DRIVER DROWSINESS DETECTION SYSTEM WITH ALARM SYSTEM

submitted in partially fulfilment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Tanish Singhal (2100970310164)
Utkarsh Mishra (2100970310171)
Vinay Kumar (2100970310177)

Under the Guidance of
Ms. Avinash Kaushal
Assistant Professor

Department of Electronics and Communication Engineering*(B.Tech ECE – Accredited by NBA),* Galgotias College of Engineering and Technology, Greater Noida.*(Affiliated to Dr.A.P.J Abdul Kalam Technical University, Lucknow)*

May, 2025

DECLARATION

We hereby declare that the thesis entitled **"Driver Drowsiness Detection System with Alarm System"** submitted by us, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering* to Galgotias College of Engineering and Technology, Greater Noida affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of bonafide work carried out by us under the supervision of **Ms. Avinash Kaushal.**

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Greater Noida

Date : **Signature of the Candidates**

CERTIFICATE

This is to certify that the thesis entitled **"Driver Drowsiness Detection System with Alarm System"** submitted by Tanish Singhal, Vinay Kumar, Utkarsh Mishra, Department of Electronics and Communication Engineering, Galgotias College of Engineering and Technology, Greater Noida affiliated to Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by him under my supervision, as per AKTU code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfils the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Greater Noida

Date: **Signature of the Guide**

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

Head of the Department

ABSTRACT

It is acknowledged that one of the main causes of traffic accidents worldwide is driver fatigue, which puts both drivers and passengers at serious risk. We advise the usage of a driver Drowsiness Detection and Alert device (DDDAS) to address this pressing hassle and appreciably lower the risks connected to drowsy riding. The DDDAS makes use of an expansion of sensors to cautiously monitor driving force behavior in real-time. The machine's fundamental factors are cameras with infrared (IR) sensors and eye reputation software program, each of which can be crucial for tracking symptoms of sleepiness. The tool may additionally analyze the accrued data and become aware of early indicators of tiredness, like extended eye closure, uncommon head angles, or frequent yawning, through utilizing machine mastering strategies. when the DDDAS detects these indicators, it immediately triggers a notification gadget supposed to warn the driver proper away, decreasing the risk of an coincidence. The efficacy of the DDDAS in exactly figuring out drowsiness and promptly alerting drivers has been established by way of massive and rigorous checking out. This has the capability to significantly improve visitors safety and forestall many collisions added on by tired drivers. The DDDAS's advent and application mark a considerable leap forward in fighting the vast hassle of sleepy using, ensuing in more secure using environments and, finally, lifesaving outcomes.

ACKNOWLEDGEMENT

First of all, we would like to express our gratitude to our beloved **Chairman Shri. Suneel Galgotia and CEO Shri. Dhruv Galgotia** for proving necessary facilities to carry out and finish the project successfully. We are grateful to our **Director Dr. V.K. Dwivedi** for his support and encouragement.

Our special thanks goes to **Dr. Reena Sharma, Head of the Department, Electronics and Communication Engineering**, for providing an environment that encouraged us in working towards our goal and supported in completion of our project. This project gave us an opportunity to apply creative and critical thinking skills.

We would like to present our deep sense of gratitude to our project guide, **Ms. Avinash Kaushal, Assistant Professor, Department of Electronics and Communication Engineering** who has always been a source of motivation and firm support for carrying out the project. The supervision and support that he gave truly helped in the progression of the project. His cooperation is much indeed appreciated. We are highly obliged to him for his valuable advices and moral support during research period.

Finally, our greatest and special gratitude goes to our family for their love and support.

Place: Greater Noida

Date:

Tanish Singhal
Utkarsh Mishra
VinayKumar
Pragati Singh

TABLE OF CONTENTS

Chapter No.	CONTENTS	Page No.
	Tittle Page	i
	Declaration	ii
	Certificate	iii
	Abstract	iv
	Acknowledgement	v
	Table of Contents	vii
	List of Figures	ix
	List of Tables	x
	List of Abbreviations	xi
1	**Introduction**	
	1.1 Introduction 1	
	1.2 Problem Statement	
	1.3 Projects Requirements	
	1.4 System Design and implementation	
	1.5 Implementation steps	
	1.6 Conclusion	
2	**Literature review**	
	2.1 Drowsiness Detection System Using Physiological System	
	2.2 Drowsiness Detection System Using Face Detection System	
	2.3 Perclos	
	2.4 Yawning Detection System	
	2.5 Drowsiness Detection with OPENCV	
	2.6 Driver Drowsiness Detection using ANN	
3	**Methodology**	
	3.1 Proposed Methodology	

3.2 Artificial intelligence 16
3.3 Machine Learning 18
3.4 Deep Learning 18
3.5 Neural Network 19
3.5.1 CNN 20
3.5.2 Back Propagation 31
3.5.3 Activation Function 32
3.5.4 Training 34
3.5.5 Train Dataset 38
3.5.6 Testing 39

4 Software Implementation 40

4.1 Python 40
4.2 Jupyter Notebook 40
4.3 Libraries 41
4.3.1 Open CV Library 41
4.3.2 Numerical Python 42
4.3.3 Dilib 46

5 Design Specification 44

5.1 Arduino UNO
5.2 Buzzer
5.3 Web Cam
5.4 16*2 LCD Display
5.5 Flowchart

6 Software Implementation 40

6.1 Result through Live Face 40
6.1.1 Frame recognized as drowsy 40
6.1.2 Frame recognized as not drowsy

7 Conclusion 51

References 52

List of Publication 56

Appendix 58

Curriculum Vitae 79

LIST OF FIGURES

Figure No.	Title	Page No.
1.1	Drowsiness Condition	2
1.2	Fatigue Condition	9
1.3	Difference between the eye marks when the eyes are open and the eyes are closed	10
1.4	Venn Diagram of AI	11
1.5	Layers in Neural Network	12
1.6	Operation of Convolution	15
1.7	Stride	2
1.8	Operation of Max pooling	9
1.9	Operation of average pooling	10
1.10	Residual Blocks	11
1.11	Architecture of Resnet	12
1.12	Back Propagation	15
1.13	Arduino UNO	2
1.14	PIN Configuration	9
1.15	Electromechanical Buzzer	10
1.16	16*2 LCD Display	11
1.17	Detecting and analyzing drowsiness.	12
1.18	Driver Input Image	15
1.19	Alert Image	2
1.20	Drowsy Image	9
1.21	Person in Drowsy state	10

LIST OF TABLES

Table No.	Tittle	Page No.
1.1	Channel for different turbulence conditions	14
5.1	Parameter used	44

LIST OF ABBREVIATIONS

FSO	Free space optical
PDF	Probability density function
GG	Generalised gamma
SIM	Subcarrier intensity modulation
PPM	Pulse position modulation
MSK	Minimum shift key
BPSK	Binary phase shift keying
QPSK	Quadrature phase shift keying
LED	Light emitting diode
RF	Radio Frequency
DPSK	Differential phase shift keying
OOK	On off keying
BER	Bit error rate
EGC	Equal gain combining
MRC	Maximum ratio combining
APD	Avalanche photo diode
OFC	Optical fibre communication
LASER	Light Amplification by Stimulated Emission of Radiation
ADSL	Asymmetric Digital Subscriber line
LAN	Local area network
DC	Direct Current
ALU	Arithmetic Logic unit
OBPF	Optical Band Pass filter
LPF	Low Pass filter
IC	Integrated Circuit
RV	Random variable
MIMO	Multiple input Multiple Output
SISO	Single Input Single Output
SIMO	Single Input Single Output

CHAPTER 1
INTRODUCTION

1.1 Introduction

Around 1.3 million individuals pass away each year due to car accidents, which are primarily caused by driver distraction and drowsiness. Many individuals travel long distances on highways, which can lead to fatigue and stress. Drowsiness can arise unexpectedly, resulting from sleep disorders, medication, or for instance, boredom can arise while driving for long periods. Therefore, drowsiness can create hazardous situations and elevate the likelihood of accidents.

Given the circumstances, it is crucial to employ modern technologies to develop and construct systems capable of monitoring drivers and assessing their attentiveness throughout the entirety of their time on the road.

Project team has developed a solution to prevent such accidents. The system involves utilizing a camera to capture the user's visual features, with the use of face detection and CNN techniques to identify any signs of drowsiness in the driver. When drowsiness is detected, an alarm will sound to alert the driver, prompting them to take precautionary measures. The amount of fatalities brought on by traffic accidents can be decreased in large part by detecting driver drowsiness.

1.2 Problem Statement

Human error-related visitors injuries are the purpose of many fatalities and accidents globally. The most important reason of those collisions is the driving force's fatigue, which can be delivered either by way of prolonged riding hours or loss of sleep. creating a system that makes use of the most modern era is vital to solving this trouble and decreasing the possibility of mishaps. This system's primary goal is to expand a model which could sound an alarm if the driving force famous any symptoms of fatigue. The motorist might be better capable of understand their kingdom and take the proper motion to keep away from an accident thanks to this be aware.

1.3 Project Requirements

1.3.1 Software

Software required are:

- Windows, Linux, MacOS used for operating system.
- Python 3.10(recent version) is used as language.
- Python IDE, Jupiter Notebook used as IDE's.

1.3.2 Hardware:

hardware required for this venture includes:

- **high Computational Processor:** to address the real-time processing wished for face detection and CNN operations
- **Minimum 4 GB RAM:** To ensure smooth operation and efficient data processing.
- **Webcam with Night Vision:** To capture the driver's facial features clearly, even in low light.
- **Arduino Microcontroller:** To serve as the central unit for processing inputs and controlling outputs.
- **LCD Display:** To provide visual feedback and alerts to the driver.
- **Speaker/Buzzer:** To provide audible alerts when drowsiness is detected.

1.4 System Design and Implementation

The machine integrates more than one hardware and software program components to locate and respond to motive force drowsiness efficiently. right here's a top level view of the important thing components and their roles:

- **Camera:** A night vision webcam continuously monitors the driver's face. The camera captures images that are processed in real-time.
- **Image Processing:** The captured snap shots are processed using face detection algorithms and CNN techniques to research the driver's facial capabilities and discover signs of drowsiness.
- **Arduino Microcontroller:** The Arduino acts because the coronary heart of the machine, interfacing with the digital camera, lcd show, and speaker. It tactics inputs from the digicam and triggers appropriate outputs.
- **Programming:** The Arduino is programmed the use of the Arduino IDE. The code consists of logic for reading information from the digital camera, studying the driver's situation, and triggering indicators.
- **LCD Display:** A 16x2 Liquid Crystal show (lcd) gives visible alerts and messages to the driving force, inclusive of "Drowsiness Detected" when symptoms of drowsiness are recognized

- **Speaker/Buzzer**: An audible alarm sounds thru a speaker or buzzer connected to the Arduino, making sure the motive force is right away notified in their drowsy circumstance

- ## 1.5 Implementation Steps

1 **Setting Up the Camera**: putting in place the digital camera: join the night time imaginative and prescient webcam to the gadget and position
2 it to surely capture the motive force's face.

3 **Programming the Arduino:** Write and upload the Arduino code that interfaces with the camera, processes the input, and controls the LCD display and speaker.

4 **Face Detection and CNN Implementation:** Use Python and relevant libraries to put into effect face detection and CNN for studying the driving force's facial capabilities. make certain the device can run these techniques in actual-time.

5 **Integration and Testing:** integrate the digital camera, Arduino, lcd show, and speaker. behavior vast checking out to make certain the machine as it should be detects drowsiness and triggers indicators promptly.

1.6 Conclusion

Human error-related traffic accidents are the cause of many fatalities and injuries globally. The main cause of these collisions is the driver's fatigue, which may be brought either by extended driving hours or lack of sleep. Creating a system that makes use of the newest technology is essential to solving this problem and reducing the possibility of mishaps. This system's primary goal is to develop a model that can sound an alarm if the driver exhibits any symptoms of fatigue. The motorist will be better able to recognize their state and take the appropriate action to avoid an accident thanks to this notice..

CHAPTER 2

LITERATURE SURVEY

2.1 Drowsiness Detection System Using Physiological Signals:

Author: T. S. Yengatiwar, Trupti K. Dange.

Publication year: 2013

Drowsiness can be detected by measuring physiological parameters like heart rate, pulse rate, breathing rate, respiratory rate, and body temperature are considered more precise and dependable in identifying drowsiness since they exhibit measurable physiological changes directly related to the driver's physical condition. When a person becomes drowsy, their physiological parameters tend to alter, for instance, a drop in blood pressure, heart rate, and body temperature. Drowsiness detection systems that rely on these physiological indicators can identify such changes and caution the driver when they are at risk of falling asleep. However, since these systems necessitate electrodes to be attached to the driver's body, they are considered invasive. Here is a compilation of drowsiness detection systems that rely on physiological conditions.

2.2 Drowsiness Detection Using Face Detection System

Sleepiness can be distinguished by utilizing face territory recognition. The techniques to identify sleepiness inside the face territory shift because of languor. Sign in are more noticeable and clear to be identified at the face territory, we can identify the eyes area. From eyes identification, the creator expressed that there are four kinds of eyelid development that can be utilized for laziness location. They are totally open, total close, and in the center where the eyes are from open to close and the other way around.

The calculation measures the picture caught in a dark scale strategy; where the tone from the pictures is then changed into highly contrasting. Working with highly contrasting pictures is simpler on the grounds that lone two boundaries must be estimated. The creator at that point plays out the edge discovery to identify the edges of eyes so the estimation of the eyelid territory can be determined.

The issue happening with this technique is that the size space of the eye may shift starting with one individual then onto the next. Somebody may have little eyes and appears as though it is drowsy yet some are most certainly not. Other than that, if the individual is wearing glasses there is a hindrance to distinguish eye area. The pictures that are caught should be in a specific reach from the camera since when the distance is a long way from the camera, the pictures are obscured.

