# HELMET DETECTION AND LICENSE PLATE RECOGNITION

**Major Project Report**

Submitted in partial fulfillment for the award of the degree of

### BACHELOR OF TECHNOLOGY

**In**

### COMPUTER SCIENCE and ENGINEERING

### By

|  |  |
| --- | --- |
| KARANAM JYOTHIKA | 19L31A05N9 |
| GUMMALA JYOTHI SHANKAR SWARUP | 19L31A05M0 |
| PUSAPATI MONITH CHAITANYA VARMA | 19L31A05I5 |
| VAJRAPU RAJA VENKATA GUNNESWARA GUPTA | 19L31A05J1 |

### Under the Guidance of

#### Dr.N. Thirupathi Rao PROFESSOR



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VIGNAN’S INSTITUTE OF INFORMATION TECHNOLOGY** (Autonomous)

Affiliated to JNTU Kakinada & Approved by AICTE, New Delhi Re-Accredited by NBA & NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2015, ISO 14001:2015, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM – 530 049

#### VIGNAN’S INSTITUTE OF INFORMATION TECHNOLOGY

**Department of Computer Science and Engineering**



## CERTIFICATE

This is to certify that the project report entitled **“HELMET DETECTION AND LICENSE PLATE RECOGNITION”** the bona fide record of project work carried out under my supervision by **KARANAM JYOTHIKA(19L31A05N9),GUMMALA JYOTHI SHANKAR SWARUP(19L31A05M0),PUSAPATI MONITH CHAITANYA VARMA(19L31A05I5), VAJRAPU RAJA VENKATA GUNNESWARA GUPTA(19L31A05J1)**  during the academic year 2019-2023, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University, Kakinada. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

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

Mr. Dinesh Reddy Dr.N. Thirupathi Rao

ASSOCIATE PROFESSOR PROFESSOR

## DECLARATION

We hereby declare that the project report entitled “**HELMET DETECTION AND LICENSE PALTE RECOGNITION**”, has been written by us and has not been submitted either in part or whole for the award of any degree, diploma or any other similar title to this or any other university.

|  |  |  |  |
| --- | --- | --- | --- |
| KARANAM JYOTHIKA | 19L31A05N9 | | 19L31A05S7 |
| GUMMALA JYOTHI SHANKEAR SWARUP | 19L31A05M0 |
| PUSAP MONITH CHAITANYA VARMA | 19L31A05I5 |
| VAJRAPU RAJA VENKATA GUNNESWARA GUPTA | 19L31A05J1 |

Date:

Place: Visakhapatnam

## ACKNOWLEDGEMENT

We consider it as a privilege to thank all those people who helped us a lot for successful completion of the project “**REAL-TIME OBJECT DETECTION WITH DISTANCE AND VOICE ALERT SYSTEM FOR IMPAIRED PEOPLE**”**.**

First of all we would like to thank our project guide **Mrs. Nishchala Thakur, Professor, Department of Computer Science & Engineering** for helping us a lot in completing our project work and for enlightening us with constructive suggestions for solving our problems patiently and helping us to improve the quality of work.

We would like to thank our ever-inspiring **Head of the Department of Computer Science and Engineering, Mr. B. Dinesh Reddy,** for his spontaneous response to every request though he is busy with his hectic schedule of administration and teaching.

We would like to acknowledge with much appreciation the crucial role of our **Project Co-Ordinator, Mr. Ch. Sekhar, Associate Professor**, **Department of Computer Science & Engineering** for helping us a lot in completing our project through the academic year. We just wanted to say thank you for being such a wonderful educator as well as a person.

We would like to thank **Principal** of Vignan’s Institute of Information Technology

**Dr. B. Arundhati,** for her encouragement to us during the course of this project work.

We also express our sincere thanks to our beloved **Chairman** of Vignan’s Institutions

**Dr. L. Rathaiah,** who has given us a lot of support and freedom during our project work.

We thank to the entire **faculty** who have been a constant source of support during our study tenure.

**ii**

## ABSTRACT

Our project aims to develop an integrated machine learning system that assists visually impaired individuals in navigating and identifying common objects in real-time. The system uses deep learning algorithms to detect and recognize objects without the need for assistance from another person. By using a camera to capture multiple objects, the system can identify the position and names of the objects, producing voice output using Google's Text-To-Speech (GTTS) API module. The system is capable of recognizing 80 categories of outdoor objects and generates speech output, even with reduced spectral information. Our solution seeks to address the daily challenges faced by visually impaired individuals and enhance their independence. The system is expected to have a significant impact on the quality of life of visually impaired individuals, allowing them to access basic day-to-day needs with ease.

**iii**

## CONTENTS

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| Declaration | i |
| Acknowledgement | ii |
| Abstract | iii |
| List of Figures | iv |

|  |  |
| --- | --- |
| **TITLES** | **PAGE NO** |
| **1. INTRODUCTION** | **1** |
| **1.1 Machine Learning** | **2** |
| 1.1.1 What is Machine Learning | 2 |
| 1.1.2 How Machine Learning works | 3 |
| 1.1.3 Machine Learning methods | 5 |
| 1.1.4 History of Machine Learning | 6 |
| **1.2 Anaconda** | **7** |
| 1.2.1 Introduction to Anaconda | 7 |
| 1.2.2 Installation and Setup | 8 |
| 1.2.3 Anaconda Navigator | 10 |
| **1.3 Jupyter Notebook** | **10** |
| 1.3.1 Introduction to Jupyter Notebook | 11 |
| 1.3.2 Working with Jupyter online | 11 |
| 1.3.3 Installation and getting started | 12 |

|  |  |
| --- | --- |
| **1.4 Tools** | **13** |
| 1.4.1Numpy | 13 |
| 1.4.2 Pandas | 14 |
| 1.4.3 Matplotlib | 16 |
| 1.4.4 Seaborn | 16 |
| 1.4.5 Scikit Learn | 18 |
| **2. LITERATURE SURVEY** | **20** |
| 2.1 Literature Survey | 21 |
| **3. FLOW DIAGRAMS** | **23** |
| 3.1 Data Flow Diagram | 24 |
| 3.2 UML Diagrams | 25 |
| **4. SYSTEM ANALYSIS** | **28** |
| 4.1 Problem Statement | 29 |
| 4.2 Existing System | 29 |
| 4.3 Proposed System | 29 |
| 4.3.1 Architecture of proposed system | 30 |
| 4.3.2 Algorithm | 32 |
| 4.5 Conclusion | 36 |
| 4.6 Source Code | 36 |

|  |  |
| --- | --- |
| **5. REQUIREMENT AND SPECIFICATION** | 37 |
| 5.1 System Specifications | 38 |
| 5.1.1 Hardware Requirements | 38 |
| 5.1.2 Software Requirements | 38 |
| **6. WEB SCREENS** | **39** |
| **6.1 Screen shots** | 40 |
| 6.1.1 Identifying the person | 40 |
| 6.1.2 Identifying the phone  6.1.3 Identifying the bottel | 40  41 |
| **7. REFERENCES** | **42** |
| 7.1 References | 43 |

**LIST OF FIGURES**

### FIGURES PAGE NO

1. Anaconda 8-10
2. Jupyter 11-12
3. Data Flow Diagram 24
4. UML Diagrams 25-28
5. Architecture of proposed system 30
6. Accuracy 34
7. Precision 35
8. Screenshot of outputs 40
9. Identifying objects 40
10. Output 41

**iv**

**Chapter 1**

# INTRODUCTION

Page 1

## INTRODUCTION

### Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decision with minimal human intervention**.**

### 1.1.1 What is machine learning?

Machine learning is a branch of artificial intelligence (AI) focused on building applications that learn from data and improve their accuracy over time without being programmed to do so.

In data science, an algorithm is a sequence of statistical processing steps. In machine learning, algorithms are 'trained' to find patterns and features in massive amounts of data in order to make decisions and predictions based on new data. The better the algorithm, the more accurate the decisions and predictions will become as it processes more data.

Today, examples of machine learning are all around us. Digital assistants search the web and play music in response to our voice commands. Websites recommend products and movies and songs based on what we bought, watched, or listened to before. Robots vacuum our floors while we do . . . something better with our time. Spam detectors stop unwanted emails from reaching our inboxes. Medical image analysis systems help doctors’ spot tumours they might have missed. And the first self-driving cars are hitting the road.

We can expect more. As big data keeps getting bigger, as computing becomes more powerful and affordable, and as data scientists keep developing more capable algorithms, machine learning will drive greater and greater efficiency in our personal and work lives.

Page 2

#### 1.1.2 How Machine Learning Works

There are four basic steps for building a machine learning application (or model). These are typically performed by data scientists working closely with the business professionals for whom the model is being developed.

