would result in many dark pixels in the area inside the inner circle, and many bright pixels in the area between the two circles. This match occurs when the inner circle is centered on the iris and the outside circle covers the sclera.

#### Classifier

A classifier is any algorithm that sorts data into labeled classes, or categories of information. A simple practical example are spam filters that scan incoming “raw” emails and classify them as either “spam” or “not-spam.” Classifiers are a concrete implementation of pattern recognition in many forms of machine learning. Classifiers are where high-end machine theory meets practical application. These algorithms are more than a simple sorting device to organize, or “map” unlabeled data instances into discrete classes. Classifiers have a specific set of dynamic rules, which includes an interpretation procedure to handle vague or unknown values, all tailored to the type of inputs being examined. Most classifiers also employ probability estimates that allow end users to manipulate data classification with utility functions.

#### 5.2.1 Haar Cascade Classifier

Object Detection using Haar feature-based cascade classifiers is an effective object detection method 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 is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

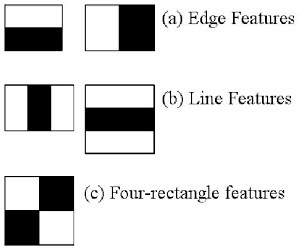


Fig 5.2.1:Haar cascade Classifiers

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by **Adaboost**.

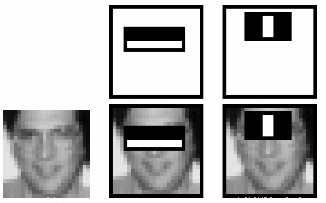


Fig 5.2.2: Adaboost diagram

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. The authors have a good solution for that.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. How is that plan!

The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub- window.

So this is a simple intuitive explanation of how Viola-Jones face detection works. Read the paper for more details or check out the references in the Additional Resources section.

**` CHAPTER 6**

**REQUIREMENTS**

* 1. **Software Requirements** Python 2.7 or above versions Anaconda software

#### Hardware Requirements

Camera

Personal computer

### CHAPTER 7

**IMPLEMENTATION**

###### Introduction

The implementation of the drowsiness detection system with its features and its installation and setup procedure are also described. Mid portion of the chapter described how the entire process of drowsiness detection occurs in low level. For conducting this libraries of OpenCv is used. Different .xml files of OpenCv is operated on the input and provide the required result. The .xml files written for drowsiness detection includes face and eye detection which basically done by algorithm developed by Viola-Jones. Those algorithm includes Haar features, Formation of integral Image, Adaboost and Cascading. Theoretical part of all those features are described briefly.

* 1. **Implementation**

They are four stages:

* + 1. Haar Feature Selection
    2. Creating an Integral Image
    3. Adaboost Training
    4. Cascading Classifiers 5 Classification

#### Haar features

Haar-like features are digital image features used in object detection. Or we can say that these are rectangle shaped dark and light areas having similar kind of features like our face. So basically we move those features throughout our face to find the output of each feature.

For example:

All faces share some similar properties

1. The eyes region is darker than the upper-cheeks.
2. The nose bridge region is brighter than the eyes.

So this features of face are used for developing haar like features. Each feature is related to a special location in the face.

Output of Rectangle features:

We will move the related kind of rectangle throughout the face to get different values.

1. Value = ∑ (pixels in black area) - ∑ (pixels in white area).
2. Three types: two-, three-, four-rectangles, Viola and Jones used two-rectangle features.

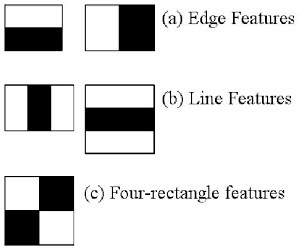


Fig7.2.1 : Different features used for Haar cascade

#### Integral Image

We know each point of an image is represented by a pixel value. As we so we need to know the output of applied Haar features so we need to find the sum of pixel value of all those area and solve the summation. But this is a huge task. To reduce the number of calculation concept of INTEGRAL IMAGE is introduced.

#### Definition of Integral Image:

Basically Integral image is a matrix same as size of the window. The integral image at location (x, y) is the sum of the pixels above and to the left of (x, y).

For example :

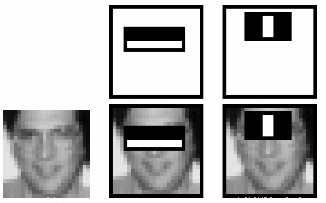


Fig 7.2.2: Integral images

#### Adaboost

which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier. 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

#### Cascade Classifier

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called decision stumps. Each stage is trained using a technique called boosting. Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners

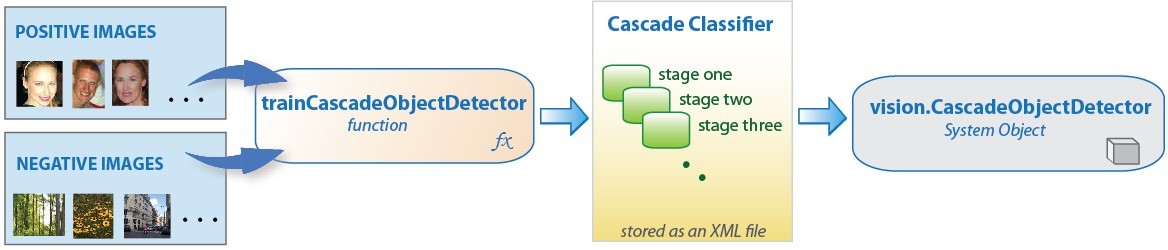


Fig 7.2.4: Cascade Classifier

#### Modification in the algorithm

* + 1. **Modification in face detection**

For detection of face, loading of cascade file is done. Then the requirement is to pass on the captured frame to a function which does detection of edge. After this process almost all the possible type objects may get detected corresponding to different sizes. Hence the task of reduction of reduction of processing amount comes here. To achieve this instead of Detecting all the objects present in the frame because we know that our required object is face and in most case it occupies almost all the portion of the captured frame. So we can modify the algorithm to detect only in that fashion.

#### Modification in eye detection

For eye detection as the amount of processing will be very high if we apply the features to all the portion of the face. Hence to avoid this situation we should be interested only in those portion of face where we are certain that eye exist there. We set our work region for finding the eye by taking the following facts.

* + - 1. Only upper parts of the face contain the maximum probability of finding an eye.
      2. The place of occurrence of eyes are a few pixel below the fore head.

#### Modification in colour selection

Now for drowsiness detection instead of using the colored image, the image is converted to gray scale to reduce the no of channel parameters which helps to increase the speed of calculation of the classifiers.