FIG 2.1: Drowsiness Condition

2.3 Perclos (PERCENTAGE OF EYE CLOSURE):

Drowsiness can be caught by recognizing the eye squints and level of eye conclusion (PERCLOS). For eye squint identification, propose a strategy which learns the example of span of eyelid shut. As per this proposed technique estimates the ideal opportunity for an individual shut their eyes and on the off chance that they are shut longer than the ordinary Eye flicker time, it is conceivable that the individual is nodding off. It is referenced that eye flicker time, it is conceivable that the individual is nodding off. It is referenced that almost 310.3ms are the normal of an ordinary individual's eye squint.

Subsequent to going through the exploration papers and the current techniques, this task suggested that eyes and yawning location strategies will be utilized. Eye flicker term gives the information that the more extended the individual's nearby their eyes, the drowsier it.

PERCLOS technique recommends that languor is estimated by ascertaining the level of the eyelid 'hangs'. Sets of eye open and eye shut have been put away in the product library to be utilized as a boundary to separate whether the eyes is completely open or completely shut. For

eyelids to hang, it occurs in much more slow time as the individual is gradually nodding off. Consequently, the progress of the driver's laziness can be recorded.

FIG 2.2: Fatigue Condition

In this way, PERCLOS strategy puts a corresponding worth where when the eyes is 80% shut, which it is almost to completely close, it accepts that the driver is lazy.

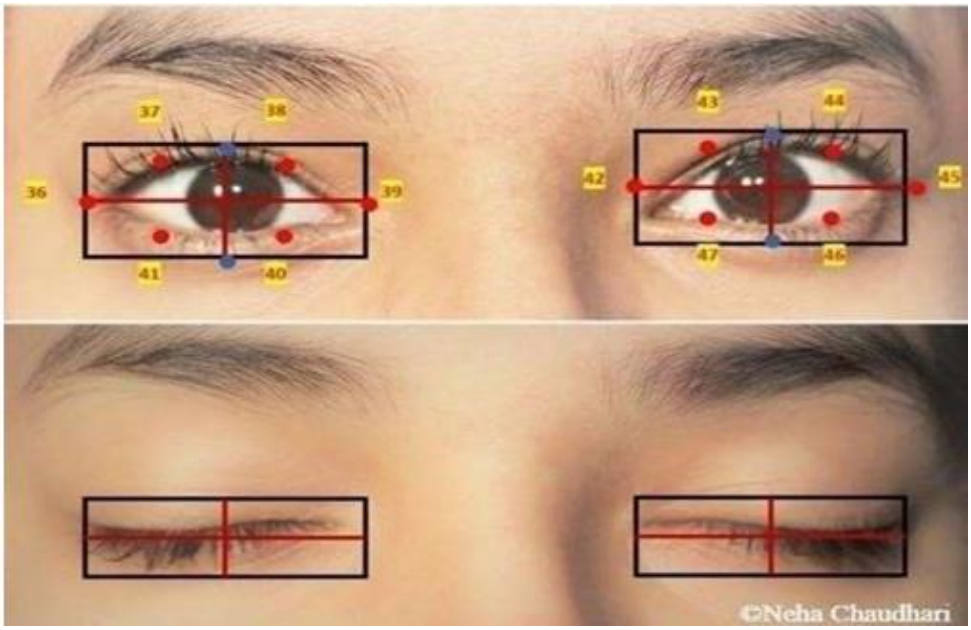


FIG 2.3: Difference between the eye marks when the eyes are open and the eyes are closed

This technique isn't advantageous to be shut to be utilized progressively driving as it needs a fixed limit estimation of enlightening for the PERCLOS strategy to Perform precisely. The two strategies to distinguish tiredness utilizing eye flicker example and PERCLOS have a similar issue where the camera should be put at a particular point to get a decent picture of video with no aggravation of eyebrow and shadow that cover the eyes.

2.4 YAWNING DETECTION METHOD:

As per, sleepiness of an individual are regularly seen by watching their face and conduct. The creator proposes a way where laziness are regularly identified by month situating and thusly the pictures were prepared by utilizing a course of classifiers that has been proposed by Viola - Jones for faces. The photos were contrasted and the arrangement of pictures information for mouth and yawning. A few group will close their mouth with their hand while yakking.

2.4.1 EEG-BASED DRIVER FATIGUE DETECTION

A system has been suggested for identifying driver fatigue and exhaustion prevent Car accidents resulting from drivers who were sleepy or drowsy. This system uses Electroencephalogram (EEG) signals to determine the degree of sleepiness or drowsiness experienced by a driver. The system first identifies an index that corresponds to various levels of drowsiness. A cheap neuro signal acquisition device with a single electrode is used to obtain the EEG signal from the driver. A collection of data designed for simulated car drivers experiencing different levels of drowsiness was collected locally to evaluate the system. The findings indicated that the system proposed was successful in detecting fatigue in all the subjects.

2.4.2 PULSE SENSOR METHOD:

Previous studies have primarily focused on using drivers' physical conditions to detect drowsiness. To address this issue, Rahim developed a system that uses infrared heart rate or pulse sensors to detect drowsy drivers. The pulse sensor gauges the heart's pulse rate by detecting the driver's finger or hand. By detecting the quantity of blood circulating through the finger, the sensor can determine the amount of oxygen in the blood, causing reflecting infrared light and transmit the data to the Arduino microcontroller. The variation in oxygen levels is then processed by HRV frequency domain software to visualize the driver's heart pulse rate.

The results of the experiment showed that the LF/HF As drivers shift from being alerted to feeling drowsy, the ratio of oxygen tends to decline. By issuing timely warnings, numerous car accidents can be averted.

2.4.3 WEARABLE DRIVER DROWSINESS DETECTION SYSTEM

In the past, Applications designed for mobile devices have been created to identify and detect driver drowsiness. However, these applications can distract drivers and lead to accidents. To address this issue, Lenget developed a drowsiness detection system in the shape of a custom-designed wristband that can be worn has been developed, featuring a PPG signal and galvanic skin response sensor. The information gathered by these sensors is sent to a mobile device, which functions as the primary assessment unit. Motion sensors in the mobile device analyse the data, and five features (heart rate, breathing rate, level of stress, variability in pulse rate, and the count of adjustments made) are extracted for computation. These characteristics are subsequently employed as calculation parameters for an SVM classifier, which is utilized to assess the driver's level of drowsiness. The findings of the experiment indicated an accuracy up to 98.02% for the proposed system. In the event of drowsiness, the mobile device generates a warning system that makes use of both visual and vibrational alerts to notify the driver.

2.4.4 WIRELESS WEARABLES METHOD:

Warwick has proposed a drowsiness detection system that utilizes a wearable biosensor known as Bio-harness to reduce the likelihood of road accidents. The system is comprised of two stages. During the first stage, the Bio-harness gathers the driver's physiological data, such as ECG, heart rate, and posture, among other metrics. This data is then analysed to determine key parameters linked to drowsiness. In the second stage, a drowsiness detection algorithm is established, and a mobile application is developed to warn drowsy drivers.

2.5 Drowsiness Detection with OpenCV using EAR:

Author: Adrian Rosebrock

Publication year: 2017

The paper proposes an algorithm that can detect eye blinks in real-time using video footage from a standard camera. The algorithm utilizes this paper proposes the use of landmark detectors that are trained on datasets containing images of people in everyday settings (in-the-

wild datasets) for the purpose of detecting eye blinks in real- time from a video sequence captured by a standard camera, which are highly robust against different factors such as head orientation, varying illumination, and facial expressions. These detectors can precisely detect facial landmarks and the algorithm is designed to estimate the degree of eye opening, which is essential for detecting eye blinks accurately.

Several techniques have been proposed for automatic identification of eye blinks in video sequences, and some of these methods depend on analysing the motion within the eye region. Various techniques have been proposed to automatically identify eye blinks in video sequences, including methods that rely on analysing the motion in the eye region. Typically, these methods involve detecting the face and eyes using a detector such as the Viola-Jones algorithm. Then, the movement in the eye area is evaluated using techniques such as estimating optical flow, sparse tracking, or frame- to-frame intensity differences with adaptive thresholding. Finally, the algorithm determines whether the eyes are covered by eyelids or not.

At present, there are facial landmark detectors available that can accurately capture various key points on a human face image, including the corners of the eyes and the eyelids, with high reliability in real-time. These landmark detectors are advanced and use a regression approach, where a mapping is learned from an image to the positions of the landmarks or another landmark parametrization. They are trained on datasets that contain images taken in diverse settings, which makes them robust to 11 challenges like changes in illumination, various facial expressions, and moderate non- frontal head rotations.

Proposed method:

To blink is to quickly shut and then reopen one's eyes., and the pattern of blinks varies slightly from person to person, including differences in speed, degree of eye squeezing, and blink duration. Typically, an eye blink lasts anywhere between 100 to 400 milliseconds. In this paper, it is proposed to use advanced facial landmark detectors to locate the eyes and define the shape of the eyelids in an image. Based on the landmarks detected, the eye aspect ratio (EAR) is computed as an indicator of the degree of eye-opening. However, since the EAR value in each frame may not be able to accurately detect eye blinks, a classifier is trained to analyse a longer sequence of frames. When an eye is open, the EAR value remains relatively stable, but it gradually decreases towards zero as the eye closes.

The article introduced an algorithm capable of detecting eye blinks in real-time. It was shown that facial landmark detectors based on regression techniques can accurately estimate the degree of eye openness. These detectors are also highly robust against various challenges, such as low image quality (mostly due to low resolution) and real-world factors like non-frontal head positions, poor lighting, and facial expressions. The proposed approach, which employs a Support Vector Machine (SVM) and considers a temporal window of the eye aspect ratio (EAR), performs better than the EAR thresholding method.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

2.6 Driver Drowsiness detection using ANN image processing:

Authors: S. Moca1, T. Vesselenyi1, B. Tătaru1, A. Rus1, T. Mitran1.

Publication year: 2017.

This study aimed to explore the feasibility of developing a drowsiness detection system for car drivers using three methods: processing of EEG and EOG signals, and analysis of driver images based on eye state (open or closed) classification. The EEG and EOG methods are used to measure brain activity and signals from the muscles responsible for eye movement, respectively. On the other hand, the eye image analysis method involves observing the state of the eye, whether it is open or closed.

The EEG and EOG sensors Electrodes need to be positioned on specific parts of the body and be connected through conductive gel or wires, causing discomfort to the user. However, advancements the problems associated with traditional EEG methods may be addressed by utilizing advancements in materials science and MEMS technology, including the application of dry electrodes for EEG.

The advancement of EEG technology has been largely fuelled by the development of brain-computer interfaces for various applications, including devices to assist people with disabilities. One of the primary goals of recent EEG research is to differentiate between low and high alpha rhythm peaks, which can be used to determine a person's level of alertness. The

study involved using EOG signals from three sensors (EOG1, EOG2, EOG3) to identify four distinct signal types after pre processing. By combining these signal types, researchers were able to determine the direction of eye movements (upward, downward, leftward, and rightward), providing the necessary information to differentiate between alert and drowsy states.

The researchers used MATLAB Neural Network Toolbox and Deep Learning Toolbox's autoencoder module to determine if these tools could be used to classify driver drowsiness based on images. They acquired 200 images of a driver during normal driving, with half showing open or half-open eyes and the other half showing closed eyes. The hypothesis was that closed eyes would indicate drowsiness, while open or half-open eyes would indicate an alert state. They used a one-layer artificial neural network for analysis.

CHAPTER 3
METHODOLOGY

3.1 Proposed Methodology

3.1.1 Existing System:

There are several existing systems for detecting driver drowsiness, and they are:

- Various technologies can detect drowsiness in drivers to prevent accidents. Video-based systems use cameras to monitor the driver's face and identify signs of drowsiness such as drooping eyelids, head nodding, or yawning.
- Infrared-based systems use infrared sensors to detect changes in skin temperature, which can indicate drowsiness.
- EEG-based systems utilize electrodes on the driver's scalp to measure brain activity and identify changes in brain waves that correspond to drowsiness.
- Wearable devices that monitor changes in heart rate, breathing, and movement can also indicate drowsiness.
- Steering-based systems use sensors in the steering wheel to detect changes in grip strength or steering behavior that can indicate drowsiness.

3.1.2 Proposed System:

In this system, instead of existing systems, an alternative approach was used:

- Artificial intelligence and machine learning algorithms are utilized in machine learning-based systems to detect drowsiness. These systems use data from multiple sources, such as video and sensor data, to identify patterns and indications of drowsiness.

A Convolutional Neural community (CNN) is the model used in this scenario, that is regularly used for image category and multi-magnificence class of snap shots. Convolution layers with adjustable filters make up the CNN. A forward propagation method is used to move the filters throughout the input; every motion is referred to as a stride. The CNN model improves the gadget's accuracy. the motive force's face is constantly photographed via a digicam, and it's miles recognized through a face detection device. next, a CNN-based totally classification

algorithm is used to determine if the driver's face is drowsy or not. The class version is constructed using the KCF and RESNET CNN.

.

3.2. Artificial Intelligence (A.I):

The intention of laptop technological know-how's artificial intelligence (AI) field is to construct sensible machines which can motive, research from experience, and make decisions the use of statistics. Self-riding vehicles and scientific analysis and remedy are best of the many uses for artificial intelligence (AI), which has advanced substantially in current years. AI comes in a spread of paperwork, such as system mastering, deep learning, and rule-primarily based or symbolic AI. Rule-based or symbolic AI bases its choices on a series of if-then statements and applies good judgment and pre-hooked up policies. AI that allows machines to study from information with out explicit programming is known as device mastering. To locate patterns in facts and generate predictions based totally on those styles, it employs algorithms.Neural networks are used in deep getting to know, a department of system studying, to mimic the functioning of the human brain. it is able to processing big volumes of records and producing enormously accurate predictions. herbal language processing (NLP), pc vision, robotics, healthcare, and finance are only a few of the fields that use synthetic intelligence (AI). synthetic Intelligence (AI) is used in natural language processing to investigate and recognise human language, enhancing system-human interaction. AI is utilized in computer vision to analyze pics and movies, permitting gadgets to pick out and apprehend gadgets, human beings, and other visible information.AI is being applied in robotics to construct intelligent, self-sufficient robots. AI is utilized in healthcare to research scientific facts and forecast ailments, allowing physicians to diagnose and treat patients extra appropriately. synthetic intelligence (AI) is used in finance to investigate economic records and forecast markets, enabling traders to make wiser selections.