#### Step 1: Select and prepare a training data set

Training data is a data set representative of the data the machine learning model labelled to solve the problem it’s designed to solve. In some cases, the training data is labelled data— ‘tagged’ to call out features and classifications the model will need to identify. Other data is unlabelled, and the model will need to extract those features and assign classifications on its own.

In either case, the training data needs to be properly prepared randomized, de-duped and checked for imbalances or biases that could impact the training It should also be divided into two subsets: the training subset , which will be used to train the application and the evaluation subset, used to test and refine it.

**Step 2: Choose an algorithm to run on the training data set** Again, an algorithm is a set of statistical processing steps. The type of algorithm depends on the type (labelled or unlabelled) and amount of data in the training data set and on the type of problem to be solved.

Common types of machine learning algorithms for use with labelled data include the following:

* **Regression algorithms:** Linear and logistic regressions are examples of regression algorithms used to understand relationships in data. Linear regression is used to predict the value of a dependent variable based on the value of an independent variable. Logistic regression can be used when the dependent variable is binary in nature: A or B. For example, a linear regression algorithm could be trained to predict a salesperson’s annual sales (the dependent variable) based on its relationship to the salesperson’s education or years of experience (the independent variables.) Another type of regression algorithm called a support vector machine is useful when dependent variables are more difficult to classify.

Page 3

* **Decision trees:** Decision trees use classified data to make recommendations based on a set of decision rules. For example a decision tree that recommends betting on a particular horse to win, place, or show could use data about the horse(e.g., age, winning percentage, pedigree) and apply rules to those factors to recommended an action or decision.
* **Instance-based algorithms**: A good example of an instance-based algorithm is K- Nearest Neighbour or k-nn. It uses classification to estimate how likely a data point is to be a member of one group or another based on its proximity to other data points.

Algorithms for use with unlabelled data include the following:

* **Clustering algorithms:** Think of clusters as groups. Clustering focuses on identifying groups of similar records and labelling the records according to the group to which they belong. This is done without prior knowledge about the groups and their characteristics. Types of clustering algorithms include the K-means, Two Step, and Kohonen clustering.
* **Association algorithms:** Association algorithms find patterns and relationships in data and identify frequent ‘if-then’ relationships called association rules. These are similar to the rules used in data mining.
* **Neural networks:** A neural network is algorithm that defines a layered network of calculations featuring an input layer, where data is ingested; at least one hidden layer, where calculations are performed make different conclusions about input; and an output layer. Where each conclusion is assigned a probability. A deep neural network defines a network with multiple hidden layers, each of which successively refines the results of the previous layer. (For more, see the “Deep learning” section below.)

### Step 3: Training the algorithm to create the model

Training the algorithm is an iterative process–it involves running variables through the algorithm, comparing the output with the results it should have produced, adjusting weights and biases within the algorithm that might yield a more accurate result, and running the variables again until the algorithm returns the correct result most of the time. The resulting trained, accurate algorithm is the machine learning model—an important distinction to note, because 'algorithm' and 'model' are incorrectly used interchangeably, even by machine learning mavens.

### Step 4: Using and improving the model

The final step is to use the model with new data and, in the best case, for it to improve in accuracy and effectiveness over time. Where the new data comes from will depend on the problem being solved. For example, a machine learning model designed to identify spam will ingest email messages, whereas a machine learning model that drives a robot vacuum cleaner will ingest data resulting from real-world interaction with moved furniture or new objects in the room.

Page 4

#### 1.1.3 Machine Learning methods

Machine learning methods (also called machine learning styles) fall into three primary categories.

#### Supervised machine learning

Supervised machine learning trains itself on a labelled data set. That is, the data is labelled with information that the machine learning model is being built to determine and that may even be classified in ways the model is supposed to classify data. For example, a computer vision model designed to identify purebred German shepherd dogs might be trained on a data set of various labelled dog images.

Supervised machine learning requires less training data than other machine learning methods and makes training easier because the results of the model can be compared to actual labelled results. But, properly labelled data is expensive to prepare, and there's the danger of over fitting, or creating a model so closely tied and biased to the training data that it doesn't handle variations in new data accurately.

#### Unsupervised machine learning

Unsupervised machine learning ingests unlabelled data—lots and lots of it—and uses algorithms to extract meaningful features needed to label, sort, and classify the data in real- time, without human intervention. Unsupervised learning is less about automating decisions and predictions, and more about identifying patterns and relationships in data that humans would miss. Take spam detection, for example—people generate more email than a team of data scientists could ever hope to label or classify in their lifetimes. An unsupervised learning algorithm can analyse huge volumes of emails and uncover the features and patterns that indicate spam (and keep getting better at flagging spam over time).

#### Reinforcement Learning

Reinforcement learning is the third and most advanced algorithm category in Machine Learning. Unlike supervised and unsupervised learning, reinforcement learning continuously improves its model by leveraging feedback from previous iterations. This is different to supervised and unsupervised learning, which both reach an indefinite endpoint after a model is formulated from the training and test data segments. Reinforcement learning can be complicated and is probably best explained through an analogy to a video game. As a player progresses through the virtual space of a game, they learn the value of various actions under different conditions and become more familiar with the field of play. Those learned values then inform and influence a player’s subsequent behaviour and their performance immediately improves based on their learning and past experience.

#### 1.1.4 History of Machine Learning

The first case of neural networks was in 1943, when neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper about neurons, and how they work. They decided to create a model of this using an electrical circuit, and therefore the neural network was born.

In 1950, Alan Turing created the world-famous Turing Test. This test is fairly simple - for a computer to pass, it has to be able to convince a human that it is a human and not a computer.

1952 saw the first computer program which could learn as it ran. It was a game which played checkers, created by Arthur Samuel.

Frank Rosenblatt designed the first artificial neural network in 1958, called Perception. The main goal of this was pattern and shape recognition.

Another extremely early instance of a neural network came in 1959, when Bernard Widrow and Marcian Hoff created two models of them at Stanford University. The first was called ADELINE, and it could detect binary patterns. For example, in a stream of bits, it could predict what the next one would be. The next generation was called MADELINE, and it could eliminate echo on phone lines, so had a useful real world application. It is still in use today.

Despite the success of MADELINE, there was not much progress until the late 1970s for many reasons, mainly the popularity of the Von Neumann architecture. This is an architecture where instructions and data are stored in the same memory, which is arguably simpler to understand than a neural network, and so many people built programs based on this.

Neural networks use back propagation (explained in detail in the Introduction to Neural Networks), and this important step came in 1986, when three researchers from the Stanford psychology department decided to extend an algorithm created by Widrow and Hoff in 1962. This therefore allowed multiple layers to be used in a neural network, creating what are known as ‘slow learners’, which will learn over a long period of time.

The late 1980s and 1990s did not bring much to the field. However in 1997, the IBM computer Deep Blue, which was a chess-playing computer, beat the world chess champion. Since then, there have been many more advances in the field, such as in 1998, when research at AT&T Bell Laboratories on digit recognition resulted in good accuracy in detecting handwritten postcodes from the US Postal Service. This used back-propagation, which, as stated above, is explained in detail on the Introduction to Neural Networks.

#### 21st Century

Since the start of the 21st century, many businesses have realised that machine learning will increase calculation potential. This is why they are researching more heavily in it, in order to stay ahead of the competition.

#### Some large projects include:

* **Google Brain (2012)** - This was a deep neural network created by Jeff Dean of Google, which focused on pattern detection in images and videos. It was able to use Google’s resources, which made it incomparable to much smaller neural networks. It was later used to detect objects in YouTube videos.
* **Alex Net (2012)** – Alex Net won the Image Net competition by a large margin in 2012, which led to the use of GPUs and Convolution Neural Networks in machine learning. They also created ReLU, which is an activation function that greatly improves efficiency of CNNs.
* **Deep Face (2014)** - This is a Deep Neural Network created by Facebook, which they claimed can recognise people with the same precision as a human can.
* **Deep Mind (2014)** - This company was bought by Google, and can play basic video games to the same levels as humans. In 2016, it managed to beat a professional at the game Go, which is considered to be one the world’s most difficult board games.
* **Open AI (2015)** - This is a non-profit organisation created by Elon Musk and others, to create safe artificial intelligence that can benefit humanity.
* **Amazon Machine Learning Platform (2015)** - This is part of Amazon Web Services, and shows how most big companies want to get involved in machine learning. They say it drives many of their internal systems, from regularly used services such as search recommendations and Alexa, to more experimental ones like Prime Air and Amazon Go.
* **ResNet (2015)** - This was a major advancement in CNNs, and more information can be found on the [Introduction to CNNs](https://www.doc.ic.ac.uk/~jce317/introduction-cnns.html) page.
* U-net (2015) - This is an CNN architecture specialised in biomedical image segmentation. It introduced an equal amount of up sampling and down sampling layers, and also skip connections. More information on what this means can be found on the [Semantic](https://www.doc.ic.ac.uk/~jce317/semantic-segmentation.html) [Segmentation](https://www.doc.ic.ac.uk/~jce317/semantic-segmentation.html) page.