###### Deep Learning Classification

Deep learning is an artificial intelligence (AI) function that imitates the workingsof the human brain in processing data and creating patterns for use in decision making.Deep learning is a subset of machine learning in artificial intelligence that has networkscapable of learning unsupervised from data that is unstructured or unlabeled. Also knownasdeep neural learning or deep neural network.

Itisaformofmachinelearning,withfunctionsthatoperateinanonlineardecision-making process. Deep learning occurs when decisions are made on unstructureddata without supervision. Object recognition, speech recognition, and language translationaresomeof thetasks performed through deep learning.

Deep learning is an AI function that mimics the workings of the human brain inprocessingdataforuseindetectingobjects,recognizingspeech,translatinglanguages,andmak ing decisions.

Deep learning AI is able to learn without human supervision, drawing from

datathat is both unstructured and unlabelled. It is a form of machine learning, can be used tohelpdetect fraud ormoney laundering,among other functions.

# NeuralNetworks

**Neuralnetworks**areartificialsystemsthat wereinspiredby biologicalneural networks.These systems learn to perform tasks by being exposed to various datasets and exampleswithout any task-specific rules. The idea is that the system generates identifyingcharacteristics from the data they have been passed without being programmed with a pre- programmedunderstanding ofthesedatasets.

Components of a typical neural network involve neurons, connections, weights, biases,propagation function, and a learning rule. Neurons will receive an input from predecessorneurons that have an activation, threshold, an activation function f, and an outputfunction.Connectionsconsistofconnections,weightsandbiaseswhichruleshowneuron i transfers output to neuron j. Propagation computes the input and outputs the output andsums the predecessor neurons function with the weight. The learning rule modifies theweightsand thresholds of thevariables in thenetwork.

Artificial Neural Networks contain artificial neurons which are called **units**. These unitsare arranged in a series of layers that together constitute the whole Artificial NeuralNetworks in a system. A layer can have only a dozen units or millions of units as thisdepends on the complexity of the system. Commonly, Artificial Neural Network has aninput layer, output layer as well as hidden layers. The input layer receives data from theoutside world which the neural network needs to analyze or learn about. Then this datapasses through one or multiple hidden layers that transform the input into data that isvaluable for the output layer. Finally, the output layer provides an output in the form of aresponseoftheArtificialNeuralNetworks to input dataprovided.

In the majority of neural networks, units are interconnected from one layer to another.Eachoftheseconnectionshas weightsthat determinetheinfluenceof oneunit onanotherunit. As the data transfers from one unit to another, the neural network learns more andmoreabout thedata which eventually resultsin anoutputfrom theoutputlayer.

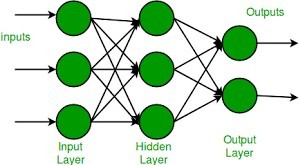


Fig.3.3.4.1.Neural Networks

Types:

1. ***FeedforwardNeural Network***

The feedforward neural network is one of the most basic artificial neural networks. In thisANN, the data or the input provided ravels in a single direction. It enters into the ANNthroughtheinputlayerandexitsthroughtheoutputlayerwhilehiddenlayersmayormaynot exist. So the feedforward neural network has a front propagated wave only andusuallydoes not havebackpropagation.

1. ***RecurrentNeuralNetwork***

The Recurrent Neural Network saves the output of a layer and feeds this output back tothe input to better predict the outcome of the layer. The first layer in the RNN is quitesimilar to the feed-forward neural network and the recurrent neural network starts oncethe output of the first layer is computed. After this layer, each unit will remember someinformation from the previous step so that it can act as a memory cell in performingcomputations.

1. ***ConvolutionalNeuralNetwork***

A Convolutional neural network has some similarities to the feed-forward neural network,where theconnectionsbetweenunitshaveweightsthatdeterminetheinfluenceofoneunit on another unit. But a CNN has one or more than one convolutional layers that use aconvolution operation on the input and then pass the result obtained in the form of outputto the next layer. CNN has applications in speech and image processing which isparticularlyuseful in computer vision.

1. ***ModularNeuralNetwork***

A Modular Neural Network contains a collection of different neural networks that workindependently towards obtaining the output with no interaction between them. Each of thedifferent neural networks performs a different sub-task by obtaining unique inputscompared to other networks. The advantage of this modular neural network is that itbreaks down a large and complex computational process into smaller components, thusdecreasingits complexity while still obtainingtherequired output.

1. ***RadialbasisfunctionNeuralNetwork***

Radial basis functions are those functions that consider the distance of a point concerningthe center. RBF functions have two layers. In the first layer, the input is mapped into allthe Radial basis functions in the hidden layer and then the output layer computes theoutput in the next step. Radial basis function nets are normally used to model the data thatrepresentsany underlying trend orfunction.

# Python

Python is an interpreted, object-oriented, high-level programming language with dynamicsemantics. It’s high-level built in data structures, combined with dynamic typing anddynamic binding, make it very attractive for Rapid Application Development, as well asforuseasascriptingorgluelanguagetoconnectexistingcomponentstogether.Python's

simple, easy to learn syntax emphasizes readability and therefore reduces the cost ofprogrammaintenance.Pythonsupportsmodulesandpackages,whichencouragesprogram modularity and code reuse. The Python interpreter and the extensive standardlibrary are available in source or binary form without charge for all major platforms, andcanbefreely distributed.

Since there is no compilation step, the edit-test-debug cycle is incredibly fast. DebuggingPython programs is easy: a bug or badinputwill never cause a segmentation fault.Instead, when the interpreter discovers an error, it raises an exception. When the programdoesn't catch the exception, the interpreter prints a stack trace. A source level debuggerallows inspection of local and global variables, evaluation of arbitrary expressions, settingbreakpoints, stepping through the code a line at a time, and so on. The debugger is writtenin Python itself, testifying to Python's introspective power. On the other hand, often thequickest way to debug a program is to add a few print statements to the source: the fastedit-test-debug cycle makes this simple approach very effective. It’s easy to learn syntaxandportability capability makes it popularthesedays.

Thefollowingsfactsgive us theintroductiontoPython:

1. PythonwasdevelopedbyGuidovanRossumatStichtingMathematischCentrumintheNethe rlands.
2. Itwaswrittenasthesuccessorofprogramminglanguagenamed‘ABC’.3.It’sf irstversionwasreleased in 1991.
3. ThenamePythonwaspickedbyGuidovanRossumfromaTVshownamedMontyPython’sFlyi ng Circus.
4. Itisanopensourceprogramminglanguagewhichmeansthatwecanfreelydownloaditandus eit to developprograms.It canbedownloaded from[www.python.org.](http://www.python.org/)
5. Python programming language is having the features of Java and C both. It is having theelegant ‘C’ code and on the other hand, it is having classes and objects like Java forobject-orientedprogramming.
6. It is an interpreted language, which means the source code of Python program would befirstconverted into bytecodeand then executed by Python virtual machine.