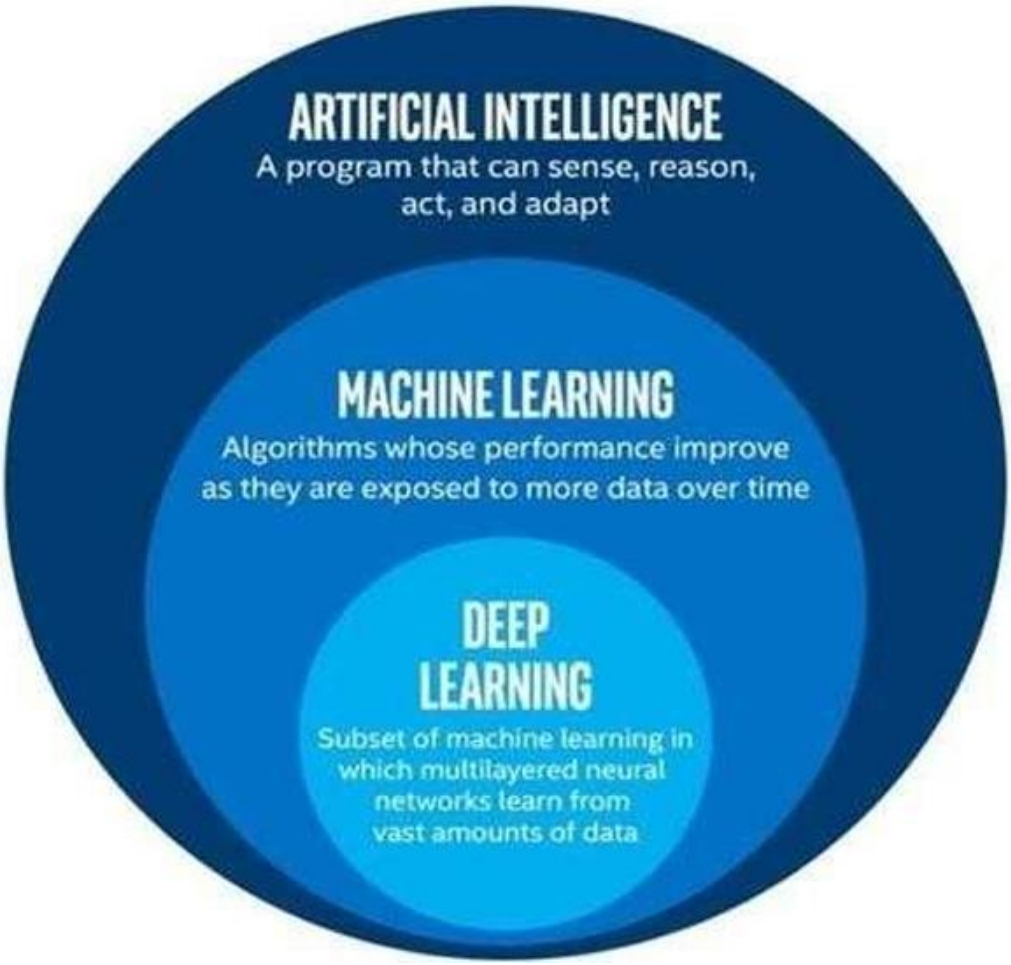


FIG 3.1: Venn Diagram of A.I

3.3 Machine Learning:

A department of artificial intelligence (AI) known as gadget gaining knowledge of allows pc systems to examine from experience and improve overall performance with out the need for explicit programming. This department of pc technological know-how specializes in creating laptop packages that become aware of and extract patterns from information using statistical fashions and strategies. massive improvements have been executed on this region in current years, and gadget mastering's primary aim is to develop programs that could accumulate data and use it to enhance their performance on their personal. in spite of the fact that there are many distinct gadget learning algorithms, supervised, unsupervised, and reinforcement studying are actually the 3 important techniques in use .not with standing the fact that there are many exceptional sorts of device mastering algorithms that are employed to be used instances, the 3 principal strategies which are currently in use are:

i oversaw an ML algorithm.

ii Reinforcement of Unsupervised ML Algorithms

iii machine gaining knowledge of set of rules A supervised machine mastering algorithm is the only used on this system out of all of the different types of system getting to know algorithms that are accessible.

3.4 Deep Learning:

artificial intelligence that mimics the human brain's ability to research information and see patterns to resource in choice-making is referred to as deep mastering. it is a department of system getting to know in artificial intelligence that uses networks that can analyze from unstructured or unlabeled records without supervision. Deep neural networks or deep neural mastering are different names for this approach. A machine gaining knowledge of approach called "deep studying" makes use of sophisticated algorithms that may system and determine on unstructured facts without human oversight. This approach makes it possible for deep getting to know structures to discover styles and carry out duties like language translation and object and speech reputation .A department of synthetic intelligence referred to as deep gaining knowledge of mimics how the human mind translates records to perform a diffusion of features, along with object detection, speech reputation, language translation, and decision-making. Deep studying, in evaluation to conventional machine mastering techniques, can take care of unstructured and unlabeled information and research on its very own with out human help. This generation may be used for a variety of purposes, along with preventing money laundering and fraud.

3.5Neural Networks:

synthetic structures referred to as neural networks are based totally on the architecture and operation of organic mind networks. without particular instructions, those networks are capable of learn and alter to new inputs. as an alternative, they independently analyze datasets and examples to discover tendencies and connections. Neurons, connections, biases, weights, a propagation characteristic, and a getting to know algorithm are some of the elements that make up a neural community. every neuron has an output function, threshold, and activation function,

and they all obtain input from in advance neurons. Weights and biases inside the connections between neurons modify the drift of records among them. each neuron's input and output are decided by means of the propagation function the use of the function of the neurons that came earlier than it and the weights that went with it. The mastering rule is in fee of enhancing the network's variable weights and thresholds. multiple layers of devices are related to each other in neural networks, and the weight of every connection controls how one unit affects the others. The enter layer of the network receives records, which is sooner or later sent via some of layers earlier than the output layer generates the very last output. The community acquires a deeper comprehension of the records and learns from it alongside the system, which permits it to supply specific classifications or predictions.

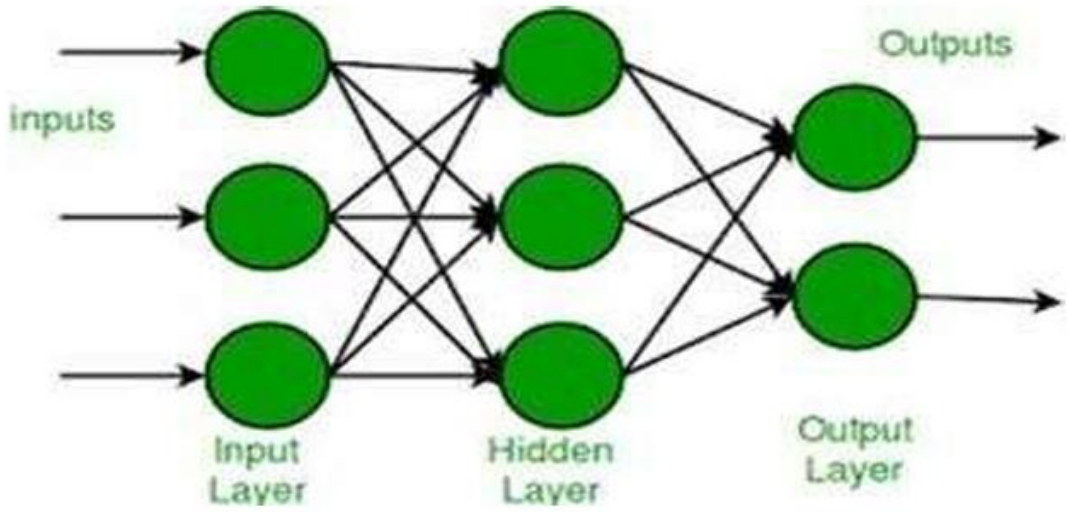


FIG 3.2 Layers in Neural Network

3.5.1 Convolution Neural Network (CNN):

One type of Deep gaining knowledge of technique is a Convolutional Neural network (ConvNet/CNN), which makes use of learnable weights and biases to investigate an image input and assign importance to numerous characteristics or items in the photograph. less pre-processing is wanted for ConvNets than for other type strategies. traditional strategies want the guide advent of filters, however with sufficient schooling, a ConvNet can analyze those filters. With the right filters, a Convolutional Neural network (ConvNet/CNN) may successfully capture the temporal and spatial relationships found in an photograph. because of its structure, which reuses weights and reduces the wide variety of parameters, the photo dataset may be higher equipped. put extra absolutely, the community may be trained to extra well recognize the photograph's complexity.a selected sort of linear method called convolution in arithmetic

is stated via the word "Convolution" in Convolutional Neural Networks (CNN). Convolution is the process of multiplying features to create a 3rd characteristic that illustrates how one alters the shape of the other. to put it another way, CNNs generate an output that extracts functions from photographs via multiplying two matrix representations of the pics.

3.5.1.1 Convolutional Layer:

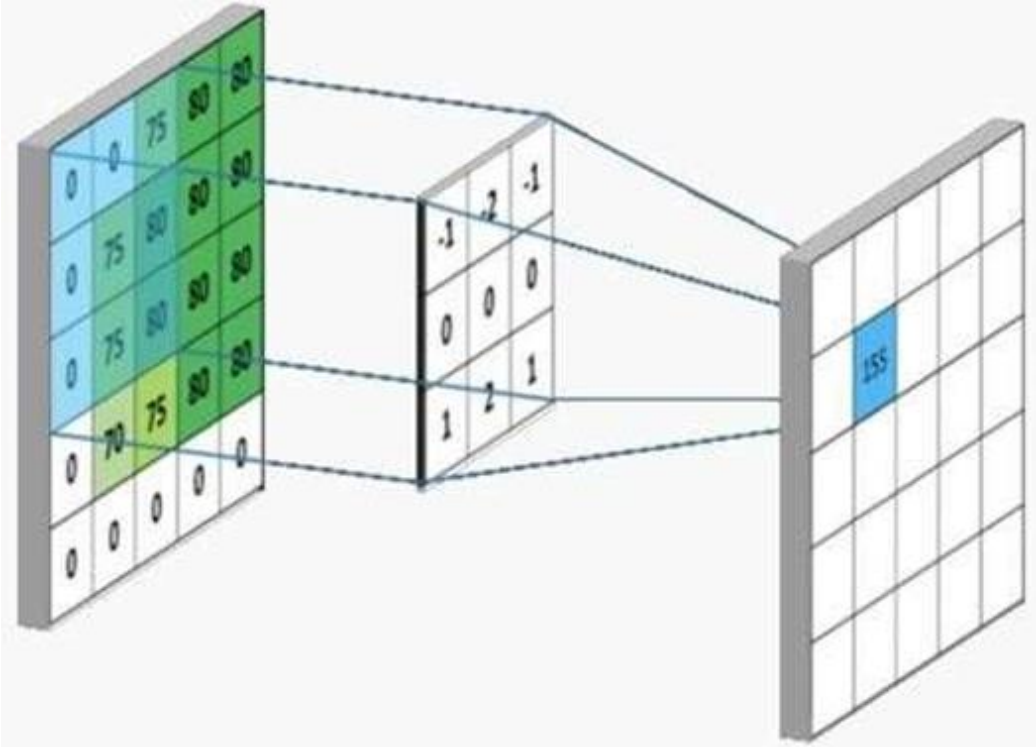
One sort of layer in neural networks that applies the convolution operation to enter information is referred to as a convolution layer. typically, it's miles carried out to computer imaginative and prescient and photograph reputation jobs. a group of filters, now and again known as kernels or characteristic maps, make up the convolution layer. these filters pass across the enter records to perform the convolution operation. every filter appears for a particular sample or function in the incoming information, such corners or edges. a collection of function maps, each representing a filter's response to the enter facts, is the convolution layer's output. To limit their length and computational complexity, the characteristic maps are generally down sampled the use of a pooling technique, like max pooling or average pooling. The convolution layer's parameters encompass the padding and stride values used at some point of the convolution process, in addition to the size and quantity of filters. The network may analyze the most beneficial set of filters for a activity by using backpropagation to analyze these parameters throughout schooling.

The Convolution Operation

A mathematical procedure that is often employed in photograph and sign processing is convolution. Convolution is a technique used in neural networks to extract characteristics from input statistics, which include audio signals or photographs. Sliding a tiny matrix, also called a kernel or filter out, across the incoming facts is the convolution process. The values of the kernel and the matching values of the input information are elevated together at each kernel point before being brought together. The outcome is a single output cost that shows how comparable the enter facts and the kernel are at that precise region.a new output facts structure is produced by repeating the process of sliding the kernel over the enter information at every

factor. commonly, the output data structure's size is smaller than the input data structure, depending on the size of the kernel and the amount of padding used.

FIG 3.3:



Operation of Convolution

Stride

the usage of a given stride cost, the filter out traverses the picture's width from left to right. The filter out repeats the manner till it has blanketed the whole picture after accomplishing the cease of the row and then proceeds down to the picture's leftmost factor the usage of the identical stride fee. The filters develop one pixel at a time if the stride price is 1. The filters shift two pixels at a time if the stride value is 2. Convolution with a stride fee of two is verified within the diagram underneath.