### Anaconda

Anaconda is the data science platform for data scientists, IT professionals and business leaders of tomorrow. It is a distribution of Python, R, etc. With more than 300 packages

for data science, it becomes one of the best platforms for any project. In this python anaconda tutorial

### Introduction to Anaconda

Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science. Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup which can deploy any project with the click of a single button.

Now that we know what anaconda is, let’s try to understand how we can install anaconda and set up an environment to work on our systems.

Page 7

### Installation and Setup

To install anaconda go to [**https://www.anaconda.com/distribution/**](https://www.anaconda.com/distribution/)

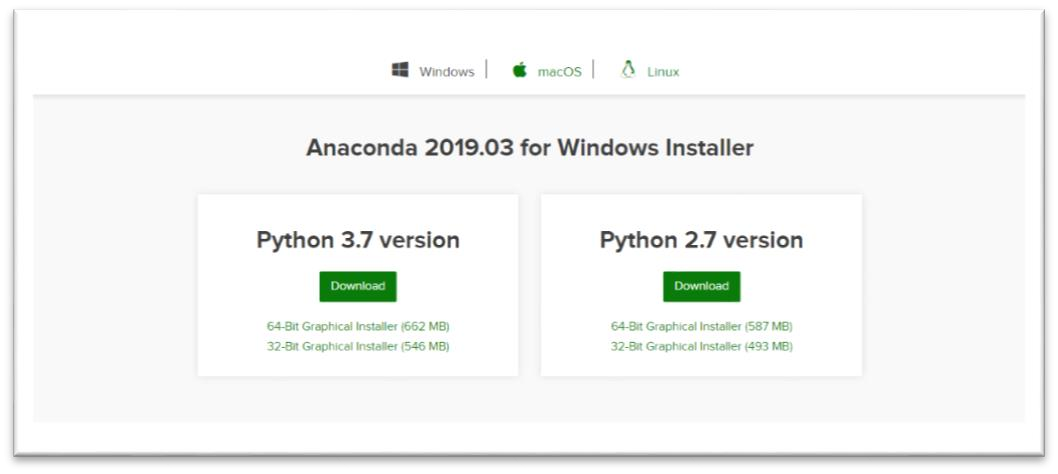


Fig.1

Choose a version suitable for you and click on download. Once you complete the download, open the setup.

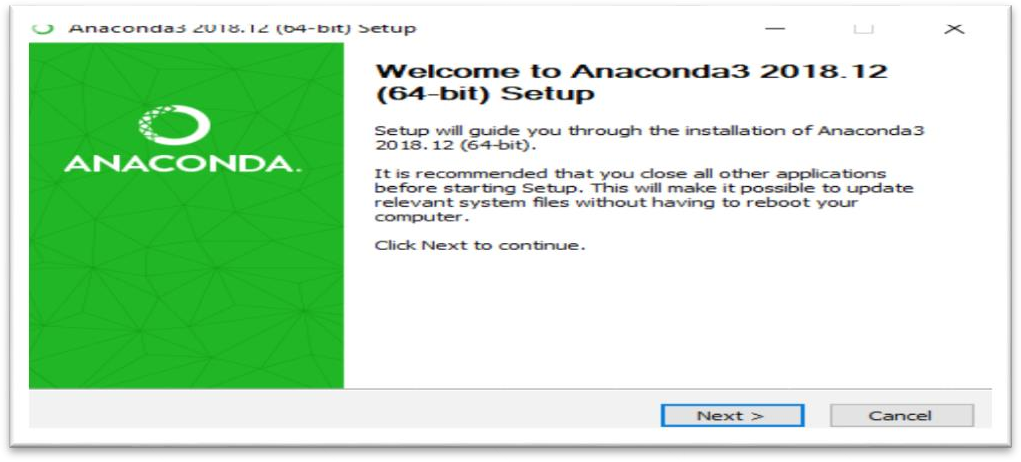


Fig.2

Follow the instructions in the setup. Don’t forget to click on add anaconda to my path environment variable. After the installation is complete, you will get a window like shown in the image below.

Page 8

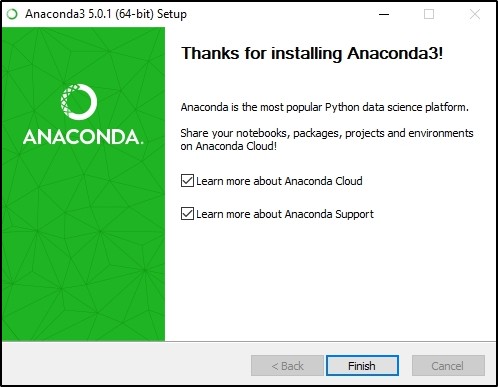
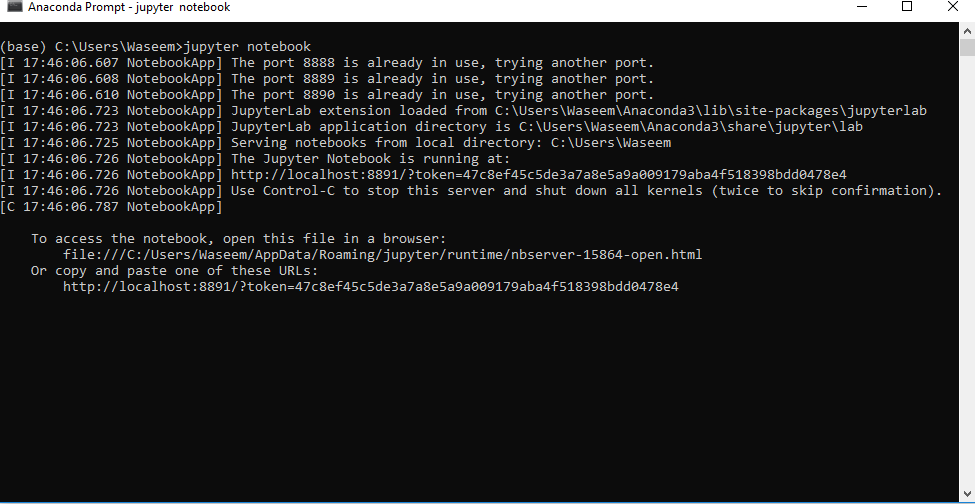


Fig.3

After finishing the installation, open anaconda prompt and type Jupyter notebook.



You will see a window like shown in the image below.

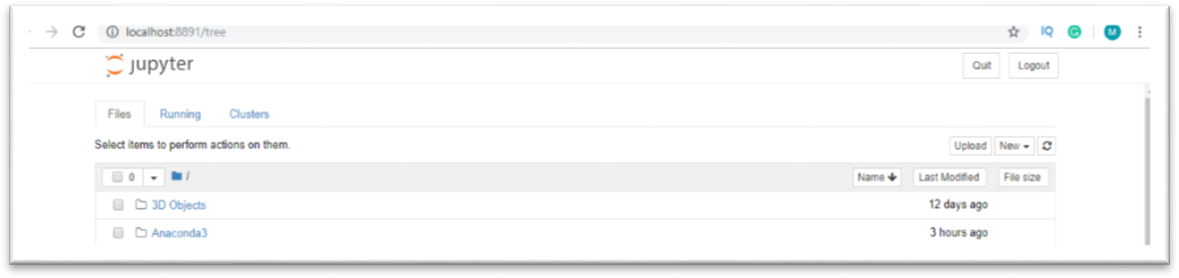
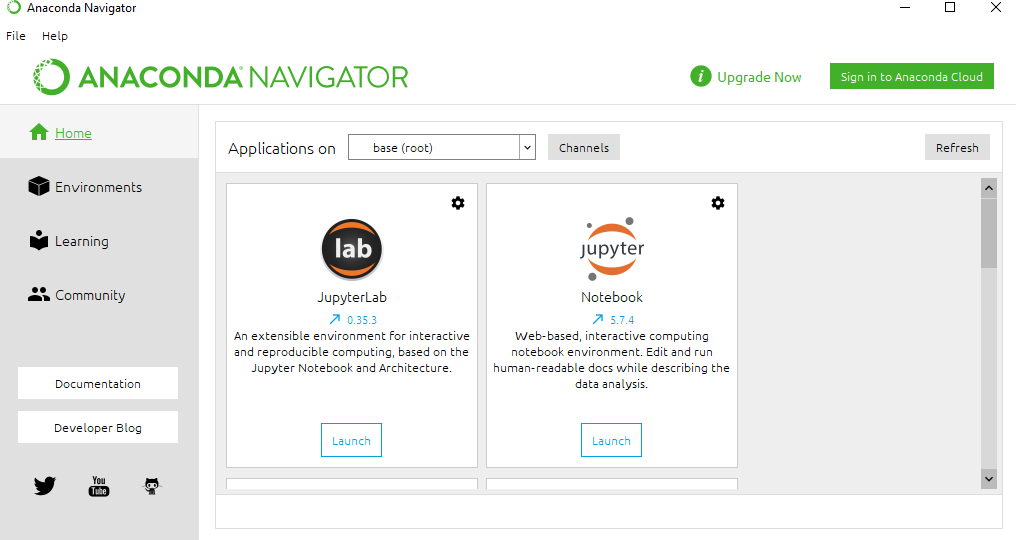


Fig.4

Page 9

### 1.2.3 Anaconda Navigator

Anaconda Navigator is a desktop GUI that comes with the anaconda distribution. It allows us to launch applications and manage conda packages, environment and without using command-line commands.