***PythonforDataScience***

Whether you’re doing straightforward data analysis or full-on data science, you’d be hard- pressedtofindabettersuiteoftoolsthanthoseinPython.ThePandaslibraryisaquantum-leap improvement over the dusty Excel spreadsheets in which financial analysiswas done for so long. If Pandas isn’t fast enough for you, most of the basic vectoroperations can be one with NumPy. Numpy also offers the ability to do linear algebra,scientific computing, and a host of other highly technical things. It is, therefore, a greattool to learn how to use well. Python is the fifth most important language as well as mostpopular language for Machine learning and data science. The following are the features ofPythonthat makes itthe preferredchoiceoflanguagefordata science:

1. Extensivesetofpackages2. Easy prototyping3.Collaboration

feature 4.Onelanguageformanydomains

***ArtificialIntelligenceandMachineLearning***

Pythoncommunityhasdevelopedmanymodulestohelpprogrammersimplementmachine learning. Artificial intelligence and machine learning have become buzzwordsthese days, but the truth is that it all comes down to algorithms, code, and logic. Given thescope and power of Python, it’s no surprise that some truly world-class tools exist forgenerating intelligent behavior in Python. Arguably the most popular is the ubiquitousmachine learning library, Scikit-Learn. Speaking from experience, Sklearn makes theprocess of building everything from classifiers to regression models orders of magnitudesimpler than it otherwise would be.If neural networks are more your jam, there’s alwaysTensorFlow. Adding in the new Keras API, building a state-of-the-art neural network iseasierthan it hasever been.

\*

**ConvolutionalNeuralNetwork**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithmwhich can take in an input image, assign importance (learnable weights and biases) tovarious aspects/objects in the image and be able to differentiate one from the other. Thepre-processing required in a ConvNet is much lower as compared to other classificationalgorithms. While in primitive methods filters are hand-engineered, with enough training,ConvNetshavethe ability to learn thesefilters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern ofNeurons in the Human Brain and was inspired by the organization of the Visual Cortex.Individual neurons respond to stimuli only in a restricted region of the visual field knownastheReceptive Field.A collectionofsuchfieldsoverlaptocovertheentirevisualarea.

AConvNetisableto successfullycapturetheSpatialandTemporaldependencies in an image through the application of relevant filters. The architectureperforms a better fitting to the image dataset due to the reduction in the number ofparameters involved and reusability of weights. In other words, the network can be trainedtounderstandthesophisticationoftheimagebetter.

The term “Convolution” in CNN denotes the mathematical function of convolution whichisaspecialkindoflinearoperationwhereintwofunctionsaremultipliedtoproduceathird function which expresses how the shape of one function is modified by the other. Insimple terms, two images which can be represented as matrices are multiplied to give anoutputthat is used to extract features from the image.

TherearetwomainpartstoCNNarchitecture:

* + Aconvolutiontoolthatseparatesandidentifiesthevariousfeaturesoftheimageforanal ysis in a processcalledasFeatureExtraction
  + Afullyconnectedlayerthatutilizestheoutputfromtheconvolutionprocessandpredict stheclass oftheimagebased onthefeaturesextractedin previousstages.

There are three types of layers that make up the CNN which are the convolutional layers,pooling layers, and fully-connected (FC) layers. When these layers are stacked, CNNarchitecture will be formed. In addition to these three layers, there are two more importantparameters which are the dropout layer and the activation function which are definedbelow.

***ConvolutionalLayer***

Convolutional layers are the major building blocks used in convolutionalneural networks. This layer is the first layer thatis used to extract the various featuresfromtheinputimages.Inthislayer,themathematicaloperationofconvolutionisperform ed between the input image and a filter of a particular size MxM. By sliding thefilter over the input image, the dot product is taken between the filter and the parts of theinput image with respect to the size of the filter (MxM). If the filter is designed to detect aspecific type of feature in the input, then the application of that filter systematically acrossthe entire input image allows the filter an opportunity to discover that feature anywhere intheimage.

The output is termed as the Feature map which gives us information about theimage such as the corners and edges. Later, this feature map is fed to other layers to learnseveralother features of theinput image.

***TheConvolutionOperation***

Weanalyzetheinfluenceofnearbypixelsbyusingsomethingcalledafilter.Wemovethisacross the image from top left to bottom right. When building the network, we randomlyspecify values for the filters, which then continuously update themselves as the network istrained. After the filters have passed over the image, a feature map is generated for eachfilter.Thesearethentakenthroughanactivationfunction,whichdecideswhetheracertainfea tureispresentat agiven locationintheimage.

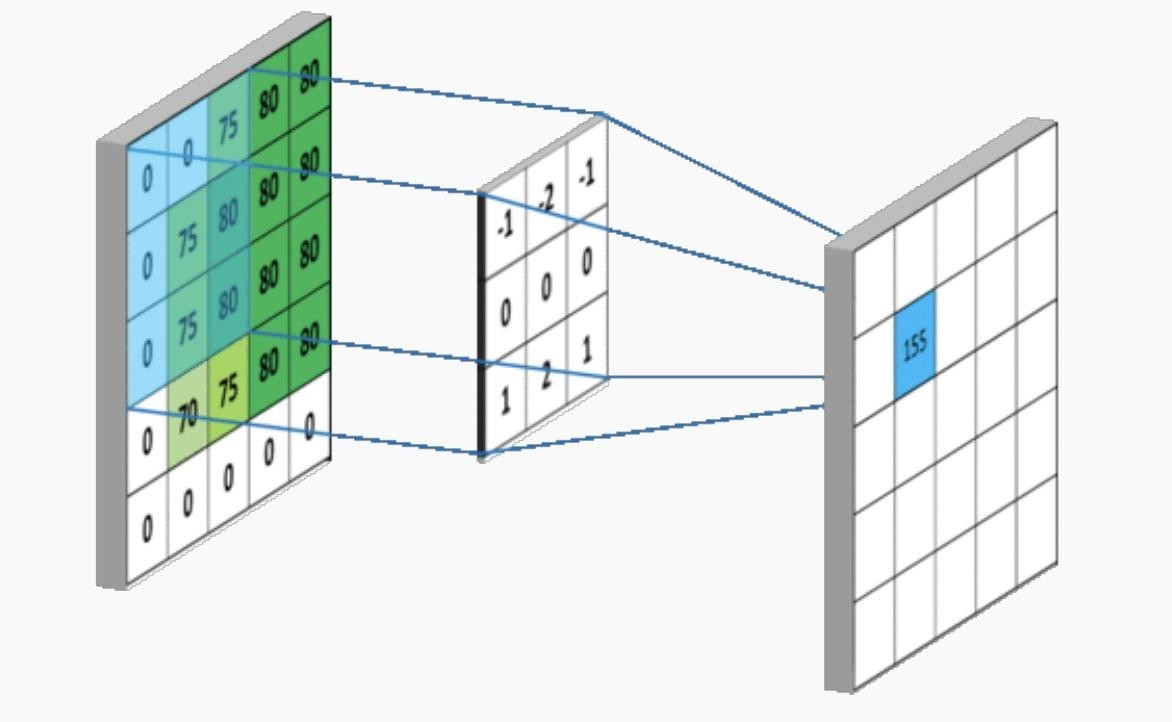


Fig.3.3.7.1.ConvolutionOperation

We have colour images which have 3 channels i.e., Red, Green and Blue. Each filter isconvolved with the entirety of the 3D input cube but generates a 2D feature map. Becausewe have multiple filters, we end up with a 3D output: one 2D feature map per filter.Convolvingtheimagewithafilterproducesafeaturemapthathighlightsthepresenceofagiv en featurein theimage.