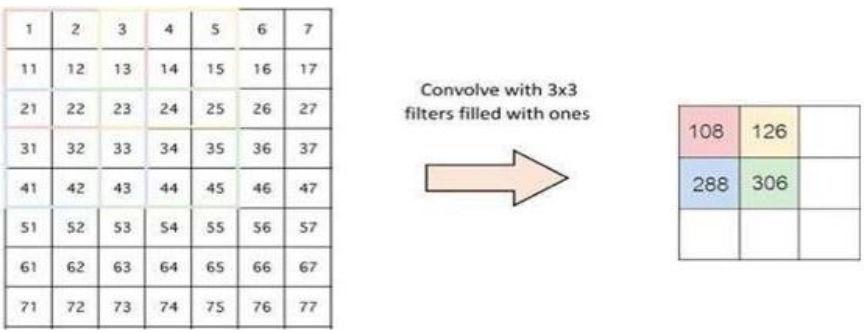


FIG 3.4: Stride

3.5.1.2 Pooling Layer:

pooling layer is a popular layer in convolutional neural networks (CNNs). Its number one motive is to preserve the big capabilities that the filters have learned at the same time as shrinking the spatial dimensions of the output characteristic maps produced by the convolutional layer. average pooling and maximum pooling are the 2 primary styles of pooling layers.

3.5.1.3 Max Pooling:

Usually placed after a convolutional layer, a The input function map is split into non-overlapping home windows or regions for optimum pooling; those home windows are usually 2x2 or 3x3. The maximum fee inner every window is chosen and delivered to the output function map, with the ultimate values being eliminated. with the aid of repeating this manner for every window, the feature map is basically down sampled, keeping simplest the most powerful activations.

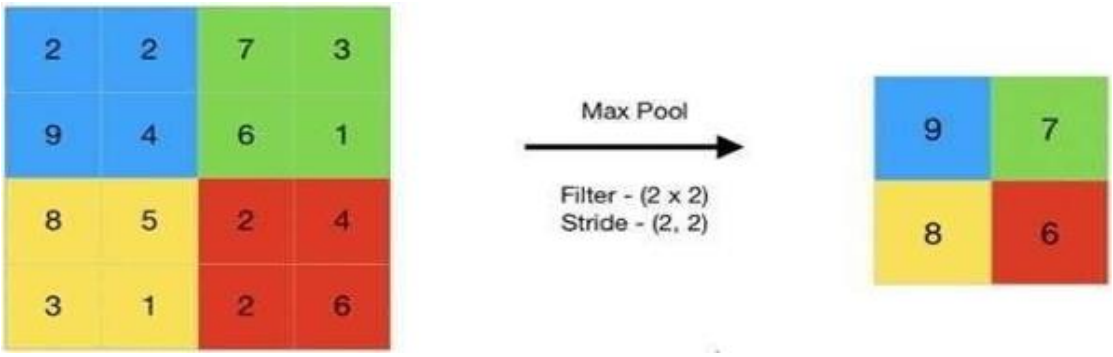


FIG 3.5:

Operation of Max Pooling

3.5.1.4 Average Pooling:

The input feature map is separated into non-overlapping windows or areas for common pooling; these home windows are normally 2x2 or 3x3. The output characteristic map is created by means of calculating the common fee of all the activations within each window. by repeating this procedure for every window, the characteristic map is essentially down sampled while the distribution of activations inside each vicinity is preserved.

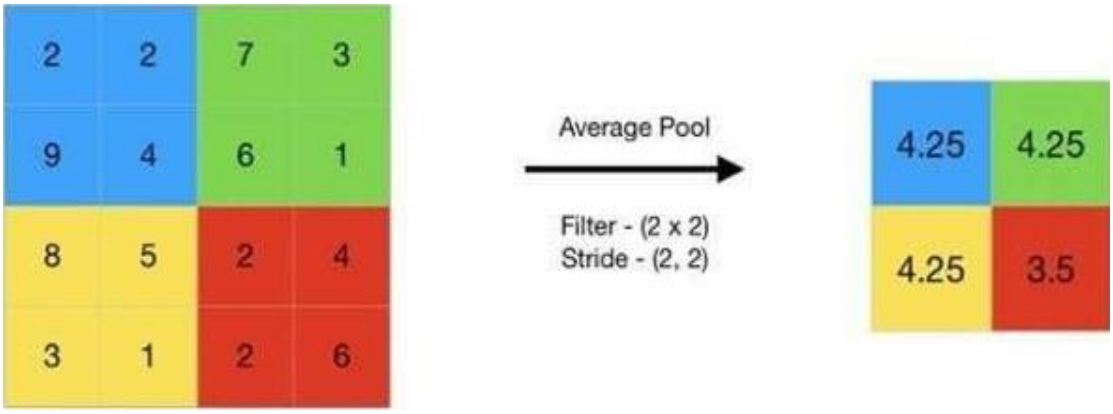


FIG 3.6: Operation of Average Pooling

3.5.1.5 Global Pooling:

a selected type of pooling layer called global pooling reduces the characteristic map to a single fee instead of segmenting it into numerous regions. global average pooling is the most famous kind of worldwide pooling, wherein the output contains the average value of each activation inside the function map. For classification duties, wherein the community's output have to be a set-size vector expressing the probability distribution across the available instructions, international pooling is often applied within the closing layers of a CNN.

3.5.1.6 Fully Connected Layer:

a fully related layer (FC layer) is a form of neural network layer utilized in deep getting to know, in which each neuron in a single layer is attached to every different neuron within the layer above. This indicates that each enter neuron has a corresponding weight and bias, and that every one input neurons are linked to all output neurons. every other name for a totally linked layer is a linear layer or a thick layer. It usually serves as the neural network's final layer, performing the input data's final classification or regression. After adding a bias term and an activation function to the weighted sum of the inputs, the output of a fully connected layer is determined. Backpropagation, a type of supervised mastering technique, is used to study the weights and biases inside the fully linked layer.

3.5.1.7Dropout Layer:

Neural networks appoint dropout as a regularization technique to avoid overfitting. it's miles positioned into exercise as the dropout layer, a layer inside the community layout. During

training, the dropout layer randomly chooses a subset of the neurons in the preceding layer and sets their outputs to zero. This implies that the network's information flow via those neurons is momentarily cut off, and the remaining neurons must learn to cooperate in order to make up for the lost data. The entire set of neurons is used for testing, while the dropout layer is only employed during the training phase.

3.5.1.8 RESNET

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun presented the ground-breaking deep neural network architecture known as ResNet, or "Residual Network," in their 2015 publication "Deep Residual Learning for Image Recognition." Deep studying has greatly benefited from this structure, mainly in tackling the difficulties worried in training extremely deep networks.

Key Concept: Residual Connections

Residual connections, from time to time called bypass connections, are the main innovation of ResNet. via immediately connecting the input of 1 layer to the output of a following layer, these connections allow the network to get round one or more layers. The idea is to let the network learn the residual mapping instead of the original unreferenced mapping.

- **Mathematical Formulation:** In a traditional feedforward neural network, the output of a layer

  $H(x)$ is a function of its input x. In ResNet, the output of a layer is given by $H(x) = F(x, \{W_i\}) + x$, where $F(x, \{W_i\})$ represents the residual function (the transformation applied by the layer) and x is the input that is directly added to the output.

Advantages of Residual Connections:

1. **Mitigating Vanishing Gradients:**
   - In very deep networks, gradients can become extremely small during backpropagation, effectively preventing the network from learning. Residual connections help by providing shorter paths for gradient flow, thus preserving gradient strength and enabling the training of much deeper networks.
2. **Improved Learning:**
   - By learning residuals (the difference between the desired output and the input), ResNet models can more easily optimize their layers. This approach simplifies the

learning process, as the network can focus on learning the residuals, which are often easier to model.

3. **Ease of Training:**
   - ResNets can be trained more efficiently and effectively compared to traditional deep networks. This efficiency has allowed researchers to experiment with much deeper models, sometimes exceeding 100 or even 1000 layers.

Architectural Details:

1 **Building Blocks:**
   - ❖ The basic building block of a ResNet is the residual block, which consists of a few convolutional layers with batch normalization and ReLU activation functions, followed by an addition operation that incorporates the residual connection.

2 **Bottleneck Layers:**
   - ❖ In deeper versions of ResNet, such as ResNet-50 and ResNet-101, bottleneck layers are used. These layers use 1x1 convolutions to reduce (and then restore) the dimensionality, which makes the model more computationally efficient without sacrificing performance.

3 **Variants:**
   - ❖ Several variants of ResNet have been developed to suit different needs. Among these are ResNet-18 and ResNet-34, which are shallower networks appropriate for easier jobs. ResNet-101, ResNet-152, and ResNet-50 are deeper networks that perform better on challenging tasks. 27 ResNeXt: An extension that adds a new dimension known as cardiacity, which boosts network performance by expanding the amount of ways data can move through the system.

   - ❖ **Impact and Applications:**

- **Image Recognition:**
   - ❖ ResNet achieved state-of-the-art performance in the ImageNet competition, significantly outperforming previous models and setting new benchmarks for image classification tasks.

- **Transfer Learning:**
  - ❖ Pre-trained ResNet models are frequently utilized in transfer learning because of their strong design. those fashions may be optimized for a number of specialized obligations, inclusive of segmentation, object detection, or even packages in adjoining fields like herbal language processing..

- **Research and Industry:**
  - ❖ ResNet is a fundamental model in the deep mastering community because the thoughts of residual getting to know have fashioned many later designs and studies avenues. it's far widely hired in each business and scholarly programs, from clinical photograph analysis to driverless cars.

3.5.1.9 Residual Blocks:

The simple unit of a Residual network (ResNet) design is known as a Residual Block. The community can bypass one or more layers and switch statistics straight between layers way to a series of shortcut connections that come after one or greater convolutional layers.

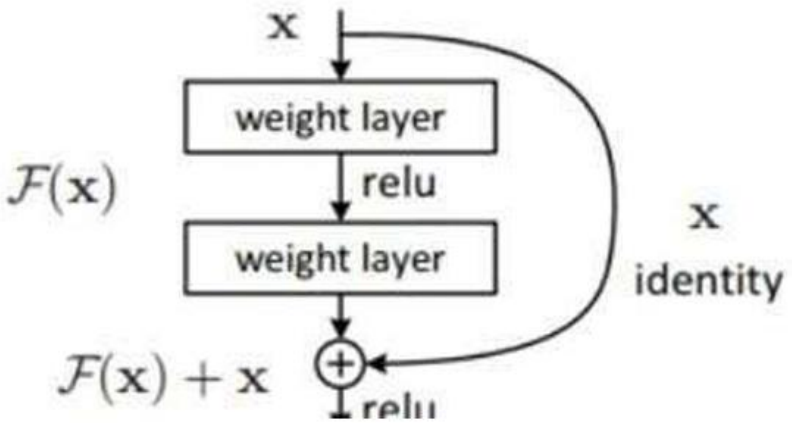


FIG 3.7 Residual Blocks

In a residual block, the shortcut connection is a bypass connection that adds one or greater layers' output to the block's output. the idea is that a layer must study a residual mapping in preference to an instantaneous mapping if its enter can be represented through the sum of its output and the input to the layer. because it's far less complicated for the community to analyze the difference among the enter and output rather than the direct mapping, this permits the

network to analyze more correct and green representations. it's been demonstrated that the residual block works quite properly to make it viable to teach very deep neural networks.The gradient can undergo the community extra with ease throughout backpropagation by using establishing shortcut connections between the layers. This lessens the issue of vanishing gradients and makes it possible to train deeper networks..

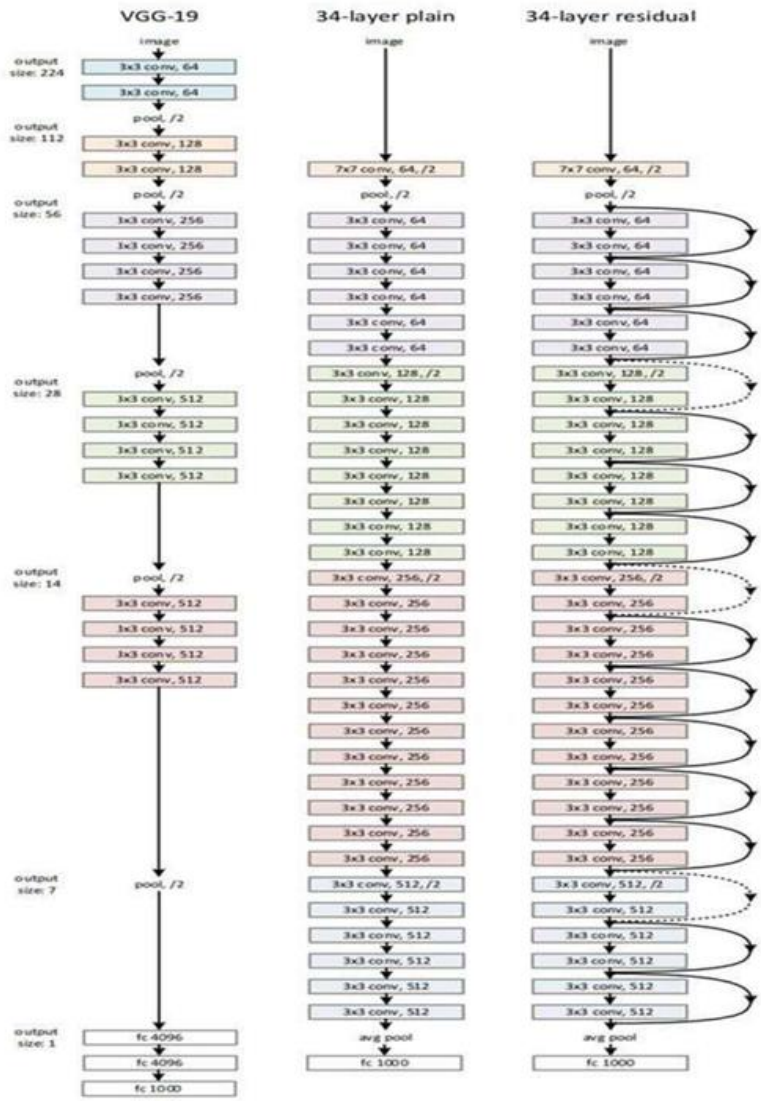


FIG 3.8 Architecture of ResNet

3.5.1.10 Architecture Of RESNET:

With the addition of skip or shortcut connections, a 34-layer plain network's architecture is based on VGG-19. Residual blocks are used to implement these connections, transforming the architecture into a residual network. This is shown in the figure below as a diagram.