### Jupyter Notebook

Project Jupyter started as a spin-off from IPython project in 2014. IPython’s

Language-agnostic features were moved under the name – Jupyter. The name is a reference to core programming languages supported by Jupyter which are Julia, Python and R products under Jupyter project are intended to support interactive data science and scientific computing.

The project Jupyter consists of various products described as under

* **IPykernel** − This is a package that provides IPython kernel to Jupyter.
* **Jupyter client** − This package contains the reference implementation of the Jupyter protocol. It is also a client library for starting, managing and communicating with Jupyter kernels.
* **Jupyter notebook** − This was earlier known as IPython notebook. This is a web based interface to IPython kernel and kernels of many other programming languages.
* **Jupyter kernels** − Kernel is the execution environment of a programming language

for Jupyter products.

* **Qtconsole** − A rich Qt-based console for working with Jupyter kernels
* **nbconvert** − Converts Jupyter notebook files in other formats
* **JupyterLab** − Web based integrated interface for notebooks, editors, consoles etc.
* **nbviewer** − HTML viewer for notebook files

Page 10

### 1.3.1 Introduction to Jupyter Notebook

IPython notebook was developed by Fernando Perez as a web based front end to IPython kernel. As an effort to make an integrated interactive computing environment for multiple languages, Notebook project was shifted under Project Jupyter providing front end for programming environments Juila and R in addition to Python.

A notebook document consists of rich text elements with HTML formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in Python or other supporting languages.

Jupyter notebook is a client-server application. The application starts the server on local machine and opens the notebook interface in web browser where it can be edited and run from. The notebook is saved as ipynb file and can be exported as html, pdf and LaTex files.

### 1.3.2 Working with Jupyter Online

If you are new to Jupyter, you can try features of Jupyter notebook before installing on your local machine. For this purpose, visit [**https://jupyter.org**](https://jupyter.org/try) in your browser and choose ‘Try Jupyter with Python’ option.



Fig.6

This will open home page of [**https://mybinder.org**](https://mybinder.org/) From the File menu, choose new notebook option to open a blank Jupyter in your browser. The input cell, as similar to that in IPython terminal, will be displayed. You can execute any Python expression in it.

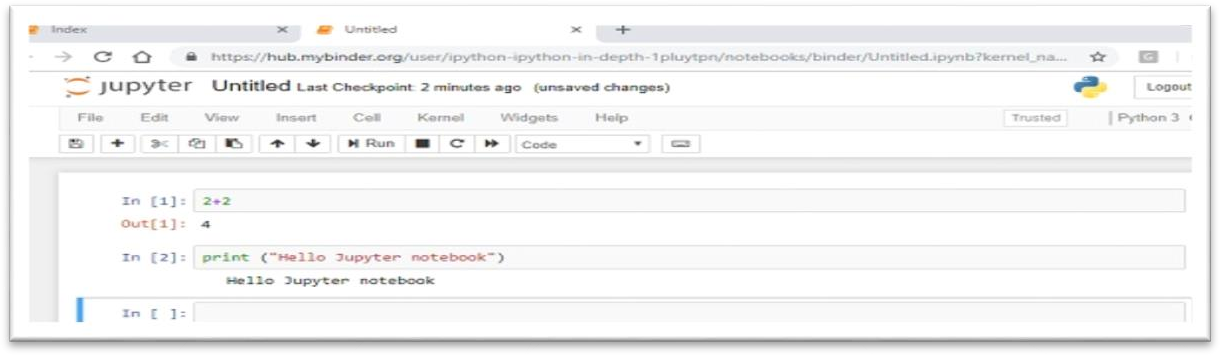


Fig.7

Page 11

### 1.3.3 Installation and getting started

You can easily install Jupyter notebook application using pip package manager.

#### Pip3 install jupyter

. To start the application, use the following command in the command prompt window. c:\python36>jupyter notebook

The server application starts running at default port number 8888 and browser window opens

to show notebook dashboard.

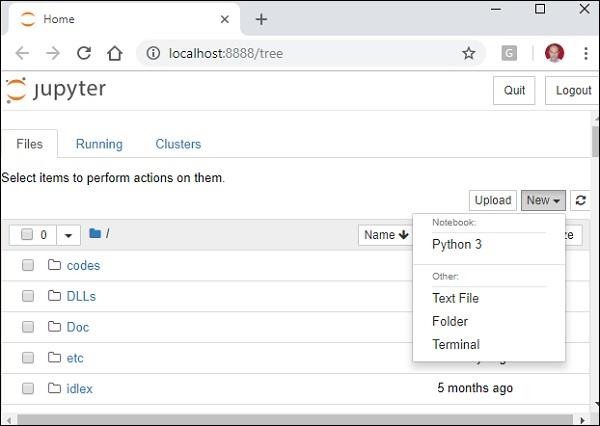


Fig.8

Observe that the dashboard shows a dropdown near the right border of browser with an arrow beside the New button. It contains the currently available notebook kernels. Now, choose Python 3, then a new notebook opens in a new tab. An input cell as similar to that of in IPython console is displayed.

You can execute any Python expression in it. The result will be displayed in the Out cell.

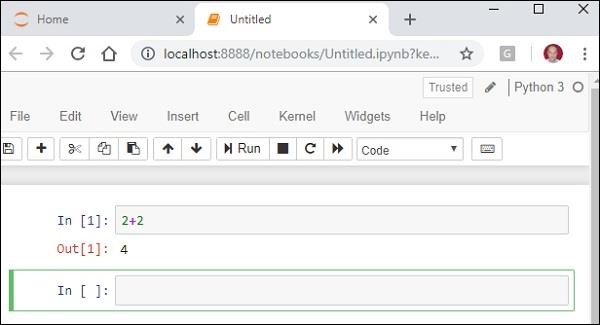


Fig.9

Page 12

### Tools

* + 1. **Numpy**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

### Operations using NumPy

Using NumPy, a developer can perform the following operations −

* + - 1. Mathematical and logical operations on arrays.
      2. Fourier transforms and routines for shape manipulation.
      3. Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

#### NumPy – A Replacement for MatLab

NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotlib (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.

### Numpy Environment Setup

Standard Python distribution doesn't come bundled with NumPy module. A lightweight alternative is to install NumPy using popular Python package installer, pip.

#### pip install numpy

The best way to enable NumPy is to use an installable binary package specific to your operating system. These binaries contain full SciPy stack (inclusive of NumPy, SciPy, matplotlib, IPython, SymPy and nose packages along with core Python).

### Windows

Anaconda ***(from*** [***https://www.continuum.io***](https://www.continuum.io/)***)*** is a free Python distribution for SciPy stack. It is also available for Linux and Mac.

Canopy ***(***[***https://www.enthought.com/products/canopy/***](https://www.enthought.com/products/canopy/)***)*** is available as free as well as commercial distribution with full SciPy stack for Windows, Linux and Mac.

Page 13

Python (x,y): It is a free Python distribution with SciPy stack and Spyder IDE for Windows OS. (Downloadable from [***https://www.python-xy.github.io/***](https://python-xy.github.io/)***)***

#### Linux

Package managers of respective Linux distributions are used to install one or more packages in SciPy stack.

#### For Ubuntu

**Sudo apt-get install python-numpy**

Python-scipy python-matplotlibipythonipythonnotebook python-pandas python-sympy python-nose

#### For Fedora

**Sudo yum install numpy scipy python-matplotlibipython**

Python-pandas sympy python-nose atlas-devel

### Building from Source

Core Python (2.6.x, 2.7.x and 3.2.x onwards) must be installed with distutils and zlib module should be enabled.

GNU gcc (4.2 and above) C compiler must be available. To install NumPy, run the following command.

##### Python setup.py install

To test whether NumPy module is properly installed, try to import it from Python prompt.

##### import numpy

If it is not installed, the following error message will be displayed.

##### Traceback (most recent call last):

***File "<pyshell#0>", line 1, in <module> import numpy***

##### ImportError: No module named 'numpy'

Alternatively, NumPy package is imported using the following syntax –

***import numpy as np***

### Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using

Page 14

Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

#### Key Features of Pandas

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

#### Pandas Environment Setup

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip.