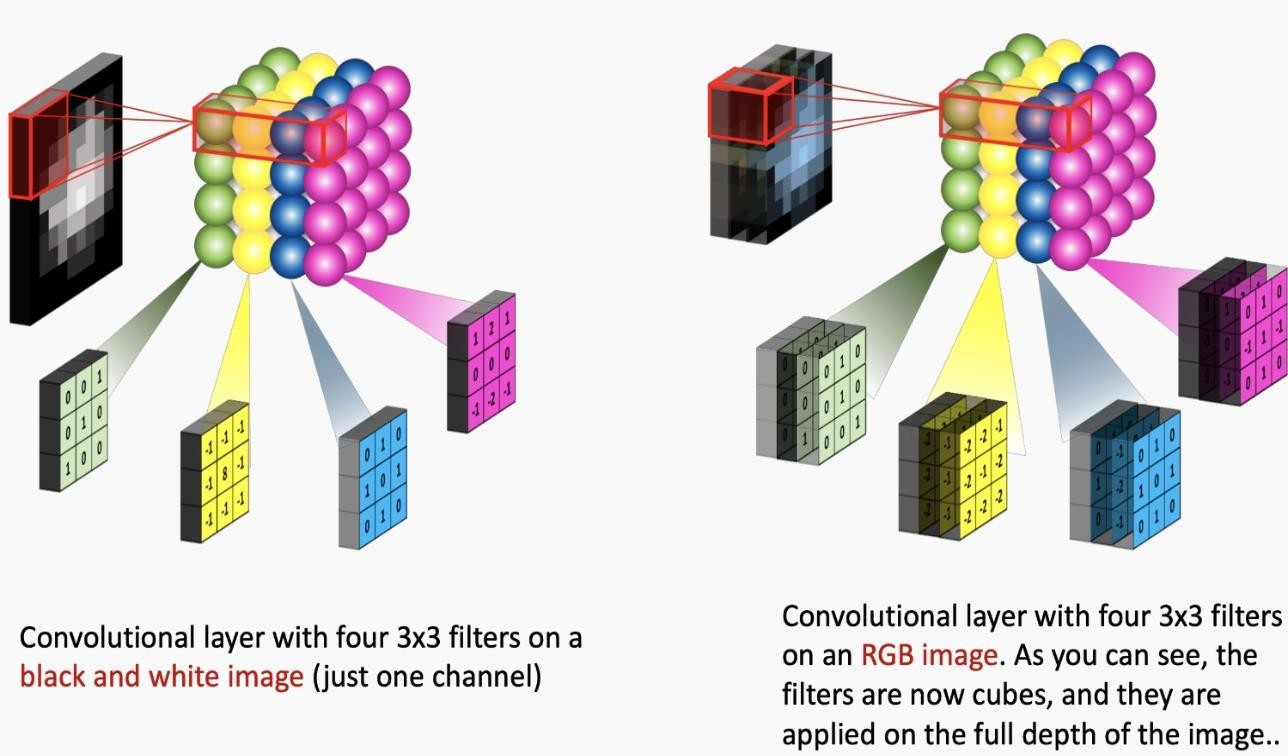


Fig..Convolution withRGBimages

***Stride***

Stride is the number of pixels shifts over the input matrix. The filter moves to the rightwith a certain Stride Value till it parses the complete width. Moving on, it hops down tothe beginning (left) of the image with the same Stride Value and repeats the process untilthe entire image is traversed. When the stride is 1 then we move the filters to 1 pixel at atime. When the stride is 2 then we move the filters to 2 pixels at a time and so on. Thebelowfigureshowsconvolutionwouldwork withastrideof2.

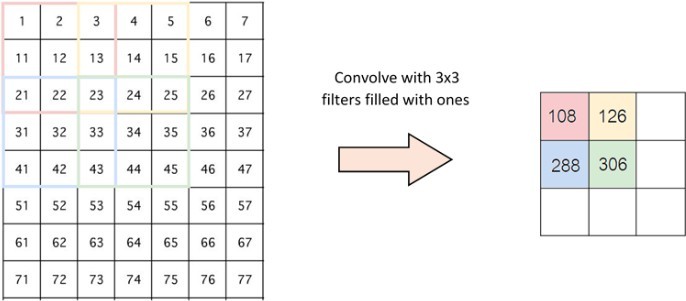


Fig..Stride

***PoolingLayer***

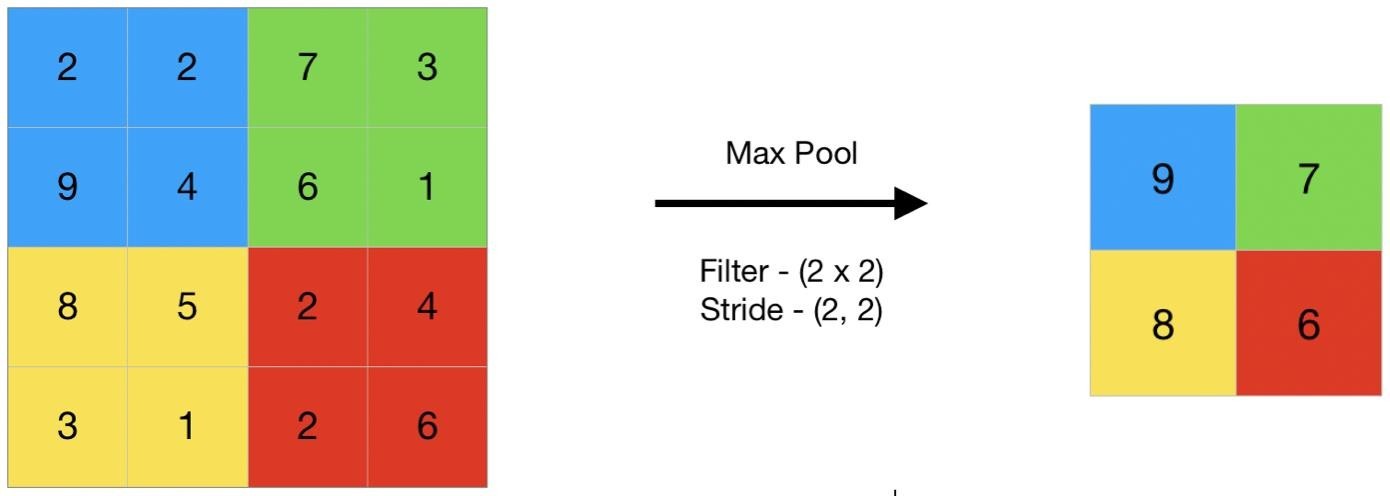
In most cases, a Convolutional Layer is followed by a Pooling Layer.