3.5.1.11 Using ResNet with Keras:

Combining ResNet with Keras: Keras is a free deep learning library that may be used with TensorFlow. There is a component in Keras called Keras Applications that provides many ResNet versions.

3.5.1.5.3.1 ResNet50

3.5.1.5.3.2 ResNet50V2

3.5.1.5.3.3 ResNet101

3.5.1.5.3.4 ResNet101V2

3.5.1.5.3.5 ResNet152

3.5.1.5.3.6 ResNet152V2

3.5.2 Back Propagation:

One popular approach for training artificial neural networks is backpropagation. This optimization technique determines the gradients of the loss function in relation to the network's weights, enabling the weights to be changed in a way that minimizes the loss.

In order for the backpropagation algorithm to function, the input must first be propagated forward through the network, followed by calculations of each neuron's output and the loss between the predicted and actual outputs.

After that, it calculates the gradient of the loss with recognize to every weight within the network and propagates the error back via the community. that is executed via calculating the spinoff of the loss with recognize to every intermediate output inside the network using the chain rule of calculus. The spinoff of the loss with respect to each weight is then determined the use of these derivatives. After the gradients are computed, an optimization approach like gradient descent is employed to update the network's weights. In order to lower the loss function, the optimization algorithm iteratively modifies the weights..

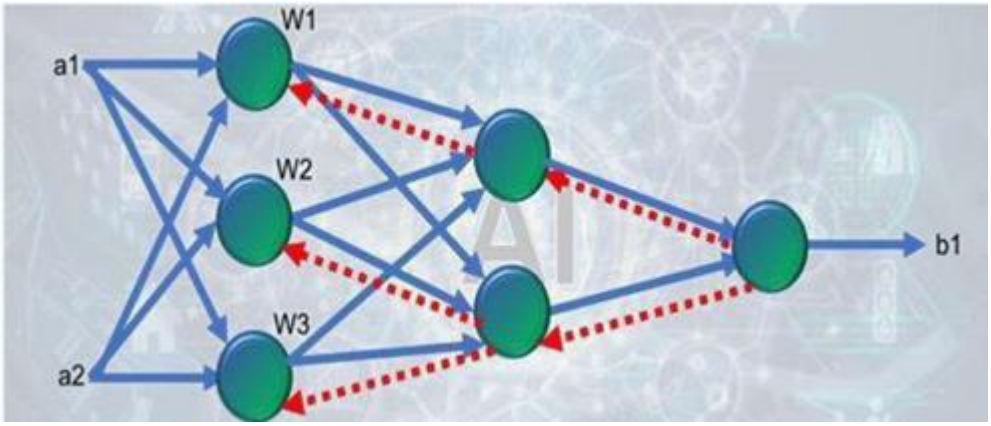


FIG 3.9 Back Propagation

3.5.3 Activation Functions

Neural networks depend on activation functions because they enable the model to recognize intricate patterns in the input. Without activation functions, the model would simply be a linear regression model, no matter how many layers it has. Let's explore the key activation functions in more detail:

1. **Sigmoid Function:**
    - **Mathematical Expression:**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

    - **Characteristics:** Produces anx S-shaped curve and outputs values between 0 and 1. It is useful for binary classification tasks where the output can be interpreted as probabilities.
    - **Limitations:** The gradients tend to vanish as the function saturates at 0 or 1, which can slow down the training of deep networks.

2. **Rectified Linear Unit (ReLU) Function:**
    - **Mathematical Expression: f(x)=max(0,x)**
    - **Characteristics:** Introduces non-linearity by outputting zero for negative inputs and the input itself for positive values. ReLU accelerates convergence in training and is computationally efficient.
    - **Limitations:** ReLU can cause the "dying ReLU" problem where neurons permanently output zero if they get stuck in the negative region.

3. **Hyperbolic Tangent (tanh) Function:**
    - **Mathematical Expression:**

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

    - **Characteristics:** Outputs values between -1 and 1, making it zero-centered. This can lead to faster convergence compared to the sigmoid function.
    - **Limitations:** Similar to sigmoid, tanh can suffer from vanishing gradients at its extremities.

4. **SoftMax Function:**
    - **Mathematical Expression:**

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

      For each class i
    - **Characteristics:** Converts logits to probabilities that sum to one, making it ideal for multi-class classification problems.
    - **Limitations:** Can be computationally expensive for models with a large number of classes due to the exponentiation and normalization steps.

3.5.4 Training:

Deep learning neural networks are designed to learn how to map inputs to outputs. This is accomplished by adjusting the weights of the network in response to the errors made by the model on the training dataset. These adjustments are made continuously to minimize the error until the learning process either comes to a stop or an acceptable level of accuracy is achieved. In other words, the goal of the network is to continually refine its mapping function to produce more accurate and precise results.

The optimization problem in deep learning neural networks is typically solved using the stochastic gradient descent algorithm. In each iteration, this approach updates the model's parameters by using the backpropagation process. Stated differently, the stochastic gradient

descent algorithm is in charge of modifying the network's weights in accordance with the errors determined during the backpropagation procedure.

Using examples, a neural network model learns how to translate a specific set of input variables to the output variable. Making ensuring this mapping performs well on both fresh, untested cases and the training dataset is the aim. The ability of the model to generalize refers to its capacity to perform well on both novel and specific examples. In essence, the model needs to be able to use what it has learnt to generate accurate results when applied to fresh, unseen data. A deep learning model must be trained using a number of procedures and fundamental ideas to guarantee that it learns efficiently:

1 **Data Preparation:**

- Normalization: To increase convergence during training, scale input features to a standard range (such as [0, 1] or [-1, 1]). Applying modifications to enhance.
- Augmentation is the process of adding variation to training data and avoiding overfitting by using transformations including rotation, scaling, and flipping

2 **Weight Initialization:**

- It is essential to initialize weights correctly to avoid problems like vanishing or ballooning gradients. He initialization and Xavier (Glorot) initialization are popular techniques.

3 **Forward Propagation:**

- The input data is passed through the network layer by layer, with each layer applying its weights and activation function to produce an output.

4 **Loss Function:**

- **Mean Squared Error (MSE):** Used for regression tasks, it measures the average squared difference between predicted and actual values.
- **Cross-Entropy Loss:** Used for classification tasks, it measures the difference between the predicted probability distribution and the actual distribution.

5 **Backward Propagation:**

- The gradients of the loss function with respect to the weights are computed using the chain rule. These gradients are used to update the weights in the opposite direction of the gradient (gradient descent).

6 **Optimization Algorithms:**

- **Stochastic Gradient Descent (SGD):** Updates weights using a subset of the training data, providing faster convergence.
- **Variants of SGD:** Include momentum, AdaGrad, RMSprop, and Adam, each improving upon SGD by adapting learning rates and incorporating momentum to speed up convergence.

7 **Regularization Techniques:**

- **L1/L2 Regularization:** Adds a penalty to the loss function proportional to the absolute value or square of the weights, helping to prevent overfitting.
- **Dropout:** To avoid co-adaptation and enhance generalization, neurons are randomly dropped during training.

3.5.4.1 Test Loss:

In machine learning, test loss is a frequently used metric to assess how well a trained model performs on a dataset that wasn't used for training. It calculates the discrepancy between a set of input data in the test dataset and the model's projected and actual outputs.

A loss function that contrasts the model's anticipated and actual outputs for each input in the test dataset is used to calculate the test loss. The kind of problem being solved determines the loss function that is employed.

1.**Loss Function:** The test dataset is subjected to the same loss function that was used for training.

2. **Evaluation:** The average error (loss) is calculated by comparing the model's predictions on the test data with the actual labels.

3. **Goal:** Better generalization and a model that is more likely to function well in real-world situations are indicated by a smaller test loss..

3.5.4.2 Test Accuracy:

One statistic for assessing how well a machine learning model performs on a test dataset is test accuracy. It calculates the proportion of accurate predictions the model produced using the test dataset some other essential statistic, specifically for category problems, is take a look at accuracy. It gives a clear indicator of the model's efficacy:

1. The calculation entails dividing the total wide variety of forecasts by way of the number of accurate guesses.

2. Interpretation: even as bad take a look at accuracy points to overfitting or underfitting, excessive take a look at accuracy shows that the version can generalize nicely to new data.

3. Balanced Datasets: Metrics like as F1 score, precision, and remember may additionally provide more data approximately imbalanced datasets than accuracy on my own.

check accuracy is determined through evaluating the model's overall performance on a validation dataset after it has been trained on a education dataset. as soon as the model has been educated and tuned to perform properly on the validation dataset, it is examined on a one-of-a-kind take a look at dataset to determine how properly it performs on new, untested records. The test accuracy is calculated as the percentage of effectively expected labels within the take a look at dataset.

3.5.4.3 Validation Loss:

A model is usually taught using a training dataset, and its performance is assessed using a different validation dataset during the training phase. Overfitting, which happens when a model performs well on training data but badly on fresh data, is avoided by using the validation dataset to track the model's performance. By applying a loss function, like mean squared error or cross-entropy, to the validation dataset, the model's validation loss is calculated. The model's parameters are then modified to enhance performance using the validation loss. Validation loss offers an early warning sign of overfitting and aids in tracking the model's performance during training:

1. Evaluation: The model is assessed on the validation set following each epoch.

2. Adjustment: When determining when to end training, the validation set's loss serves as a guidance for hyperparameter tuning.

3. Early Stopping: To avoid overfitting, training is stopped if the validation loss does not get better after a predetermined number of epochs

.

3.5.4.4 Validation Accuracy:

A model is normally taught the use of a training dataset, and its performance is classed the use of a unique validation dataset in the course of the schooling section. Overfitting, which takes place while a model plays nicely on training facts but badly on sparkling information, is averted by using the usage of the validation dataset to track the version's overall performance. via testing the version at the validation dataset and calculating the proportion of accurate predictions, validation accuracy is calculated.

just like validation loss, validation accuracy concentrates at the correct prediction charge:

1. tracking: it is used to compare numerous fashions or hyperparameter settings and aids in measuring the version's mastering development.

2. **version choice:** typically, the model chosen for final assessment on the check set is the one with the quality validation accuracy.

3.5.5 Train Dataset:

a set of information known as the education records is used to train a model to apprehend developments and connections among enter and output variables. The version's parameters are modified using the education records to enable it to supply precise predictions on fresh, untested records. The studying system is built upon the schooling dataset. Its representativeness and nice have a huge have an effect on on how properly the model works:

1. range: A varied training dataset complements the version's potential to generalize via enabling it to find out a greater diversity of patterns.

2. size: although diminishing returns might also arise beyond a positive factor, larger datasets generally permit for higher getting to know.

3. Label exceptional: For supervised getting to know tasks, specific and reliable labeling is essential. misguided or biased version predictions may also end result from negative labeling

.

3.5.6 Testing:

within the context of device learning fashions, "checking out" generally way assessing the version's precision or accuracy. This isn't how the phrase is used in traditional software program improvement.

inside the context of system learning, testing involves studying the model's performance the use of a check dataset:

1. purpose: gives an objective assessment of the model's capacity to extrapolate to novel facts.

2. procedure: Metrics such as take a look at accuracy and test loss are calculated when the model is administered at the test set.

3. comparison: To make sure the version continuously works well on unknown data, the outcomes are in comparison to validation measures.

4. final evaluation: to provide a very last performance estimate, the take a look at dataset is frequently utilized just as soon as following model choice and adjustment

.

CHAPTER 4

SOFTWARE IMPLEMENTATION

4.1 Python:

Guido van Rossum to begin with posted Python, a high-level, interpreted programming language, in 1991. it's miles a popular option for each novice and pro programmers because of its easy-to-study and write layout. Python's syntax is clear and simple, with a focus on readability. Python's giant module library, which permits developers to perform a spread of obligations without having to start from scratch when writing code, is one in every of its primary strengths. Python is renowned for its adaptability, with uses spanning from information evaluation and net improvement to machine gaining knowledge of and medical computing.

Because of its vibrant developer community and the contributions made to its open-source codebase, Python has continuously increased in popularity over time. Python is now regarded as one of the most frequently used programming languages worldwide and is utilized extensively in government, business, and academia. Python is a popular choice for scripting and automation jobs, as well as web development utilizing frameworks like Django and Flask, and it is compatible with a variety of platforms, including Windows, Linux, and macOS. Python is still a preferred option for developers and companies alike because of its robust and adaptable syntax, extensive module library, and vibrant community.

4.2 Jupyter NoteBook:

customers can create and proportion files with live code, equations, graphics, and narrative text the usage of the open-source internet application Jupyter notebook. Python, R, and Julia are a number of the extra than forty programming languages it helps. users can write and run code in cells in the interactive computing surroundings that Jupyter pocket book gives. Code, markdown textual content, or uncooked text may be found in each cell. users may also view the outcomes of their code immediately for the reason that output of a code cell is shown without delay underneath the cell.

additionally, Jupyter pocket book facilitates the development of interactive visualizations through the usage of tools like Matplotlib, Bokeh, and Plotly. because it gives an effective and adaptable surroundings for statistics exploration, visualization, and evaluation, Jupyter pocket book is frequently used in facts technological know-how and scientific computing. Its interactive functions allow for short improvement and experimentation, and its capability to combine narrative prose, facts, and code makes it a beneficial device for teamwork and communique.

4.3 Libraries:

4.3.1 Open Source-Computer Vision Library:

A complete open-supply library for device learning, image processing, and pc imaginative and prescient is referred to as OpenCV. For modern-day systems, its packages are vital to actual-time operations. It allows forty three AI Writing Submission Submission id trn:oid:::26066:456626117, web page forty five of 78 AI Writing Submission Submission identity trn:oid:::26066:456626117, page forty six of seventy eight the recognition of individuals, gadgets, and handwriting in photographs and movies. Python can manner pics through reading the OpenCV array structure together with different tools like NumPy. Vector space is used to find visual patterns and their unique features, and mathematical operations are performed on these features.