#### pip install pandas

If you install Anaconda Python package, Pandas will be installed by default with the following –

#### Windows

* **Anaconda** *(****from*** [***https://www.continuum.io***](https://www.continuum.io/)*)* is a free Python distribution for SciPy stack. It is also available for Linux and Mac.
* **Canopy *(***[***https://www.enthought.com/products/canopy/***](https://www.enthought.com/products/canopy)***)*** is available as free as well as commercial distribution with full SciPy stack for Windows, Linux and Mac.
* **Python** (x,y) is a free Python distribution with SciPy stack and Spyder IDE for Windows OS. ***(Downloadable from*** [***http://python-xy.github.io/***](http://python-xy.github.io/)***)***

#### Linux

Package managers of respective Linux distributions are used to install one or more packages in SciPy stack.

#### For Ubuntu Users

##### sudo apt-get install python-numpy python-scipy python-matplotlibipythonipythonnotebook Python-pandas python-sympy python-nose

Page 15

#### For Fedora Users

***Sudo yum install numpy scipy python-matplotlibipython python-pandas sympy Python-nose atlas-devel***

#### Matplotlib

* + - 1. Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.
      2. Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open source equivalent of MATLAB.
      3. Matplotlib was originally written by John D. Hunter in 2003. The current stable version is 2.2.0 released in January 2018.

#### Matplotlib Environment Setup

Matplotlib and its dependency packages are available in the form of wheel packages on the standard Python package repositories and can be installed on Windows, Linux as well as MacOS systems using the pip package manager.

#### pip3 install matplotlib

In case Python 2.7 or 3.4 versions are not installed for all users, the Microsoft Visual C++ 2008 (64 bit or 32 bit for Python 2.7) or Microsoft Visual C++ 2010 (64 bit or 32 bit for Python 3.4) redistributable packages need to be installed.

If you are using Python 2.7 on a Mac, execute the following command −

#### xcode-select –install

* + 1. **Seaborn**

In the world of Analytics, the best way to get insights is by visualizing the data. Data can be visualized by representing it as plots which is easy to understand, explore and grasp. Such data helps in drawing the attention of key elements.

To analyse a set of data using Python, we make use of Matplotlib, a widely implemented 2D plotting library. Likewise, Seaborn is a visualization library in python

Page 16

#### Seaborn Vs Matplotlib

It is summarized that if Matplotlib “tries to make easy things easy and hard things possible”, Seaborn tries to make a well-defined set of hard things easy too.”

Seaborn helps resolve the two major problems faced by Matplotlib; the problems are

* Default Matplotlib parameters
* Working with data frames

As Seaborn compliments and extends Matplotlib, the learning curve is quite gradual. If you know Matplotlib, you are already half way through Seaborn.

#### Important Features of Seaborn

Seaborn is built on top of Python’s core visualization library Matplotlib. It is meant to serve as a complement, and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in −

* Built in themes for styling matplotlib graphics
* Visualizing univariate and bivariate data
* Fitting in and visualizing linear regression models
* Plotting statistical time series data
* Seaborn works well with NumPy and Pandas data structures
* It comes with built in themes for styling Matplotlib graphics

In most cases, you will still use Matplotlib for simple plotting. The knowledge of Matplotlib is recommended to tweak Seaborn’s default plots.

#### Seaborn Environment Setup

In this section, we will understand the steps involved in the installation of Seaborn.

#### Using Pip Installer

To install the latest release of Seaborn, you can use

#### pip install seaborn

**For Windows, Linux & Mac using Anaconda**

Anaconda ***(from*** [***https://www.anaconda.com/***](https://www.anaconda.com/) ***)*** is a free Python distribution for SciPy stack. It is also available for Linux and Mac.

It is also possible to install the released version using conda *−* **conda install seaborn**

To install the development version of Seaborn directly from github

***https://github.com/mwaskom/seaborn***

#### Scikit learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

#### Features

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modelling the data. Some of the most popular groups of models provided by Sklearn are as follows −

**Supervised Learning algorithms** − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

**Unsupervised Learning algorithms** − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

**Clustering** − This model is used for grouping unlabeled data.

**Cross Validation** − It is used to check the accuracy of supervised models on unseen data.

**Dimensionality Reduction −** It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

**Ensemble methods** − As name suggest, it is used for combining the predictions of multiple supervised models.

**Feature extraction** − It is used to extract the features from data to define the attributes in image and text data.

**Feature selection** − It is used to identify useful attributes to create supervised models. Open Source − It is open source library and also commercially usable under BSD license. **Installation**

If you already installed NumPy and Scipy, following are the two easiest ways to install scikit-learn –

#### Using pip

Following command can be used to install scikit-learn via pip −

#### pip install -U scikit-learn

**Using Conda**

Following command can be used to install scikit-learn via conda −

#### conda install scikit-learn

Page 18

On the other hand, if NumPy and Scipy is not yet installed on your Python workstation then, you can install them by using either pip or conda.

Another option to use scikit-learn is to use Python distributions like Canopy and Anaconda because they both ship the latest version of scikit-learn.

**Chapter 2**

# LITERATURE SURVEY

Page 20

## Literature Survey

#### 2.1 Literature Survey

As the bikers in our country are increasing ,the road mishapes are also increasing day by day, due to which many deaths occur, most of them are caused due to most common negligence of not wearing helmets, also many deaths occur due to lack prompt medical attention needed by the injured person.

A Hybrid Approach for Helmet Detection for Riders Safety”In this model various previous methods related to automatic helmet detection has been taken into consideration and the new model has been given. This is a technique of automatic helmet detection , where the input is of either the video which has been recorded or it might be a video through a web camera.

Image procurement :This is the very first step of any vision system , where cameras are used to capture images of riders on road.

Preliminary processing technique :This step is mainly focused on elimination of background noise , enhancement of contrast and image binarization.

Vehicle classification - This step is mainly focused on vehicle classification based on two main parameters i.e aspect ratio and size of the particular vehicle and then the vehicle are classified.

Helmet detection - This step includes extraction of head part from the classified image and providing it to ROI where the matching of ROI and trained features happen to determine whether helmet is there or no.

This model gives a idea of the number of people who violate the traffic rules. It is also cost effective as we use open source technology like OpenCV , etc. for development purpose. Further this model can be used to detect people talking on phone while driving and to identify people driving at a high speed.

“Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network”[5] This model tells us that since the motorcycles are affordable, people use it for daily transportation. Due to this increased use the occurance of accidents are high . Major of the accidents include head injury, which is due to helmet violation by the motorcycle users. As many cities have surveillance system for safety purpose , we can use it for detecting non helmet riders which would be a cost effective approach. This approach uses a machine learning technique , CNN(Convolution Neural Network) for getting good images inspite of various problems like illumination, climate changes , etc. There are four different steps included in the process of this model

Background modeling and object detection: This step is basically used for applying adaptive background subtraction to get the images properly and of same quality no matter what ever the conditions might be whether its day time, night or rainy , etc. To separate various factors not needed we use Gaussian mixture model.

Object detection using Convolution neural network: This technique is basically a type of feed forward neural network using back propagation network. The idea of using this technique was due to the ability to extract interdependent data from the images. This technique involves various levels for detecting the object , where in each level we get the data and in final level the entire image is finally formed.

Recognizing motorcycle from moving objects: We use bounding boxes technique for the identification of the motorcycle from other objects. These boxes are evaluated by providing them as an input to the CNN model , which in reference to the various data in test model gets to know motorcycle and other .

Recognizing motorcyclists with helmet: To identify motorcyclists we apply cropping for the top one fourth of the image, cause that’s the position where the head of the motorcyclists would always be. Then we find the doing subtraction of the binary image of the same. Then CNN is applied to get the output.

This model gives a well defined way of dealing with helmet detection and various way of getting rid from the problem. Thus this is a new approach using machine learning apart from the previous approach which used image processing and other old technologies.

Page 21

**CHAPTER 3**

# FLOW DIAGRAMS

* 1. **Data Flow Diagram**
  2. **Flow Diagrams**

#### Data Collection:

Fig.11

Data Collection involves collecting the data as per the requirements of the project. In this paper we have performed two experiments with two different data-sets respectively. The first data-set was obtained from our college placement department. It consists of 1343 records and the other data-set was obtained from kaggle and classified them into two classes manually based on the criteria. This data-set consists of 13818 samples.

#### Data Preparation:

Data preparation involves the cleaning and exploration of data to find relationships among the features of the data. In first experiment the data was inconsistent and had duplicates. These had to be removed and the data-set is cleaned. The second data-set had many features and the data is explored and the right features were selected for the utilization of the model.