Theprimaryaimofthislayeristodecreasethesizeoftheconvolvedfeaturemaptoreducethe computational costs. This is performed by decreasing the connections between layersand independently operates on each feature map. Depending upon method used, there areseveraltypes of Pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Poolingcalculates the average of the elements in a predefined sized Image section. The total sumof the elements in the predefined section is computed in Sum Pooling. The Pooling Layerusuallyserves asabridge betweenthe ConvolutionalLayerand the FC Layer.

***MaxPooling***

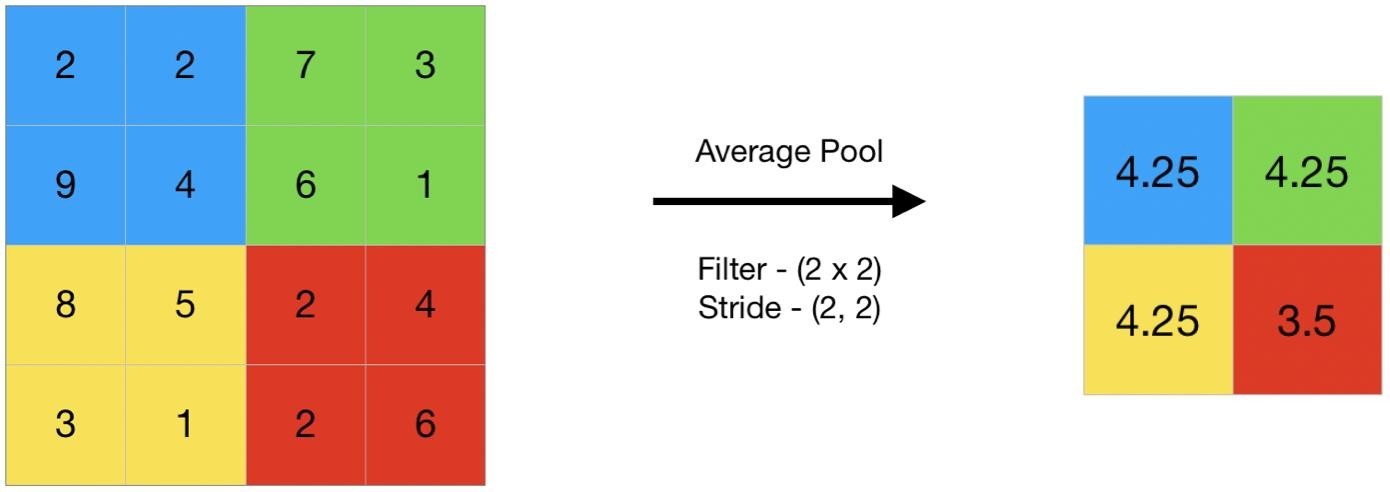
Max pooling is a pooling operation that selects the maximum element from the region ofthefeaturemapcoveredbythefilter.Thus,theoutputaftermax- poolinglayerwouldbeafeaturemapcontainingthemostprominentfeaturesofthepreviousfeatur emap.



MaxPooling

***AveragePooling***

Average pooling computes the average of the elements present in the region of featuremapcoveredbythefilter.Thus,whilemaxpoolinggivesthemostprominentfeatureina particular patch of the feature map, average poolinggives the average of featurespresentinapatch.



AveragePooling

***GlobalPooling***

Global pooling reduces each channel in the feature map to a single value. Thus, an**nhxnwx nc**feature map is reduced to **1 x 1 x nc**feature map. This is equivalent to using afilter of dimensions **nh xnw** i.e.the dimensions of the feature map.Further, itcan beeitherglobalmaxpoolingorglobalaveragepooling.

***FullyConnectedlayer***

The Fully Connected (FC) layer consists of the weights and biases alongwith the neurons and is used to connect the neurons between two different layers. Theselayers are usually placed before the output layer and form the last few layers of a CNNArchitecture.

In this, the input image from the previous layers are flattened and fed to the FClayer. The flattened vector then undergoes few more FC layers where the mathematicalfunctions operations usually take place. In this stage, the classification process begins totakeplace.

***Dropout***

When all the features are connected to the FC layer, it can cause overfittingin the training dataset. Overfitting occurs when a particular model works so well on thetraining data causing a negative impact in the model’s performance when used on a newdata.

To overcome this problem, a dropout layer is utilised wherein a few neurons aredropped from the neural network during training process resulting in reduced size of themodel. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from theneuralnetwork.

# BackPropagation

Backpropagation is an algorithm commonly used to train neural networks. When theneural network is initialized, weights are set for its individual elements, called neurons.Inputs are loaded, they are passed through the network of neurons, and the networkprovides an output for each one, given the initial weights. Backpropagation helps to adjustthe weights of the neurons so that the result comes closer and closer to the known trueresult.

Backpropagation,shortfor"backwardpropagationoferrors,"isanalgorithmforsupervised learning of artificial neural networks using gradient descent. Given an artificialneural network and an error function, the method calculates the gradient of the errorfunction with respect to the neural network's weights. It is ageneralization of the deltarule for perceptron’s to multilayer feedforward neural networks. The "backwards" part ofthe name stems from the fact that calculation of the gradient proceeds backwards throughthe network, with the gradient of the final layer of weights being calculated first and thegradient of the first layer of weights being calculated last. Partial computations of thegradient from one layer are reused in the computation of the gradient for the previouslayer.Thisbackwardsflowoftheerrorinformationallowsforefficientcomputationofth e gradient at each layer versus the naive approach of calculating the gradient of eachlayerseparately.