OpenCV was first released in version 1.0 and is freely usable for both academic and commercial applications under a BSD license. OpenCV is compatible with a number of operating systems, including Mac OS, Windows, Linux, iOS, and Android, and features interfaces for Java, Python, C++, and C. Because supporting real-time applications was its main goal, it was constructed with C/C++ code that was tuned to benefit from multi-core processing.

.

OpenCV Functionality:

- OpenCV is capable of reading and writing image and video files in a number of different formats, including JPEG, PNG, BMP, and MPEG.
- Image processing: OpenCV offers several image processing features, including morphology, edge detection, thresholding, filtering, and many more.
- Feature extraction and detection: OpenCV has methods for identifying and removing many types of features from pictures, like lines, blobs, and corners.
- OpenCV offers a number of methods for object detection and recognition, including face detection, pedestrian detection, and object recognition..

Applications of OpenCV:

Many applications are solved with OpenCV; a few of these are mentioned below.

- OpenCV is frequently utilized for object identification and recognition in a variety of domains, including robotics, security, and surveillance.
- It can be used, for instance, to identify faces, cars, and people in live video broadcasts.
- OpenCV finds applicability in medical imaging for tasks including picture segmentation, classification, and analysis.

- It can be used to track a robot's location within a certain environment or for the purposes listed below**.**

4.3.2 Numerical Python :

A well-known Python package for scientific and numerical computation is called NumPy, short for Numerical Python. It has robust array computing capabilities as well as a broad 44 AI Writing Submission variety of mathematical functions for working with arrays, matrices, and other numerical data structures of the

Similar to a list or a Python array, NumPy offers an array object with extra capabilities including quick and effective broadcasting, slicing, and indexing. moreover, NumPy arrays are homogeneous, because of this that every element in an array should be of the same statistics kind. This permits extra powerful computation and reminiscence allocation. severa mathematical functions, together with primary mathematics operations, linear algebra, Fourier transforms, random number technology, and greater, are available in NumPy in addition to arrays. moreover, it really works nicely with other libraries for scientific computing, consisting of Pandas, Matplotlib, and SciPy.

4.3.6 OS module in Python:

Python's OS module offers a way of speaking with the running machine that the Python interpreter is executing on. It gives access to device records, method control, and record and listing management functions. further to performing shell instructions, accessing surroundings variables, and producing baby processes, the module has functions for adding, putting off, renaming, and transferring files and directories. features for working with document paths and enhancing report and directory names are to be had inside the os.route submodule. An powerful tool for speaking with the running gadget from inner a Python application is the OS module. It permits programmers to jot down scripts that may automate routine operations like process control, gadget administration, and record and directory control.
some of submodules that supply in addition functionality for coping with the working machine are to be had within Python's os module. the subsequent are a number of the OS module's fundamental submodules
The os module in Python provides numerous submodules that provide extra functionality for interacting with the running system. here are a number of the key submodules of the os module:

- **os.path:** This submodule provides functions for working with file paths and manipulating file and directory names. Some of the functions in this submodule include.
- **os.system:** This submodule provides functions for executing shell commands from within a Python script. To run a command in the shell and get the result, use the os.machine() function.

- **os.environ:** This submodule provides access to the environment variables on the system. The os.environ dictionary contains key-value pairs for each environment variable.
- **os.fdopen:** This submodule shows how to use the file object interface to open file descriptors. A file object that can be read and written to is returned by the os.fdopen() function after receiving a file descriptor.

4.3.7 Dlib:

A well-favored C++ library for growing laptop imaginative and prescient and device cutting-edge packages is known as Dlib. Researchers and developers inside the quarter cutting-edge use it ultra-modern its splendid performance and versatility. Dlib has ultra-modern pre-educated models for more than a few responsibilities, which includes picture segmentation, item detection, face detection, and facial landmark detection. applications can speedy reach 49a2d564f1275e1c4e633abc331547db performance by integrating these fashions comfortably. Dlib offers resources for schooling custom models in addition to pre-educated fashions. those encompass powerful optimization techniques for schooling those models in addition to aid for an expansion modern-day device latest algorithms, including SVMs, choice timber, and neural networks.

modern day software lessons and capabilities for working with matrices, pics, and different data structures brand new utilized in laptop imaginative and prescient and machine getting to knowmodern are also protected in dlib. it can be applied on quite a number working structures, which includes home windows, macOS, Linux, and Android, and is made to be go-platform transportable.
amongst DLIB's number one attributes are

Among DLIB's primary attributes are:

- Face detection and recognition: DLIB offers both individual face recognition and pre-trained models for identifying faces in photos and videos.

- Object detection: To find and identify things of interest in pictures and videos, DLIB offers training and usage tools for object detectors.

- Image segmentation: Using color, texture, and other characteristics, DLIB offers algorithms for dividing images into distinct areas.

- Machine learning: DLIB offers resources for training and utilizing a range of machine learning models, including deep neural networks, decision trees, and support vector machines (SVMs).

CHAPTER 5
DESIGN SPECIFICATION

5.1 Arduino UNO:

Designed by way of Arduino.cc, the Arduino Uno is an open-source microcontroller board based at the Microchip ATmega328P. The board's analogue and digital input/output (I/O) pins can be used to link to different circuits and expansion boards (shields). The board has 14 digital I/O pins (six of which are capable of PWM output) and 6 analogue I/O pins, and it can be programmed using the Arduino IDE and a type B USB connector (Integrated Development Environment). With voltages ranging from 7 to 20 volts, it can be powered by a USB cable or an external 9-volt battery. The Italian word "uno," which means "one," was chosen to commemorate Arduino Software's first release. The Arduino Uno is the first of a series of Arduino boards powered by USB. Version 1.0 of the Arduino IDE was the reference version of Arduino, which has since been updated. You can upload new code to the ATmega328 on board boot loader without requiring an external hardware programmer.

The Arduino Uno is a widely-used microcontroller board created by Arduino.cc, an open-source electronics platform. It is designed around the Microchip ATmega328P microcontroller. The Arduino Uno is highly popular in the maker community and is used for a variety of electronics projects, from simple to complex.

Key Features:

- **Input/Output Pins:**
    - *i* **Digital I/O Pins:** The board features 14 digital input/output pins. These pins can be used to read digital signals (on/off states) from sensors or to control devices like LEDs and motors. Out of these 14 pins, 6 can output PWM signals, which are used for tasks like controlling the speed of motors or dimming LEDs.
    - *ii* **Analog Input Pins:** It has 6 analog input pins, which can read signals from analog sensors, like temperature or light sensors, providing a range of values instead of just on/off.
- **Programming:**
- **Arduino IDE:** The Arduino incorporated improvement surroundings (IDE) is used to application the Arduino Uno. This software program, that's loose to download, is

renowned for being honest and simple to apply. you could write code (called sketches) inside the IDE, assemble it, and then add it to the board..

- **USB Connection:** A type B USB connection, which is likewise used to upload code to the microcontroller, hyperlinks the board to your computer.
- **Power Supply:**
- **USB Power:**For ease of development and checking out, the Arduino Uno may be powered through your laptop through a USB connection.
- **External Power**:An outside electricity source, like a 9-volt battery or an outside strength adapter, can also be used to electricity it. The board gives a variety of power options by helping enter voltages among 7 and 20 volts

- **Bootloader:** A bootloader is pre-mounted at the Arduino Uno's ATmega328P microprocessor. This bootloader removes the want for an additional programmer by way of permitting you to upload new code instantly over the USB connection. it is extraordinarily easy to apply and update because to this capability.

Applications:

- The Arduino Uno is utilized in many extraordinary applications and is pretty versatile. ordinary applications include: academic tasks: The Arduino Uno is a famous desire for educational purposes because of its simplicity and ease of use. schools and faculties often put it to use to educate college students in programming and electronics. DIY tasks: From fundamental LED light presentations to more tricky robotics and home automation systems, fans make use of the Arduino Uno to build a wide variety of initiatives. The Arduino Uno is utilized by engineers and developers to prototype new electronic devices. it's far a first-rate device for developing with clean thoughts due to its versatility and large choice of well suited shields and modules. The Arduino Uno is utilized by designers and artists to encompass interactive components into art installations along with lights

FIG 5.1: Arduino Uno

5.1.1 General Pin Functions:

- **LED:** It has an LED that is powered by digital pin 13. When the pin is high, the LED is on, and when the pin is low, the LED is off.

- **VIN:** When using an external power supply, the Arduino board's input voltage (as opposed to 5 volts from a USB port or other regulated power supply). This pin can be used to supply power. Access this pin if electricity is supplied through the power jack.

- **5V:** The board's regulator provides a controlled 5V to this pin. The board can be powered by the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin (7-20V). When power is supplied via the 5V or 3.3V pins, the regulator is bypassed, and the board may be damaged.

- **3.3V:** A 3.3 volt supply is generated by the on-board regulator. 50 milliamperes is the maximum current draw.

- **GND:** Ground pins.

- **IOREF:** This pin on the Arduino board provides the voltage reference with which the microcontroller runs. A properly designed shield can read the IOREF pin voltage and then select the appropriate power source or enable voltage translators on the outputs to work with 5V or 3.3V.

- **Reset:** Used to Reset pie board.

5.1.2 Special pin functions:

Under software control (using the pinMode (), digitalWrite (), and digitalRead () routines), Uno's 14 digital pins and 6 analogue pins can be utilised as inputs or outputs. They are powered by 5 volts. As recommended operating circumstances, each pin can provide or receive 20 mA and has a 2050 kOhm internal pull-up resistor (disconnected by default). To avoid irreversible damage to the microcontroller, the I/O pins should not drain more than 40mA. A0 through A5 are the six analogue inputs on Uno. Each has a resolution of 10 bits (that is, 1024 different values). It measures from ground to 5 volts by default, however the AREF pin and the analogue Reference () function can be used to adjust the upper bound of the range.

Furthermore, several pins have specific functions:

- **Serial:** 0 (RX) and 1 (TX) (TX). TTL serial data is received (RX) and transmitted (TX) using this device. These pins are connected to the ATmega8U2 USB-to-TTL serial chip's corresponding pins.
- **External interrupts:** pins 2 and 3. On a low value, a rising or falling edge, or a change in value, these pins can be programmed to generate an interrupt.
- **PWM :** pins 3, 5, 6, 9, 10, and 11. The analog Write() method can produce 8-bit PWM output.
- **SPI** (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), and 13 (MISO) (SCK). The SPI library is used to communicate with these pins.
- **TWI** (two-wire interface) / I²C: SDA (A4) and SCL (A5) pins (A5). Wire library is used to provide TWI communication.
- **AREF** (analog reference): The analogue inputs have a reference voltage.

5.1.3 APPLICATION:

- Weighing Equipment
- Countdown Timer for Traffic Lights
- Parking Garage Counter
- Embedded software.
- Household Automation

5.1.4 PIN CONFIGURATION:

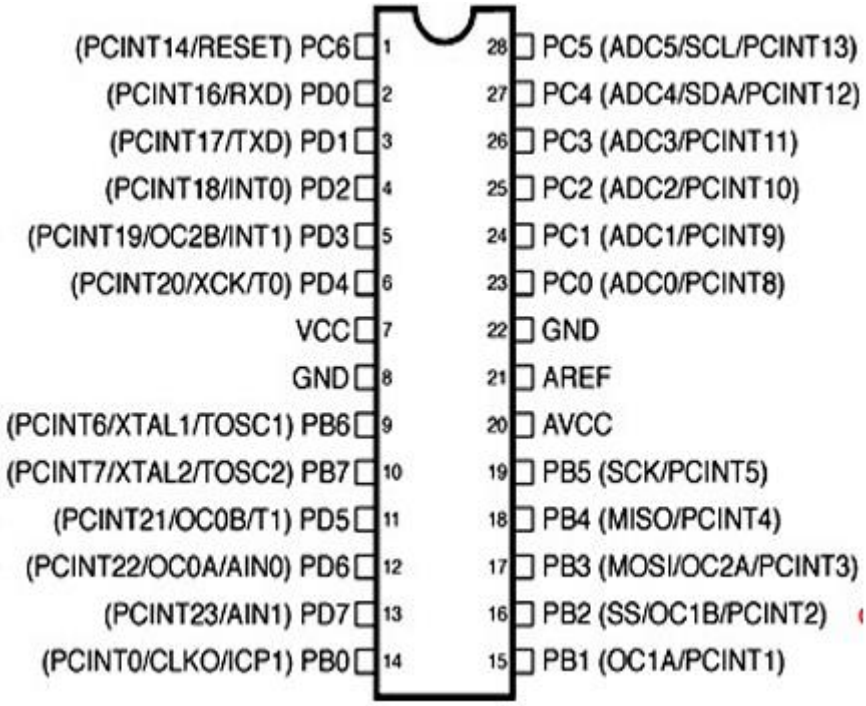


FIG 5.2: Pin Configuration

5.2 Buzzer:

A buzzer is a device that produces sound to signal or alert users to various conditions or actions. These devices can be found in numerous applications, including alarm systems, timers, and user interfaces (like confirming a mouse click or a keypress). Buzzers come in different types based on their operating principles, such as mechanical, electromagnetic, and piezoelectric (often abbreviated as "piezo").

5.2.1 Electromechanical Buzzer:

In the early days, many buzzers were designed using an electromechanical system. This design was similar to that of an electric bell but without the metal gong that typically strikes to produce sound. Instead, these buzzers utilized a relay mechanism that would interrupt its own current flow, causing the contacts to vibrate rapidly. This vibration produced the characteristic buzzing sound. To enhance the sound, these units were often mounted on walls or ceilings, which served as a sounding board. This design not only amplified the noise but also distributed it over a larger area. The term "buzzer" itself comes from the distinctive, harsh buzzing noise that these electromechanical devices generated.