#### ML Model Training:

Initializing the ML models and fitting them to train. We have used various classifiers like Logistic Regression, SVM, KNN, Decision Tree, Random Forest and Gradient Boosting Classifier. The training to testing ratio for the first data-set is 80/20 and 70/30 for the second data-set.

#### Model Evaluation:

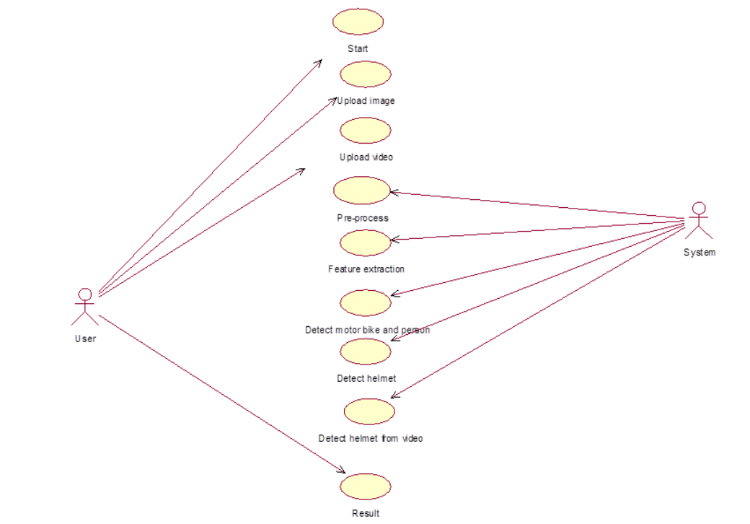
This phase involves evaluating the models to see the performance of each model. Both experiments were evaluated by using confusion matrix and accuracy score. Update the model parameters to improve the performance.

#### Deployment:

Deployment is to host the application in some cloud platform. We have deployed our models in Heruko cloud platform.

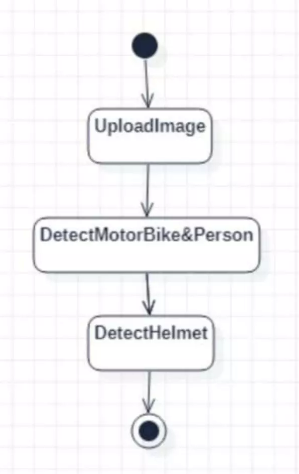
#### 3.2 UML Diagrams

* + 1. **Use Case Diagram**

****

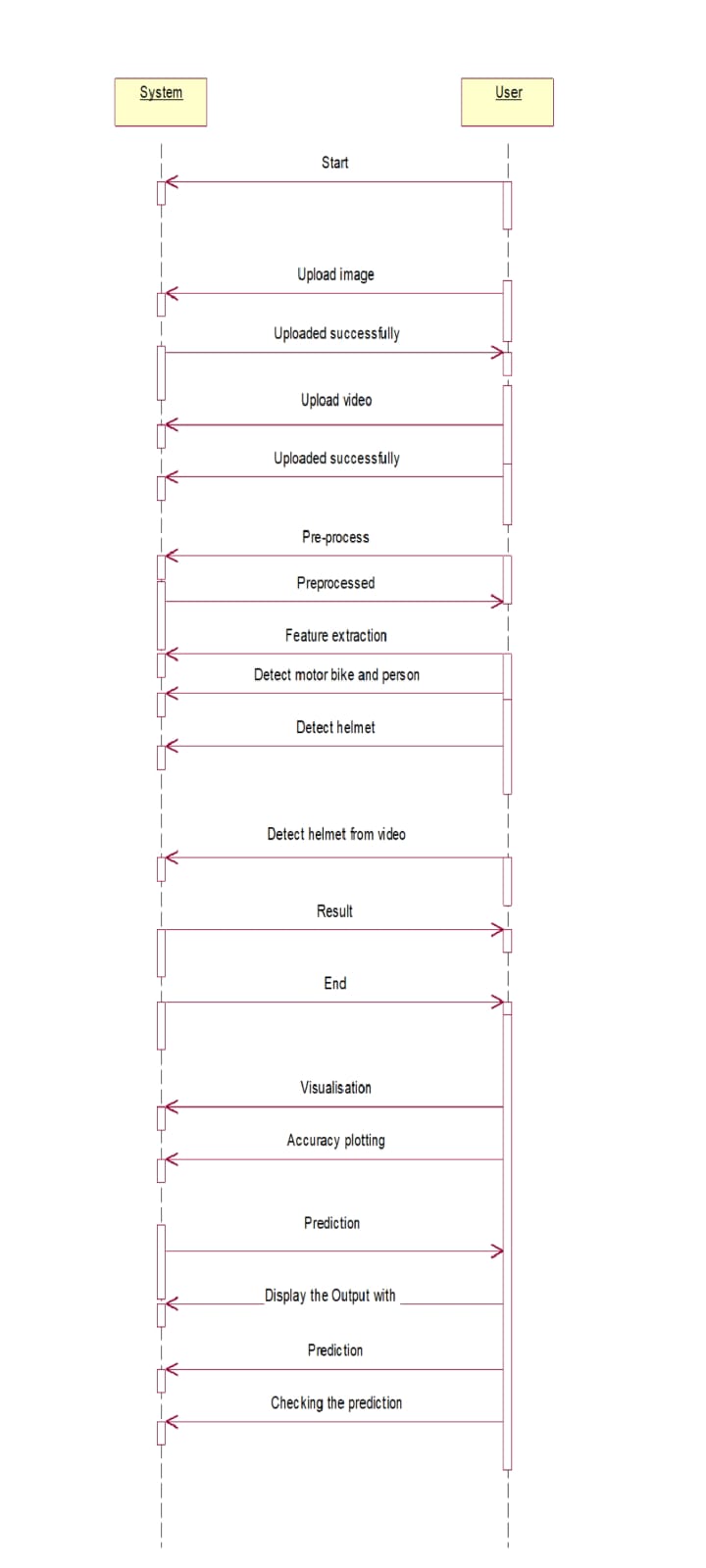
* In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system.
* Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use- case diagrams describe what the system does and how the actors use it, but not how the system operates internally.
* A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is just a special kind of diagram and shares the same common properties as do all other diagrams, i.e. a name and graphical contents that are a projection into a model.

**Activity Diagram**

****

An activity diagram shows the flow from activity to activity. An activity diagram is basically a projection of the elements found in an activity graph, a special case of a state machine in which all or most states are activity states and in which all or most transitions are triggered by completion of activities in the source.

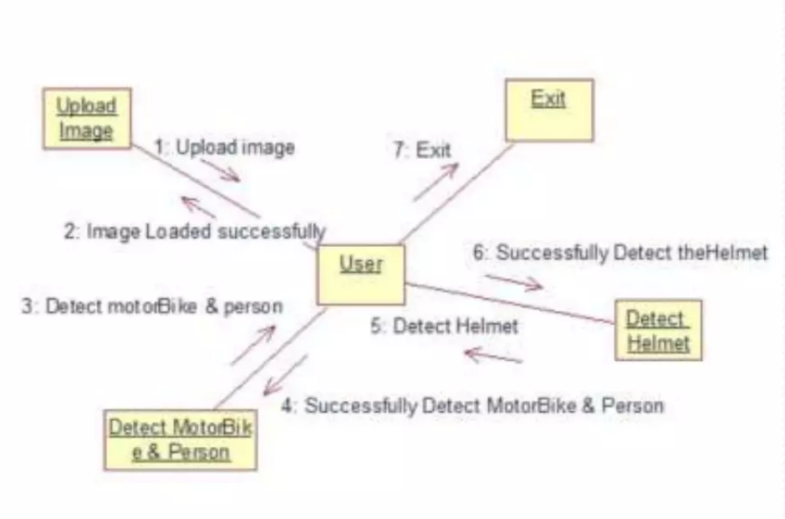
#### Sequence Diagram

****

Page 26

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

**Collaboration Diagram**

****

* The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Page 27

**CHAPTER 4**

# SYSTEM ANALYSIS

Page 28

## SYSTEM ANALYSIS

### 4.1 Problem Statement

As the bikers in our country are increasing ,the road mishapes are also increasing day by day, due to which many deaths occur, most of them are caused due to most common negligence of not wearing helmets, also many deaths occur due to lack prompt medical attention needed by the injured person..

### 4.2 Existing System

### Over the past years, multiple approaches have been proposed to solve the problem of helmet detection. And they used Support Vector Machines(SVM) to classify helmet and human heads without helmets. They used hough transform with SVM to detect the head of the motorcyclist. They applied it to detect helmet on the surveillance system .the drawback of this work is that they only use geometric feature to verify if any safety helmet exists in the set. A helmet detection system is used, and the helmet presence verifies that there is a motorcycle. In order to detect the helmet presence, the edges are computed on the possible helmet region.