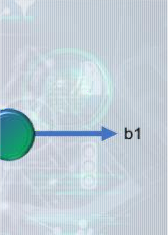
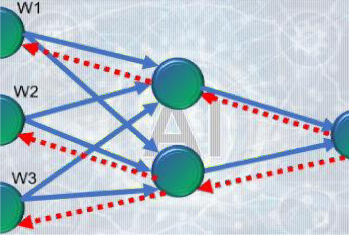


Fig..BackPropagation

Backpropagation's popularity has experienced a recent resurgence given the

widespreadadoption recognition.

of deep neural networks for image recognition and speech

It

isconsideredanefficientalgorithm,andmodernimplementationstakeadvantageofspecialized GPUs to further improveperformance.

# ActivationFunctions

OneofthemostimportantparametersoftheNeuralNetworksistheactivation function. They are used to learn and approximate any kind of continuous andcomplex

relationship between

variables of the network. In simple

words, it decides

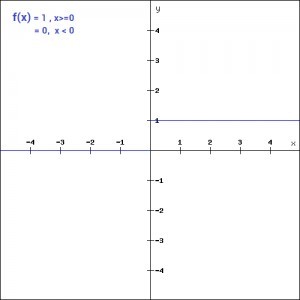
whichinformation of the model should fire in the forward direction and which ones should not attheendofthenetwork.Activationfunctiondecides,whetheraneuronshouldbeactivatedornotb ycalculatingweightedsumandfurtheraddingbiaswithit.

It adds non-linearity to the network. There are several commonly used activationfunctions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of thesefunctionshasaspecificusage.Fora binaryclassificationCNN model,sigmoidandsoftmaxfunctionsarepreferredforamulti-class classification,generallysoftmaxusused.

SomeoftheActivationFunctionsare:

***BinaryStepFunction***

If the input to the activation function is greater than a threshold, then the neuron isactivated,else it is deactivated.

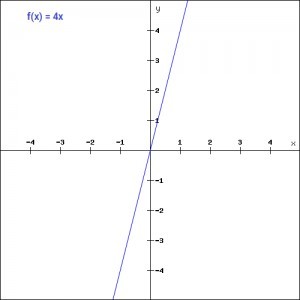


BinaryStepFunction

***LinearFunction***

Heretheactivationisproportionaltotheinput.Thevariable‘a’inthiscasecanbeanyconstantvalue

.



Linear Function

***Sigmoid***

Itisoneofthemostwidelyusednon- linearactivationfunction.Sigmoidtransformsthevaluesbetween the range0 and 1.

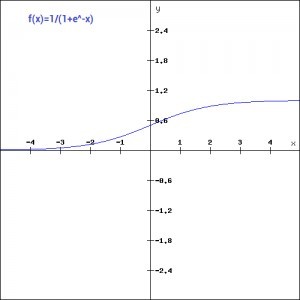


Fig.SigmoidFunction

***Tanh***

Thetanhfunctionisverysimilartothesigmoidfunction.Theonlydifferenceisthatitissymmetrica roundthe origin. Therangeof values inthis caseis from-1 to 1.

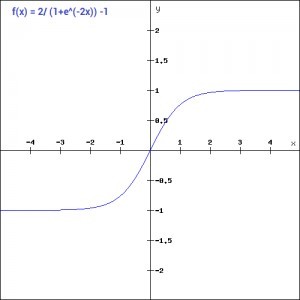


Fig..TanhFunction

***ReLU***

The ReLU function is another non-linear activation function that has gained popularity inthe deep learning domain. ReLU stands for Rectified Linear Unit. The neurons will onlybedeactivated ifthe output of the linear transformation is less than 0.

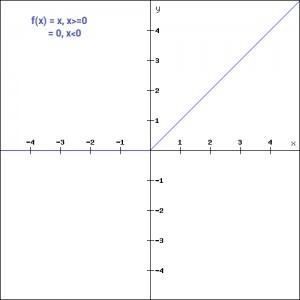


Fig. ReLUFunction

***Softmax***

Softmax function is often described as a combination of multiple sigmoids. We know thatsigmoid returns values between 0 and 1, which can be treated as probabilities of a datapoint belonging to a particular class. The softmax function can be used for multiclassclassification problems. This function returns the probability for a datapoint belonging toeachindividual class.



# Training

Deep learning neural networks learn a mapping function from inputs to outputs. This isachieved by updating the weights of the network in response to the errors the modelmakes on the training dataset. Updates are made to continually reduce this error untileitheragood enoughmodel is foundorthe learning processgets stuckandstops.

The stochastic gradient descent algorithm is used to solve the optimization problem wheremodelparameters areupdated each iterationusing the backpropagationalgorithm.

A neural network model uses the examples to learn how to map specific sets of inputvariablestotheoutputvariable.

Itmustdothisinsuchawaythatthismappingworkswellforthetrainingdataset,butalsoworkswell onnewexamplesnotseenbythe modelduring training. This ability to work well on specific examples and new examples is calledtheability of themodel to generalize.

* + **Loss Function**: The function used to estimate the performance of a model with aspecificset ofweightson examples fromthetraining dataset.
  + **WeightInitialization**:Theprocedurebywhich theinitialsmall randomvaluesareassignedto model weightsat thebeginning ofthe training process.
  + **Batch Size**:Thenumberof examplesusedto estimatethe errorgradientbeforeupdatingthe model parameters.
  + **Learning Rate**: The amount that each model parameter is updated per cycle of thelearningalgorithm.
  + **Epochs**: The number of complete passes through the training dataset before thetrainingprocess is terminated.

***TrainDataset:***

Thesampleofdatausedtofitthemodel. Thepartofdataweusetotrainourmodel.Thisisthedatawhich yourmodel actuallysees(both inputand output)and learnfrom.

# Testing

The usage of the word “testing” in relation to Machine Learning models isprimarily used for testing the model performance in terms of accuracy/precision of themodel. It can be noted that the word, "testing" means different for conventional softwaredevelopmentand Machine Learning models development.

Thegoal ofML testing:

Qualityassuranceisrequiredtomakesurethatthesoftwaresystemworksaccordingtothe requirements. Were all the features implemented as agreed? Does the program behaveas expected? All the parameters that you test the program against should be stated in thetechnicalspecification document.

Moreover, testing has the power to point out all the defects and flaws during development.We don’t want your clients to encounter bugs after it is released and come to you wavingtheir fists. Different kinds of testing allow us to catch bugs that are visible only duringruntime.

However, in machine learning, a programmer usually inputs the data and the desiredbehavior, and the logic is elaborated by the machine. This is especially true for deeplearning. Therefore, the purpose of machine learning testing is, first of all, to ensure thatthislearned logicwill remainconsistent, no matterhowmanytimes wecall theprogram.

First of all, we split the database into three non-overlapping sets. You use a training set totrainthemodel.Then, toevaluatetheperformanceof themodel, youusetwosets ofdata:

***ValidationDataset:***

Having only a training set and a testing set is not enough if you do many rounds ofhyper parameter-tuning(which isalways).Andthat canresultinoverfitting. To avoidthat, you can select a small validation data set to evaluate a model. Only after you getmaximumaccuracy onthevalidation set,you makethe testing setcomeintothe game.