5.2.2 Piezoelectric Buzzer:

Piezoelectric buzzers operate on a different principle compared to their electromechanical counterparts. A piezoelectric component, a substance that can transform electrical energy into mechanical energy (sound) and vice versa, is what they employ. This component is driven by an oscillating electronic circuit or another type of audio signal source. When an electrical signal is applied, the piezoelectric material vibrates, producing sound waves. Piezoelectric buzzers are known for their efficiency and reliability. They are commonly used in modern electronic devices to provide auditory feedback. For instance, when a button is pressed, the buzzer might emit a click, ring, or beep to indicate the action. This immediate feedback helps users understand that their input has been registered by the device.

Overall, buzzers are essential components in many electronic systems, providing a simple and effective way to communicate with users through sound.

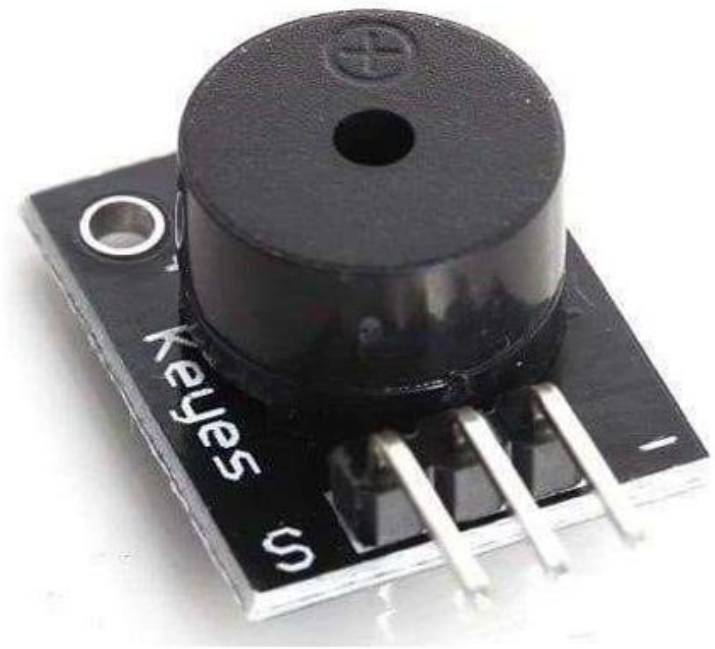


FIG 5.3: Electromechanical Buzzer

5.3 WebCam:

- A webcam is a camcorder that feeds or transfers a picture or video continuously to or through a PC to a PC organization, like the Internet.

- Webcams are commonly little cameras that sit on a work area, connect to a client's screen, or are incorporated into the equipment.
- Webcams can be utilized during a video talk meeting including at least two individuals , with discussions that incorporate live sound and video.
- Webcam programming empowers clients to record a video or transfer the video on the Internet.
- As video real time over the Internet requires a lot of transmission capacity such streams ordinarily utilize compacted designs.
- The greatest goal of a webcam is likewise lower than most handheld camcorders, as higher goals would be diminished during transmission.

The lower goal empowers webcams to be moderately reasonable contrasted with most camcorders, however the impact is sufficient for video visit meetings.

Characterstic:

- Webcams are known for their low assembling cost and their high adaptability, making them the most reduced expense type of video communication. As webcams developed all the while with show innovations, USB interface speeds and broadband web speeds, the goal went up step by step from 320*240 to 640*480, and some presently significantly offer 1280*720 or 1920*1080 goal.
- Despite the ease, the goal offered starting at 2019 is amazing, with now the low- end webcams offering goals of 720p, mid reach webcams offering 1080p goal, and top of the line webcams offering 4K goal at 60 fps.
- Webcams have become a wellspring of safety and security issues, as some implicit webcams can be distantly initiated by spyware. To address this worry, numerous webcams accompany an actual focal point cover.

USES:

- The most famous utilization of webcams is the foundation of video joins, allowing PCs to go about as videophones or videoconference stations.
- Other mainstream utilizes incorporate security observation, PC vision, video broadcasting, and for recording social recordings.

- The video transfers given by webcams can be utilized for various purposes, each utilizing proper programming.

5.3.1 Video Monitoring:

- Webcams might be introduced at spots, for example, childcare focuses, workplaces, shops and private regions to screen security and general movement.
- Webcams are known for their low assembling cost and their high adaptability, making them the most reduced expense type of video communication. As webcams advanced all the while with show innovations, USB interface speeds and broadband web speeds, the goal went up step by step from 320*240 to 640*480, and some presently significantly offer 1280*720 or 1920*1080 goal.
- Despite the ease, the goal offered starting at 2019 is noteworthy, with now the lowend webcams offering goals of 720p, mid reach webcams offering 1080p goal, and very good quality webcams offering 4K goal at 60 fps.
- Webcams have become a wellspring of safety and security issues, as some inherent webcams can be distantly enacted by spyware. To address this worry, numerous webcams accompany an actual focal point cover able to be saved to the PC, messaged, or transferred to the Internet. In one very much plugged case, a PC messaged pictures of the criminal during the robbery of the PC, empowering the proprietor to give police a reasonable image of the thief's face even after the PC had been taken.

5.4 16x2 LCD Display:

A 16x2 LCD display is a widely-used electronic component that can display information in a visual format. The "16x2" designation means that the display has 16 columns and 2 rows,

allowing it to show up to 32 characters at a time. Each character is made up of a 5x8 pixel matrix, which enables the display of text and simple custom symbols.

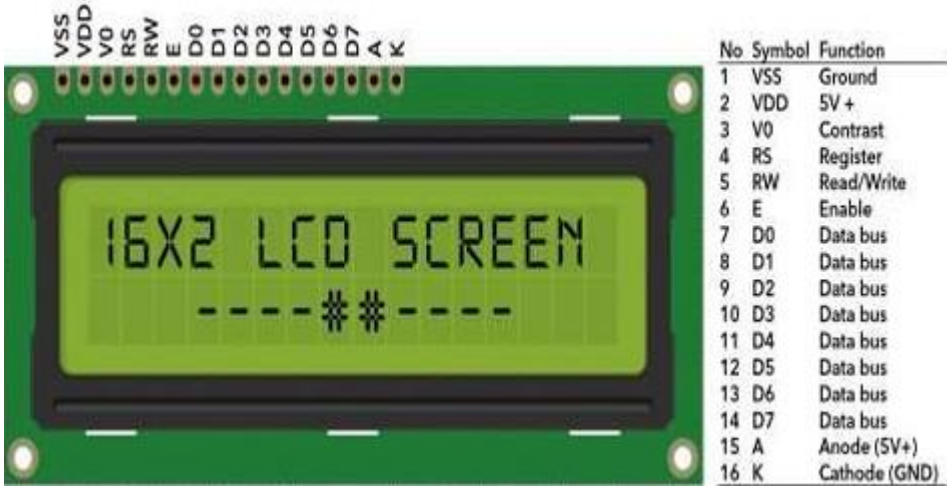


FIG 5.4: 16x2 LCD display

- **Character Display:** The 16x2 LCD can display 32 characters at once, arranged in two rows of 16 characters each. This makes it suitable for displaying short messages, data readings, and simple menus.
- **Backlight:** Most 16x2 LCDs come with a backlight that improves visibility, especially in low-light conditions. The backlight can usually be controlled programmatically.
- **Adjustable Contrast:** The contrast of the display can be adjusted using a potentiometer, allowing for better readability under different lighting conditions.
- **Ease of Interface:** These displays are commonly interfaced with microcontrollers like the Arduino Uno. They typically use a standard parallel interface, which involves multiple pins for data and control signals.

5.4.1 Pins And Connections:

A typical 16x2 LCD display has 16 pins, though not all of them need to be used in simple applications:

1. **VSS:** Ground connection.
2. **VDD:** Power supply (usually +5V).
3. **VO:** Contrast adjustment (connected to a potentiometer).
4. **RS:** Register Select (command/data selection).
5. **RW:** Read/Write mode selection.

6. **E:** Enable pin (to initiate data read/write). 7-14. **D0-D7**: Data pins (used to send 8 bit data).
7. **A (LED+):** Backlight anode (positive end).
8. **K (LED-):** Backlight cathode (negative end).

Most applications only use the essential pins: power (VSS, VDD), contrast (VO), control (RS, E), and data (D4-D7 for 4-bit mode). RW is often tied to ground to keep the display in write mode.

5.4.2 Functionality:

- **Text Display:** The primary function is to display alphanumeric characters. Each character is displayed in a 5x8 pixel matrix.
- **Custom Characters:** Users can create and display custom characters by defining their pixel patterns. This is useful for special symbols not included in the standard character set.
- **Cursor Control:** The display allows cursor control, such as moving the cursor to specific positions or blinking the cursor for text input interfaces.

5.4.3 Applications:

The 16x2 LCD display is used in various applications due to its simplicity and effectiveness in displaying information:

- **Microcontroller Projects:** Widely used in Arduino and other microcontroller projects for displaying sensor readings, system status, and user interface menus.
- **Industrial Equipment:** Used in control panels of machinery to show settings, status messages, and error codes.
- **Home Appliances:** Found in microwaves, washing machines, and other appliances to display settings and timer information.
- **Educational Tools:** Common in educational kits and DIY electronics projects to teach programming and hardware interfacing.

5.5 FLOWCHART:

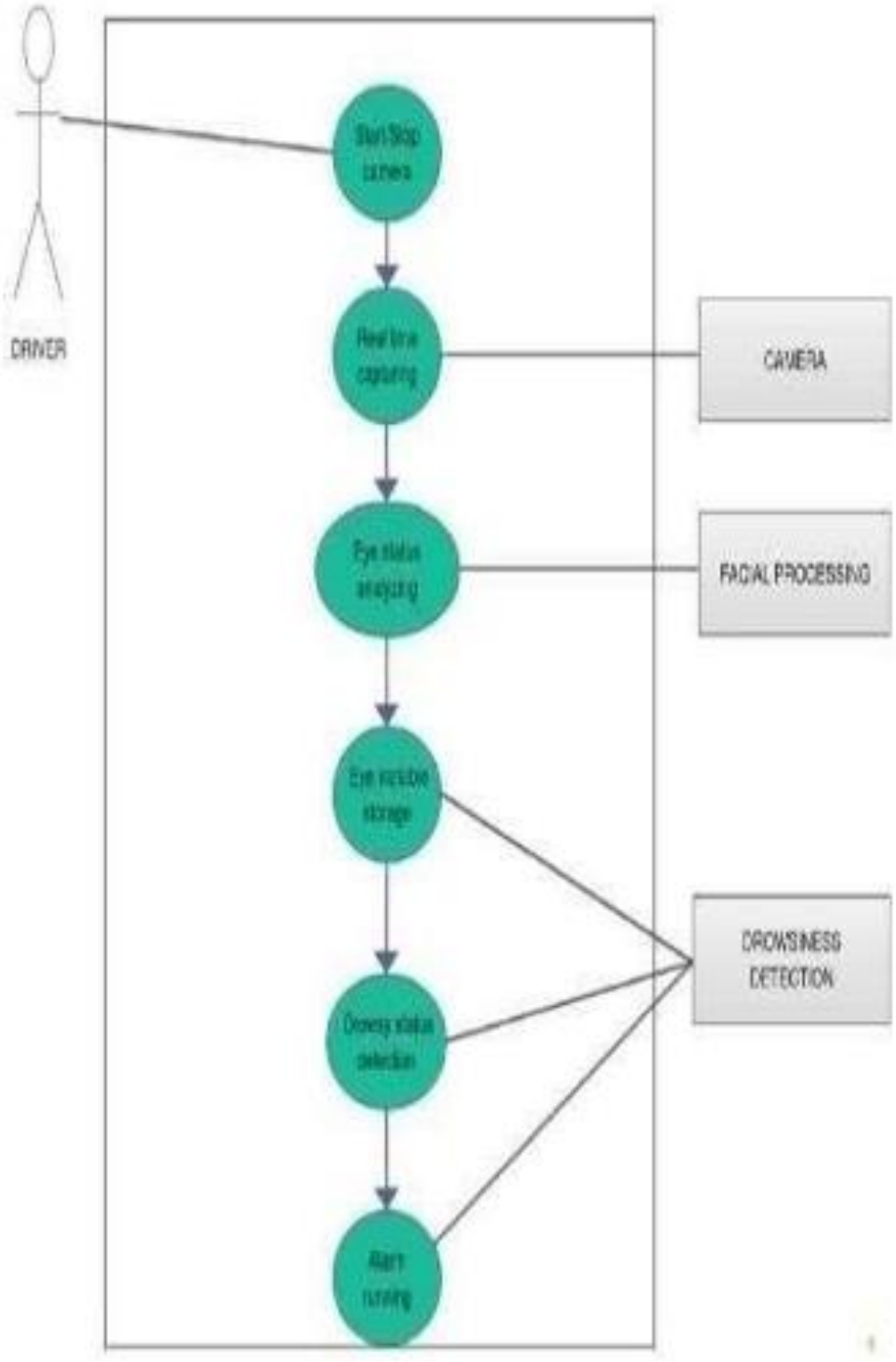


FIG 5.5: Detecting and analyzing drowsiness

CHAPTER 6

EXPERIMENTAL RESULTS

Input:

The input for system is a human face image. The system has MRL eye dataset consisting of 80,000 cropped images of eye region, and the following are driver input images which were given to detection system.

FIG 6.1: Driver Input Images

Output:

The input is captured through camera, then face tracking and detection is done through MC-KCF. Once face detection and image resizing have been carried out, the resulting images are as follows:

FIG 6.2: Alert Images

FIG 6.3: Drowsy Images

From the above image's features like eyes are extracted. The extracted features are passed through ResNet CNN and produce buzzer alert if the driver is in drowsy state.

6.1 Results through live face tracking:

6.1.1 Frames recognized as drowsy:

The system works by continuously analysing a stream of video input to detect signs of driver drowsiness. When the system detects drowsiness, it will highlight the driver's eyes using a red rectangle and emit a beep sound as an alert to make the driver aware of their drowsiness. The detection of drowsiness is achieved by utilizing a well- trained ResNet Convolutional Neural Network (CNN) model.