### Proposed System.

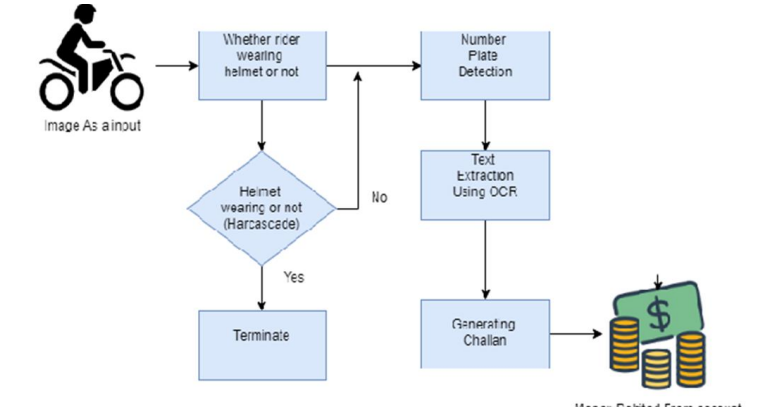
### Therefore, by training with a specific dataset, a Helmet detection model can be implemented. Using this helmet detection model helmet-less riders can be easily detected.

### Based one the detected classes the license plate of the rider is cropped out and saved as an image. This image is given to an Optical Character Recognition (OCR) model which recognizes the text and gives the License Plate number as output in the form of Machine encoded text. And it can also be implemented in real time using a Webcam.

### For real-time helmet detection, there is a need for accuracy and speed. Hence a DNN based model You Only Look Once (YOLO) was chosen. ADVANTAGE OF PROPOSED SYSTEM: 1. Accuracy is more.

#### Page 29

**4.3.1** **Architecture of proposed system**



### Page 30

### Algorithm

### 

### YOLO ALGORITHM

### 

* In 2015, Joseph Redmon and his team proposed the YOLO algorithm, which stands for "You Only Look Once," for real-time object recognition. The YOLO algorithm is a CNN that makes predictions through a single forward pass in the neural network, resulting in faster and more efficient object detection compared to other methods.
* YOLO's speed advantage is due to its ability to familiarize its network with abstract descriptions of objects. Object detection methodologies, which frequently use feature extraction and learning algorithms, are essential for recognizing instances of objects or images belonging to an object class.
* The main objective of object detection is to locate one or more objects in digital images or videos, while object class recognition categorizes objects into specific categories or classes. Each object has unique characteristics that aid in recognizing similar objects in other videos or images and distinguishing them from other classes.
* By combining the YOLO architecture algorithm with the COCO dataset, a fast and efficient deep learning method can be achieved for object recognition.

### Page 35

### 4.4 Conclusion

A Non-Helmet Rider Detection system is developed where a video file is taken as input. If the motorcycle rider in the video footage is not wearing helmet while riding the motorcycle, and then here we are uploading an image to identify the license plate number of that motorcycle is extracted from image and displayed. Object detection principle with YOLO architecture is used for motorcycle, person, helmet and license plate detection. OCR is used for license plate number extraction if the rider is not wearing a helmet. Not only the characters are extracted, but also the frame from which it is also extracted so that it can be used for other purposes. All the objectives of the project are achieved satisfactorily.

### 4.5 Source code

### Object\_detection.py

### import numpy as np

### import time

### import cv2

### INPUT\_FILE='images.jpg'

### OUTPUT\_FILE='predicted.jpg'

### LABELS\_FILE='coco.names'

### CONFIG\_FILE='yolov3.cfg'

### WEIGHTS\_FILE='yolov3.weights'

### CONFIDENCE\_THRESHOLD=0.3

### LABELS = open(LABELS\_FILE).read().strip().split("\n")

### np.random.seed(4)

### COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),

### dtype="uint8")

### net = cv2.dnn.readNetFromDarknet(CONFIG\_FILE, WEIGHTS\_FILE)

### cap = cv2.VideoCapture(0)

### while True:

### ret, frame = cap.read()

### image = frame

### (H, W) = image.shape[:2]

### ln = net.getLayerNames()

### # outputlayers = [layer\_names[i-1] for i in net.getUnconnectedOutLayers()]

### ln = [ln[i- 1] for i in net.getUnconnectedOutLayers()]

### blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),

### swapRB=True, crop=False)

### net.setInput(blob)

### start = time.time()

### layerOutputs = net.forward(ln)

### end = time.time()

### print("[INFO] YOLO took {:.6f} seconds".format(end - start))

### boxes = []

### confidences = []

### classIDs = []

### for output in layerOutputs:

### for detection in output:

### scores = detection[5:]

### classID = np.argmax(scores)

### confidence = scores[classID]

### if confidence > CONFIDENCE\_THRESHOLD:

### box = detection[0:4] \* np.array([W, H, W, H])

### (centerX, centerY, width, height) = box.astype("int")

### x = int(centerX - (width / 2))

### y = int(centerY - (height / 2))

### boxes.append([x, y, int(width), int(height)])

### confidences.append(float(confidence))

### classIDs.append(classID)

### idxs = cv2.dnn.NMSBoxes(boxes, confidences, CONFIDENCE\_THRESHOLD,

### CONFIDENCE\_THRESHOLD)

### if len(idxs) > 0:

### for i in idxs.flatten():

### (x, y) = (boxes[i][0], boxes[i][1])

### (w, h) = (boxes[i][2], boxes[i][3])

### color = [int(c) for c in COLORS[classIDs[i]]]

### cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)

### text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])

### cv2.putText(image, text, (x, y - 5), cv2.FONT\_HERSHEY\_SIMPLEX,

### 0.5, color, 2)

### cv2.imshow('Image',image)

### cv2.waitKey(1)

### cv2.destroyAllWindows(

**Sample.py**

import numpy as np

import time

import cv2

from gtts import gTTS

import os

INPUT\_FILE='images.jpg'

OUTPUT\_FILE='predicted.jpg'

LABELS\_FILE='coco.names'

CONFIG\_FILE='yolov3.cfg'

WEIGHTS\_FILE='yolov3.weights'

CONFIDENCE\_THRESHOLD=0.3

LABELS = open(LABELS\_FILE).read().strip().split("\n")

np.random.seed(4)

COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),

dtype="uint8")

net = cv2.dnn.readNetFromDarknet(CONFIG\_FILE, WEIGHTS\_FILE)

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

image = frame

(H, W) = image.shape[:2]

ln = net.getLayerNames()

# outputlayers = [layer\_names[i-1] for i in net.getUnconnectedOutLayers()]

ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]

blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),

swapRB=True, crop=False)

net.setInput(blob)

start = time.time()

layerOutputs = net.forward(ln)

end = time.time()

print("[INFO] YOLO took {:.6f} seconds".format(end - start))

boxes = []

confidences = []

classIDs = []

for output in layerOutputs:

for detection in output:

scores = detection[5:]

classID = np.argmax(scores)

confidence = scores[classID]

if confidence > CONFIDENCE\_THRESHOLD:

box = detection[0:4] \* np.array([W, H, W, H])

(centerX, centerY, width, height) = box.astype("int")

x = int(centerX - (width / 2))

y = int(centerY - (height / 2))

boxes.append([x, y, int(width), int(height)])

confidences.append(float(confidence))

classIDs.append(classID)

idxs = cv2.dnn.NMSBoxes(boxes, confidences, CONFIDENCE\_THRESHOLD,

CONFIDENCE\_THRESHOLD)

if len(idxs) > 0:

for i in idxs.flatten():

(x, y) = (boxes[i][0], boxes[i][1])

(w, h) = (boxes[i][2], boxes[i][3])

color = [int(c) for c in COLORS[classIDs[i]]]

cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)

text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])

cv2.putText(image, text, (x, y - 5), cv2.FONT\_HERSHEY\_SIMPLEX,

0.5, color, 2)

language = 'en'

myobj = gTTS(text=LABELS[classIDs[i]], lang=language,slow=False)

myobj.save("welcome.mp3")

os.system("start welcome.mp3")

cv2.imshow('Image',image)

cv2.waitKey(1)

cv2.destroyAllWindows()

From this sample.py source code ,the object gets detected , based on the trained dataset the detected object get recognized and gives output by an audio message to the impaired people.