***TestDataset:***

Our model might fit the training dataset perfectly well.In order to assure that itwilldoequallywellinreal-life,youselectsamplesforatestingsetfromyourtrainingset

— examples that the machine hasn’t seen before.It is important to remain unbiasedduring selection and draw samples at random. Also, you should not use the same set manytimes to avoid training on your test data. Your test set should be large enough to providestatisticallymeaningful results andberepresentativeof thedata set as awhole.

### CHAPTER 8

**CODING**

import numpy as np

import matplotlib.pyplot as plt import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

import tensorflow as tf import cv2 as cv import os

test\_path="E:\github\ML-ProjectKart\Driver Drowsiness Detection System\Dataset\New\_Test"

img = cv.imread("E:\github\ML-ProjectKart\Driver Drowsiness Detection System\Dataset\New\_Test\Closed\_Eyes\s0012\_00309\_0\_0\_0\_0\_0

\_01.png",cv.IMREAD\_GRAYSCALE)

plt.show(img)

Datadirectory= test\_path #training dataset Classes=["Closed\_Eyes","Open\_Eyes"] #list of classes for category in Classes:

path = os.path.join(Datadirectory,category) for img in os.listdir(path):

img\_array =

cv.imread(os.path.join(path,img),cv.IMREAD\_GRAYSCALE) backtorgb = cv.cvtColor(img\_array,cv.COLOR\_GRAY2RGB) plt.imshow(img\_array, cmap="gray")

plt.show() break break

img\_size=224

new\_array = cv.resize(backtorgb,(img\_size,img\_size)) plt.imshow(new\_array, cmap="gray")

plt.show()

training\_Data = []

def create\_training\_Data():

for category in Classes:

path = os.path.join(Datadirectory, category) class\_num = Classes.index(category) # 0 1, for img in os.listdir(path):

try:

img\_array = cv.imread(os.path.join(path,img), cv.IMREAD\_GRAYSCALE)

backtorgb =

cv.cvtColor(img\_array,cv.COLOR\_GRAY2RGB) new\_array = cv.resize(backtorgb, (img\_size,img\_size)) training\_Data.append([new\_array,class\_num])

except Exception as e: pass

create\_training\_Data()

print(len(training\_Data))

import random random.shuffle(training\_Data)

X = []

y = []

for features,label in training\_Data: X.append(features)

y.append(label)

X = np.array(X).reshape(-1, img\_size, img\_size, 3)

X.shape

#normalizing the data

X= X/255.0; ## normalizing data

Y=np.array(y)

import pickle

pickle\_out = open("X.pickle","wb") pickle.dump(X, pickle\_out) pickle\_out.close()

pickle\_out = open("y.pickle","wb") pickle.dump(y, pickle\_out) pickle\_out.close()

pickle\_in = open("X.pickle","rb") X = pickle.load(pickle\_in)

pickle\_in = open("y.pickle","rb") y = pickle.load(pickle\_in)

import numpy as np

import matplotlib.pyplot as plt import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers import cv2

model=tf.keras.applications.mobilenet.MobileNet()

model.summary()

base\_input = model.layers[0].input

base\_output = model.layers[-4].output

Flat\_layer= layers.Flatten()(base\_output) final\_output= layers.Dense(1)(Flat\_layer)

final\_output= layers.Activation('sigmoid')(final\_output)

new\_model = keras.Model(inputs = base\_input, outputs= final\_output)

new\_model.summary()

new\_model.compile(loss="binary\_crossentropy", optimizer="adam", metrics= ["accuracy"])

new\_model.fit(X, Y, epochs= 8,validation\_split=0.1)

# also try training the model for 20 epochs & check the predictions (prediction=0 for closed eyes & prediction=1 for open eyes)

new\_model.save('my\_model2.h5')

import tensorflow as tf

new\_model = tf.keras.models.load\_model('E:\github\ML- ProjectKart\Driver Drowsiness Detection System\models\my\_model2.h5')

import numpy as np

import matplotlib.pyplot as plt import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers import cv2

img\_size=224

img\_array = cv2.imread('E:\github\ML-ProjectKart\Driver Drowsiness Detection System\Images\img2.jpg', cv2.IMREAD\_GRAYSCALE)

backtorgb =cv2.cvtColor(img\_array, cv2.COLOR\_GRAY2RGB) new\_array= cv2.resize(backtorgb, (img\_size, img\_size))

X\_input = np.array(new\_array).reshape(1, img\_size, img\_size, 3)

X\_input.shape

plt.imshow(new\_array)

X\_input=X\_input/255.0

prediction = new\_model.predict(X\_input)

prediction

img = cv2.imread('E:\github\ML-ProjectKart\Driver Drowsiness Detection System\Images\img4.jpg')

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))

faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_eye.xml')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

eyes = eye\_cascade.detectMultiScale(gray,1.1,4)

for(x,y,w,h) in eyes:

cv2.rectangle(img,(x, y), (x+w, y+h), (0, 255, 0), 2)

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))

eye\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades

+'haarcascade\_eye.xml')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) #print(faceCascode.empty())

eyes = eye\_cascade.detectMultiScale(gray,1.1,4)

for x,y,w,h in eyes:

roi\_gray = gray[y:y+h, x:x+w] roi\_color = img[y:y+h, x:x+w]

eyess = eye\_cascade.detectMultiScale(roi\_gray) if len(eyess) == 0:

print("eyes are not detected") else:

for (ex,ey, ew,eh) in eyess:

eyes\_roi = roi\_color[ey: ey+eh, ex:ex+ew]

plt.imshow(cv2.cvtColor(eyes\_roi, cv2.COLOR\_BGR2RGB))

eyes\_roi.shape

final\_image = cv2.resize(eyes\_roi, (224, 224)) final\_image = np.expand\_dims(final\_image, axis =0) final\_image=final\_image/255.0

final\_image.shape

new\_model.predict(final\_image) # 0 --> closed & 1 --> open

### CHAPTER 9

* 1. **Introduction to Testing**

### 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, subassemblies, 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.

#### Types of Testing

* + 1. **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputsproduce 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 a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process 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 is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully 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 test

Functional tests provide systematic demonstrations that functions = tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

* + - * Valid Input : identified classes of valid input must be accepted.
      * Invalid Input : identified classes of invalid input must be rejected.
      * Functions : identified functions must be exercised.
      * Output : identified classes of application outputs must be exercised.
      * Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered fortesting. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

#### System Testing

System testing ensures that the entire integrated software system meetsrequirements. It tests a configuration to ensure known and predictableresults. An example of system testing is the configuration oriented systemintegration test. System testing is based on process descriptions andflows, emphasizing pre-driven process links and integration points.