FIG 6.4: Person 1 in drowsy state

FIG 6.5: Person 2 in drowsy state

FIG 6.6: Person 3 in drowsy state

FIG 6.7: Person 4 in drowsy state

FIG 6.8: Person 5 in drowsy state

By observing above figures in this section, the system detects eye region in whole face expression and as the eyes are closed, the detecting system alerts driver through buzzer sound.

6.1.2 Frames recognised as not drowsy:

The system operates by continuously analysing a live video stream to determine if the driver is not drowsy. If the system detects that the driver is not drowsy, it will highlight their eyes using a green-coloured rectangle and will not generate an alert. The detection of driver drowsiness is accomplished using a well-trained ResNet Convolutional Neural Network (CNN) model.

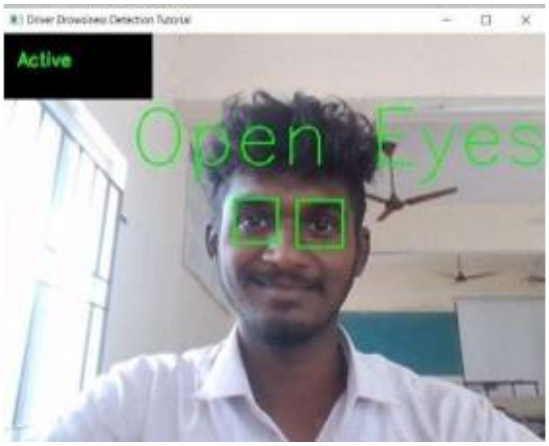


FIG 6.9: Person 1 in alert state

FIG 6.10 Person 2 in alert state

FIG 6.11: Person 3 in alert state

FIG 6.11: Person 4 in alert state

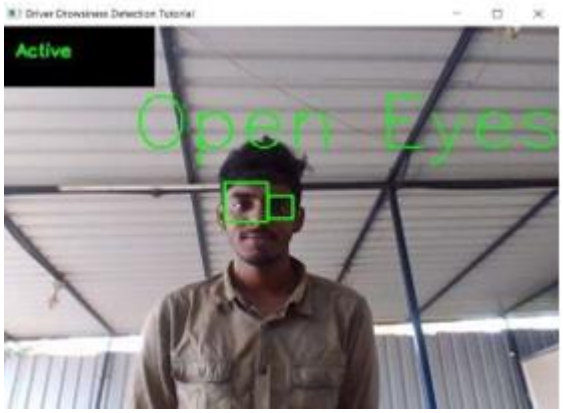


FIG 6.12: Person 5 in alert state

By observing above figures in this section, the system detects eye region in whole face expression and as the eyes are open, the detecting system states that the driver is in active mode i.e., eyes are not closed.

Results on test data:

Test loss: 0.0183

Test accuracy: 0.9817

Validation loss:0.0167

Validation accuracy:0.9833

In general, test loss is calculated by evaluating dataset which was not used during training. A lower test loss indicates that the model is more accurate and better at generalizing to new data.

The test accuracy is calculated as the percentage of correctly 63 predicted labels in the test dataset. Validation loss is computed by evaluating the model on the validation dataset using a loss function, such as mean squared error or cross- entropy. The goal of training a model is to minimize the validation loss, which indicates that the model is becoming more accurate at predicting outputs for new, unseen data.

Validation accuracy is computed by evaluating the model on the validation dataset and measuring the percentage of correct predictions. The goal of training a model is to maximize the validation accuracy, minimize the validation loss, which indicates that the model is becoming more accurate at predicting outputs for new, unseen data. From the above test results, model contains minimum test loss and validation loss and maximum validation accuracy, which states that model is well trained and produces better results than previous models in detection

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

3.6 Conclusion:

This project proposes a new method for driver drowsiness detection using a combination of MC-KCF and ResNet CNN. The proposed method utilizes the advantages of both methods to track eye movements using MC-KCF and classify the driver's face to determine the drowsiness through ResNet CNN in real-time.

Experimental results demonstrated that the proposed method produced high accuracy rates in detecting driver drowsiness based on testing results and through Resnet CNN model, when driver is in drowsy state upto 1sec, the alarm sound is produced and alerts the driver stating that he/she is in drowsy state. The proposed method has the potential to be used in various settings, including in-vehicle systems, to provide real time feedback to drivers and prevent accidents caused by driver fatigue.

This project also contributes to the field of driver drowsiness detection by providing an overview of existing methods. However, there are few limitations to the proposed algorithm, such as camera placement and light conditions, variability in eye movements, computational complexity but the proposed algorithm work can be extended to other related fields, such as human-computer interaction, facial expression recognition and emotion detection.

Thus, in this project, algorithm for fatigue detection using MC-KCF and ResNet CNN is successfully designed and executed. Overall, this project paper provides a promising approach to driver drowsiness detection that can help improve road safety and prevent accidents caused by driver fatigue.

3.7 Future Scope:

The future scope for driver drowsiness detection using MC-KCF and Residual Neural Networks is vast and promising. Here are a few potential areas of development and improvement for this project:

Real-time application: Currently, the implementation of this project requires 65 processing video frames offline, which may not be practical for real-time application. Future work could focus on optimizing the algorithms and hardware to enable real time processing of video streams, which would make the system more practical and effective for real-world scenarios.

Multi-modal input: While the current implementation of the project relies solely on visual cues to detect drowsiness, incorporating other modalities, such as audio or physiological signals, could improve the accuracy and robustness of the system.

Personalization: People exhibit different signs of drowsiness, and some individuals may exhibit unique cues that are not captured by the current algorithm. Future work could explore how to personalize the drowsiness detection algorithm to individual drivers to improve its accuracy and effectiveness.

REFERENCES

[1]. Snehal S.Bharambe and P.M.Mahajan, “Implementation of Real Time Driver Drowsiness Detection System” IJSR, vol 4 issue1, Jan 2015.

[2]. Wanghua Deng and Ruoxue wu, “Real-Time Driver-Drowsiness Detection System Using Facial Features” IEEE access, vol 7,2019.

[3]. Pratyush Agarwala and Rizul Sharma, “Driver Drowsiness Detection Techniques: Review” EasyChair, Dec 21,2019.

[4]. Kusuma Kumari B.M, “Review on Drowsy Driving:Becoming Dangerous Problem” IJSR, vol 3 issue 1, Jan 2014.

[5]. Tejal Jadhav, Siddhi Gajare, Shubhman Salve and Prof. Hani Patil, “Advanced Driver Assistance System for drivers using Machine Learning and Artificial Intelligence Techniques” IJRASET, vol 10 issue V, May 2022.

[6]. V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N and Dr. Nagamani N P, “Driver Drowsiness Detection” IJERT, vol 8 issue 15,2019.

[7]. Jagbeer Singh, Ritika Kanokia, Rishika Singh, Rishita Bansal and Sakshi Bansal, “Driver Drowsiness Detection System-An approach by Machine Learning Application”, Journal of Pharmaceutical Negative Results, vol 13, special issue 10, 2022.

[8]. Whui Kim, Kyong Hee Lee, Hyun Kyun Choi and Byung Tae Jan, “A Study on Feature Extraction Methods Used to Estimate a Driver’s Level of Drowsiness”, IEEE, February 2019[2019 21st International Conference on Advanced Communication Technology (ICACT)].

[9]. Tianyi Hong, Huabiao Qin, “Drivers Drowsiness Detection in Embedded System.”, IEEE, December 2007[2007 IEEE International Conference on Vehicular Electronics and Safety].

[10]. Dwipjoy Number Sarkar and Atanu C, “Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection”, IJETT, Volume 10 9, April 2014.

[11]. Fouzia, Roopalakshmi R, Jayantkumar A Rathod, Ashwitha S, Supriya K, “Driver Drowsiness Detection System Based on Visual Features.” , IEEE, April 2018[2018 67 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)].

[12]. SaeidFazli, Parisa Esfehani, “Tracking Eye State for Fatigue Detection”, ICACEE, November 2012.

[13]. Shruti Wasudeo Deolikar, “ Drowsy Driving Safety Detection”, IRJMETS, vol 05, issue 01, Jan 2023.

[14]. Gao Zhenhai, Le DinhDat, Hu Hongyu, Yu Ziwen and Wu Xinyu, “Driver Drowsiness Detection Based on Time Series Analysis of Steering Wheel Angular Velocity”, IEEE, January 2017[2017 9th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)].

[15]. Seth Weidman, “Deep Learning from Scratch”, O’Reilly ,2019.

[16]. Eli Stevens, Luca Antiga, and Thomas Viehmann, “Deep Learning with PyTorch”, Manning,Jul-2020.

[17]. Aleksandar Čolić , Oge Marques and Borko Furht, “ Driver Drowsiness Detection- Systems and Solutions”, Spring,2014.

[18]. Shehzad Saleem, “Risk assessment of road traffic accidents related to sleepiness during driving: a systematic review”, Eastern Mediterranean Health Journal,vol 28, issue 9, 2022.

[19]. Z Tian and H Qin, “Real-time driver's eye state detection. Proceedings of the IEEE International Conference on Vehicular Electronics and Safety, October 2005.

[20]. Marco Javier Flores, José María Armingol and Arturo de la Escalera, “ Real-Time Drowsiness Detection System for an Intelligent Vehicle”, IEEE, June 2008[2008 IEEE Intelligent Vehicles Symposium].

[21]. Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng and Rong Qu, “ A Survey of Deep Learning-based Object Detection”, IEEE access, vol 7,2019.

[22]. Kaiming He, Xiangyu Zhang, “Deep Residual Learning for Image Recognition”,ILSVRC,2015.

[23]. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, ‘‘High-speed tracking with kernelized correlation filters,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015.

[24]. Y. Zhang and C. Hua, ‘‘Driver fatigue recognition based on facial expression analysis using local binary patterns,’’ Optik, vol. 126, no. 23, pp. 4501–4505, Dec. 2015.

[25]. R. O. Mbouna, S. G. Kong, and M.-G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1462–1469, Sep. 2013.

[26]. International Organization of Motor Vehicle Manufacturers. Provisional Registrations or Sales of New Vehicles. Available: http://www.oica.net/wp-content/uploads/

[27]. Wards Intelligence. World Vehicles in Operation by Country, 2013– 2017. Available: http://subscribers.wardsintelligence.com/databrowse-world.

[28]. National Highway Traffic Safety Administration. Traffic Safety Facts 2016. Available: https://crashstats.nhtsa.dot.gov.

APPENDIX

Python Code

```
#Importing OpenCV Library for basic image processing functions
import cv2
# Numpy for array related functions
import numpy as np
# Dlib for deep learning based Modules and face landmark detection
import dlib
#face_utils for basic operations of conversion
from imutils import face_utils
import serial
import time
s = serial.Serial('COM12',9600)
#Initializing the camera and taking the instance
cap = cv2.VideoCapture(0)
#Initializing the face detector and landmark detector
hog_face_detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
#status marking for current state
sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)
def compute(ptA,ptB):
        dist = np.linalg.norm(ptA - ptB)
```

```
 return dist
def blinked(a,b,c,d,e,f):
    up = compute(b,d) + compute(c,e)
    down = compute(a,f)
    ratio = up/(2.0*down)
    #Checking if it is blinked
    if(ratio>0.25):
        return 2
    elif(ratio>0.21 and ratio<=0.25):
        return 1
    else:
        return 0
while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = hog_face_detector(gray)
    #detected face in faces array
    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()
        face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)
```

```
#The numbers are actually the landmarks which will show eye
left_blink = blinked(landmarks[36],landmarks[37],
landmarks[38], landmarks[41], landmarks[40], landmarks[39])
right_blink = blinked(landmarks[42],landmarks[43],
landmarks[44], landmarks[47], landmarks[46], landmarks[45])
#Now judge what to do for the eye blinks
if(left_blink==0 or right_blink==0):
        sleep+=1
        drowsy=0
        active=0
        if(sleep>6):
                s.write(b'a')
                time.sleep(2)
                status="SLEEPING !!!"
                color = (0,0,255)
elif(left_blink==1 or right_blink==1):
        sleep=0
        active=0
        drowsy+=1
        if(drowsy>6):
                s.write(b'a')
                time.sleep(2)
                status="Drowsy !"
                color = (0,0,255)
else:
```

```
                drowsy=0
                sleep=0
                active+=1
                if(active>6):
                    s.write(b'b')
                    time.sleep(2)
                    status="Active :)"
                    color = (0,0,255)
            cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX, 1.2,
color,3)
        for n in range(0, 68):
            (x,y) = landmarks[n]
            cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)
    cv2.imshow("Frame", frame)
    #cv2.imshow("Result of detector", face_frame)
    key = cv2.waitKey(1)
    if key == 27:
        break
```

APPENDIX A2

ARDUINO UNO CODE

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
const int buzzer_Pin = 7;
const int led_Pin = 9;
char sleep_status = 0;

void setup() {
  Serial.begin(9600);
  pinMode(buzzer_Pin, OUTPUT);
  pinMode(led_Pin, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("DriverDrowsiness");
  lcd.setCursor(0,1);
  lcd.print("Detection System");
  digitalWrite(buzzer_Pin, LOW);
  digitalWrite(led_Pin, LOW);

  // Initial startup sequence as in old code
  delay(500);
  digitalWrite(buzzer_Pin, HIGH);
  delay(500);
  digitalWrite(buzzer_Pin, LOW);
  delay(500);
}

void loop()
{
    while (Serial.available() > 0)
  {
    sleep_status = Serial.read();
```

```
    if(sleep_status == 'a')
    {
       lcd.clear();
       lcd.print("Please wake up");
       digitalWrite(buzzer_Pin, HIGH);
       digitalWrite(led_Pin, HIGH);
       delay(2000);
       digitalWrite(buzzer_Pin, LOW);
       digitalWrite(led_Pin, LOW);

       delay(100);
    }
    else if(sleep_status == 'b')
    {
       lcd.clear();
       lcd.print("All Ok");
       lcd.setCursor(0,1);
       lcd.print("Drive Safe");
       digitalWrite(buzzer_Pin, LOW);
       digitalWrite(led_Pin, LOW);
       delay(2000);
    }
    else
    {
      /* Do Nothing */
    }
  }
}
```