Page36

**CHAPTER 5**

# REQUIREMENT AND SPECIFICATION

Page 37

## REQUIREMENT AND SPECIFICATION

**5.1 System Specifications**

* + 1. **5.1.1 Hardware Requirements**

#### CPU : Pentium IV 2.4 GHz.

#### Storage : 512 MBRAM

#### Hard disk : 40 GB.

#### Input device : Standard Keyboard and Mouse

#### Output device: VGA and High-Resolution Monitor

### 5.1.2 Software requirements

* Operating System : Windows
* Coding Language : Python 3.6.8

Page 38

**Chapter 6**

# Model Screens

Page 39

**6.1 Screenshots**

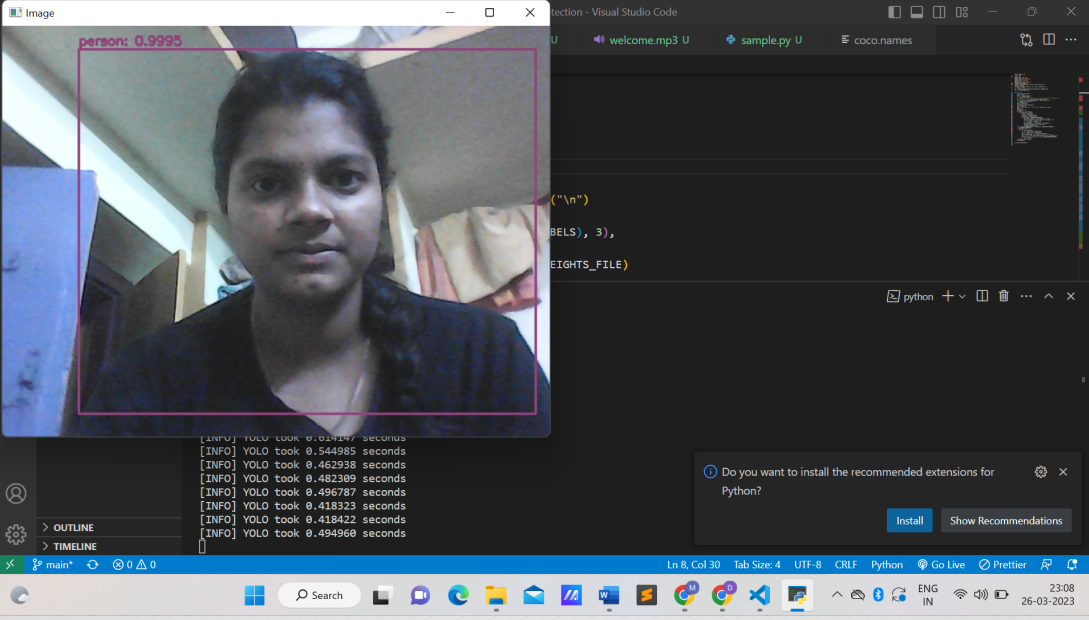


Figure : Identifying Person

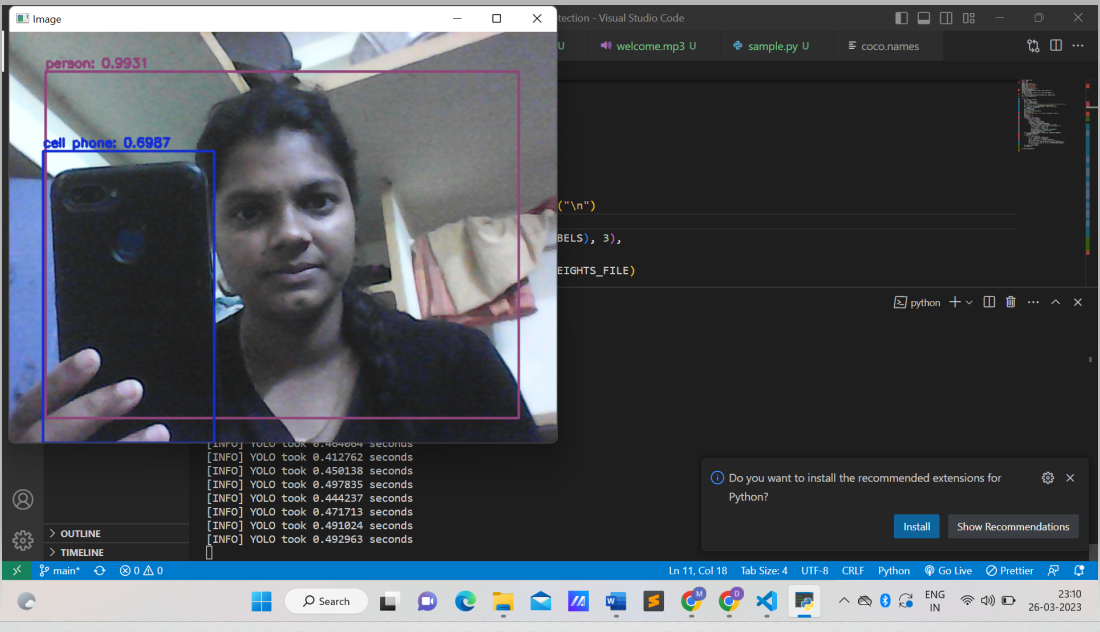


Figure : Identifying Mobile Phone

Page 40

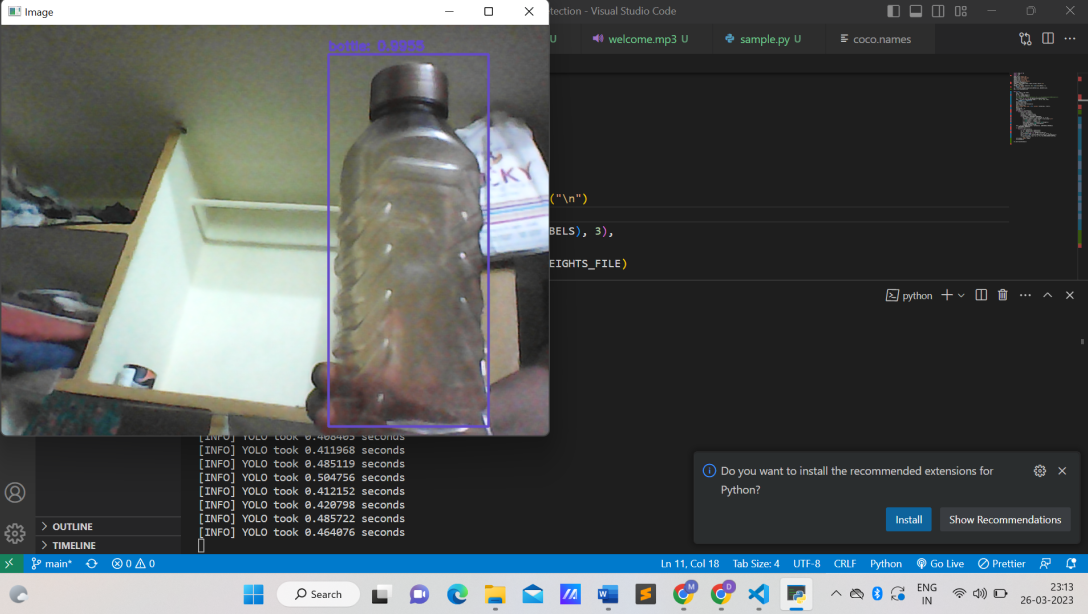


Figure : Identifying Bottel

Page 41

**Chapter 7**

# REFERENCES

Page 42

**7.REFERENCES**

**7.1 References**

**[1].** J. Chiverton, “Helmet Presence Classification with Motorcycle Detection and Tracking”, IET Intelligent Transport Systems ,Vol. 6, Issue 3, pp. 259–269, March 2012.

**[2].**rattapoom Waranusast, Nannaphat Bundon, Vasan Timtong and Chainarong Tangnoi, “Machine Visiontechniques for Motorcycle Safety Helmet Detection”, 28th International Conference on Image and Vision Computing New Zealand, pp 35-40, IVCNZ 2013.

**[3].** Romuere ilva, Kelson Aires, Thiago antos, Kalyf A dala, Rodrigo Veras, Andr e oares, “Automatic Detection Of Motorcyclists without Helmet”, 2013 XXXIX Latin America Computing Conference (CLEI). IEEE, 2013.

**[4].** Romuere ilva, “Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers”, 27th IBGRAPI Conference on Graphics, Patterns and Images.IEEE, 2014.

**[5].** Thepnimit Marayatr, Pinit Kumhom, “Motorcyclist‟s Helmet Wearing Detection Using Image Processing”, Advanced Materials Research Vol 931- 932, pp. 588-592, May-2014.

**[6].** Amir Mukhtar, Tong Boon Tang, “Vision Based Motorcycle Detection using HOG features”, IEEE International Conference on Signal and Image Processing Applications (ICSIPA) IEEE, 2015.

**[7].** Abu H. M. Rubaiyat, Tanjin T. Toma, Masoumeh Kalantari-Khandani, “Automatic Detection of Helmet Uses for Construction afety”, IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW). IEEE, 2016.

**[8].** XINHUA JIANG “A tudy of Low-resolution Safety Helmet Image Recognition Combining Statistical Featureswith

Artificial Neural Network”. IN: 1473-804x

**[9].** Kunal Dahiya, Dinesh ingh, C. Krishna Mohan, “Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time”, International joint conference on neural network (IJCNN). IEEE, 2016.

Page 43