#### White Box Testing

White Box Testing is a testing in which in which the software testerhas knowledge of the inner workings, structure and language of thesoftware, or at least its purpose. It is purpose. It is used to test areas thatcannot be reached from a black box level.

#### Black Box Testing

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

#### Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail. Test objectives:

* + - * + All field entries must work properly.
        + Pages must be activated from the identified link.
        + The entry screen, messages and responses must not be delayed.

Features to be tested:

* + - * + Verify that the entries are of the correct format.
        + No duplicate entries should be allowed.
        + All links should take the user to the correct page.

#### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

#### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**All the test cases mentioned above passed successfully. No defects encountered.

### CHAPTER 10 OUTPUT SCREENSHOTS

**CHAPTER 11**

### CONCLUSION AND FURTHER ENCHANCEMENT

The driver abnormality monitoring system developed is capable of detecting drowsiness, drunken and reckless behaviours of driver in a short time. The Drowsiness Detection System developed based on eye closure of the driver can differentiate normal eye blink and drowsiness and detect the drowsiness while driving. The proposed system can prevent the accidents due to the sleepiness while driving. The system works well even in case of drivers wearing spectacles and even under low light conditions if the camera delivers better output. Information about the head and eyes position is obtained through various self-developed image processing algorithms. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. processing judges the driver’s alertness level on the basis of continuous eye closures.

### BIBLIOGRAPHY

[1] Miaou, “Study of Vehicle Scrap page Rates,” Oak Ridge National Laboratory, Oak Ridge, TN,, S.P.,April 2012.

{2] Wreggit, S. S., Kim, C. L., and Wierwille, W. W., Fourth Semi-Annual Research Report”, Research on Vehicle-Based Driver Status Performance Monitoring”, Blacksburg, VA: Virginia Polytechnic Institute and State University, ISE Department, January 2013.

1. Bill Fleming, “New Automotive Electronics Technologies”, International Conference on Pattern Recognition, pp. 484- 488,December 2012.
2. Ann Williamson and Tim Chamberlain,“Review of on-road driver fatigue monitoring devices”, NSW Injury Risk Management Research Centre, University of New South Wales, , July 2013.
3. E. Rogado, J.L. García, R. Barea, L.M. Bergasa, Member IEEE and E. López, February, 2013, “Driver Fatigue Detection System”, Proceedings of the IEEE International Conference on Robotics and Biometics, Bangkok, Thailand.
4. Boon-Giin Lee and Wan-Young Chung, Member IEEE, “Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals”, IEEE Sensors Journal, VOL. 12, NO. 7, July 2012.
5. H. Singh, J. S. Bhatia, and J. Kaur, “Eye tracking based driver fatigue monitoring and warning system”, in Proc. IEEE IICPE, New Delhi, India, Jan. 2014.
6. M. Hemamalini, P. Muhilan“Accident preventionusing eye blink sensor”, vol 1, Issue L11, 2017.
7. RamalathaMarimuthu, A. Suresh, M. Alamelu and S.Kanagaraj “Driver fatigue detection using image processing and accident prevention”, International journal of pure and applied mathematics, vol. 116, 2017.
8. TejaswiniJagdale, PradnyaJadhav, PrajaktaTotre,MayuraZadane, ShrilekhaMankhai “Driver drowsiness detection, alcohol detection and accident prevention”, IJET, vol3 issue1, jan 2017
9. BappadityaMandal, LiyuanLiyuan Li, Gang Sam Wang and JieLin ”Towards detection of bus driver fatigue based on robust visual analysis of eye state”,IEEE transaction on intelligent transportation systems,2016.
10. Suhaskatkar, Mahesh ManikKumbhar, PritiNavanathKadam”Accident prevention system using eye blink sensor”, IRJET, Vol.03 Issue 05, 2016.
11. Tejasweenimusale, prof B,H. Pansambal, ”Real time driver drowsiness detection system using image processing”, IJREAM, Vol 02, Issue 08, 2016.
12. Omkar, RevatiBhor, PranjalMahajan, H.V. Kumbhar “Survey on Driver‟s drowsiness detection system”,vol.132,2015.
13. Christy, Jasmeen Gill, “A Review: Driver drowsiness detection system”, IJCST, Vol.3 Issue 4,jul-aug 2015.
14. Deepa K B, Chaitra M, Ankit Kumar Sharma, Sreedhar V S ,Prashanth Kumar H.K “Accident prevention by eye blinking sensor and alcohol detector”, IJER, vol.no.4,issue no.7, 2015.
15. Rajasekar.R, Vivek Bharat Pattni, S.Vanangamudi “Drowsy driver sleeping device and driver alert system”, IJSR, Vol.3 Issue4,2014.
16. Swathi kale, RashmiBhadke,AnujaSali, NanasahebKadu”Drowsiness detection and warning system”IJARCST, Vol2, issue 2, 2014.
17. Davies, E.R. “Machine Vision: theory, algorithms, and practicalities”, Academic Press: San Diego, 1997.
18. Dirt Cheap Frame Grabber (DCFG) documentation, file dcfg.tar.z available from <http://cis.nmclites.edu/ftp/electronics/cookbook/video/>
19. Eriksson, M and Papanikolopoulos, N.P. “Eye-tracking for Detection of Driver Fatigue”, IEEE Intelligent Transport System Proceedings (1997), pp 314-319.
20. Gonzalez, Rafel C. and Woods, Richard E. “Digital Image Processing”, Prentice Hall: Upper Saddle River, N.J., 2002.
21. Grace R., et al. “A Drowsy Driver Detection System for Heavy Vehicles”, Digital Avionic Systems Conference, Proceedings, 17th DASC. The AIAA/IEEE/SAE, I36/1- I36/8 (1998) vol. 2.
22. Perez, Claudio A. et al. “Face and Eye Tracking Algorithm Based on Digital Image Processing”, IEEE System, Man and Cybernetics 2001 Conference, vol. 2 (2001), pp 1178-1188.
23. Singh, Sarbjit and Papanikolopoulos, N.P. “Monitoring Driver Fatigue Using Facial Analysis Techniques”, IEEE Intelligent Transport System Proceedings (1999), pp 314-318.
24. Ueno H., Kanda, M. and Tsukino, M. “Development of Drowsiness Detection System”,IEEE Vehicle Navigation and Information Systems Conference Proceedings, (1994), ppA1-3,15-20.
25. Weirwille, W.W. (1994). “Overview of Research on Driver Drowsiness Definition and Driver Drowsiness Detection,” 14th International Technical Conference on Enhanced Safety of Vehicles, pp 23-26.