HELMET DETECTION AND LICENSE PLATE RECOGNITION

Major Project Report

Submitted in partial fulfillment for the award of the degree of

BACHELOROFTECHNOLOGY

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

Dr. N. Thirupathi Rao

ASSITANT PROFESSOR

DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING

VIGNAN'SINSTITUTEOFINFORMATIONTECHNOLOGY(Autonomous)

Affiliated to JNTU Kakinada & Approved by AICTE, New Delhi

Re-Accredited by NBA&NAAC(CGPAof3.41/4.00)

ISO 9001:2015, ISO14001:2015, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM –530049

VIGNAN'SINSTITUTEOFINFORMATIONTECHNOLOGY

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 there quirements 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
Mr. Dinesh Reddy
ASSOCIATE PROFESSOR

Signature of Project Guide
Dr. N. Thirupathi Rao
ASSITANT 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
GUMMALA JYOTHI SHANKAR SWARUP	19L31A05M0
PUSAPATI MONITH CHAITANYA VARMA	19L31A05I5
VAJRAPU RAJA VENKATA GUNNESWARA GUPTA	19L31A05J1

Date:

Place:

ACKNOWLEDGEMENT

We consider it as a privilege to thank all those people who helped us a lot for successful completion of the project "**HELMET DETECTION AND LICENSE PLATE RECOGNITION**".

First of all we would like to thank our project guide **Dr. N. Thirupathi Rao, Assitant 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. Wejustwantedtosaythankyouforbeingsuchawonderfuleducatoraswellasaperson.

Wewouldliketothank**Principal**ofVignan'sInstituteofInformationTechnology

Dr. B. Arundhati, for her encourage mentors during the course of this project work.

We also express our sincere thanks tour be loved **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.

ABSTRACT

Deep learning approach Road collisions and deaths have shot up disproportionately as motorization has grown more prevalent This study points to A means of communication to locate helmets and license plates using object detection models based on Riders of motor bikes should put on helmets to stay out of incidents such as these to monitor and swear to the safety of motorcycle passengers by suing Their for not wearing helmets, we need a substantial traffic police force. dentification of motorcyclists without helmets in real-time is a crucial task to prevent the occurrence of accidents. the advanced features in YOLOv3 are a feature extractor network with multi-scale detection and some changes in loss function combining detection and classification in a single architecture. In this project, the main principle involved is object detection using deep learning at three levels. The objects detected are person, motorcycle at the first level, helmet detection at the second level, and license plate detection at a third level all using YOLOv3. The license plate is detected and a cropped image of the license plate is used to extract its digits using OCR. We have used the above-mentioned methods to build integrated systems for helmet detection and license plate number extraction. This project utilizes YOLOv3 to locate things at three levels using deep learning, enabling helmet detection at stage two and license plate detection at level three. Whenever a license plate appears, an OCR technology uses it to extract the digits using a cropped image of the plate. Our equipment for retrieving registration plates and identifying.

TITLE	PAGE NO
Declaration	i
Acknowledgment	ii
Abstract	iii
List of figures	Iv

TITLE	PAGE NO
1.INTRODUCTION	1
1.1 Deep Learning	2
1.1.1 What is Deep Learning	2
1.1.2 How Deep Learning works	3
1.2 Deep Learning Architecture	4
1.2.1 components of Deep Learning	4
1.3 Deep learning methods	5
1.4 History of Deep Learning	6
1.5 Anaconda	8
1.5.1 introduction to Anaconda	8
1.5.2 Installation and setup	9
1.5.3 Anaconda Navigator	11
1.6 Jupyter Notebook	11
1.6.1 Introduction of Jupyter Notebook	12
1.6.2 Working with Jupyter Notebook	12
1.6.3 Installation and Getting Started	13
1.7 Tools	14
1.7.1 Numpy	14
1.7.2 Windows	15
1.7.3 pandas	16
1.7.4 Matplotlib	18

2.SYSTEM IMPLEMENTATION 19

2.1 Frontend

2.1.1 About Html 20

2.1.2 About Tailwind CSS 23

2.1.3 About Javascript 28

2.1.4 Lottie Files for SVG 36

2.2 Backend 42

2.2.1 About Python 42

2.2.2 About Flask 43

3.LITERATURE SURVEY 45

4.FLOW DAIGRAMS 48

4.1 Data Flow Daigram 49

4.2 UML Daigrams 50

4.2.1 Use Case Daigram 50

4.2.2 Activity Daigram 51

4.2.3 Sequence Daigram 51

4.2.4 Collaboration Daigram 52

5.SYSTEM ANALYSIS 54

5.1 Problem Statement 55

5.2 Existing System 55

5.3 proposed System 55

5.4 Architecture of proposed system 55

5.5 Algorithm 56

5.5.1 YOLO Algorithm 56

5.5.2 Convolutional Neural Network 57

5.5.3 OpenCV 58

5.5.4 TensorFlow 59

5.5.5 Optical Character Recognition 59

5.5.6 Object Dectection 60

5.6 CONCLUSION 61

5.7 SOURCE CODE

5.7.1 Object Detection 62

6.System Specifications 75

6.1.1 Hardware Requirements 75

6.1.2 Software Requirements 76

7. MODEL SCREENS

7.1 Screenshots 77

8. REFRENCES 82

FIGURES

1.Anaconda Installation 9

2.Anaconda Setup 9

3.Anaconda Download 10

4.Jupyter Notebook 10

5.Jupyter Window 11

6.Anaconda Navigator 11

7.Jupyter Code 12

8.Jupyter Files 13

9.HTML 13

10.Tailwind CSS 20

11.Javascript 24

12.Lottie Files for SVG 28

13.Python 37

14.Flask Framework 43

15.Advantages of Flask 44

16.Usecase Diagram 44

17.Activity Diagram 50

18.Sequqnce Diagram 51

19.Collaboration Diagram 52

20.Architecture 53

21.YOLO 56

22.OpenCV 56

23.TensorFlow 57

24.Optical Character Recognition 58

25.Object Detection 59

26.Homescreen 60

27.After Uploaded Image 61

28.Output of Image Upload 77

29.After Upload Video 78

78

79

79

30.Output of Video upload 80

31.Live Streaming 81

Chapter1

INTRODUCTION

1. INTRODUCTION

1.1Deep Learning

Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions**.**

1.1.1 What is Deep learning?

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture. A formal definition of deep learning is- neurons

Deep Learning is a subset of Machine Learning that is based on artificial neural networks (ANNs) with multiple layers, also known as deep neural networks (DNNs). These neural networks are inspired by the structure and function of the human brain, and they are designed to learn from large amounts of data in an unsupervised or semi-supervised mannerDeep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture. A formal definition of deep learning is- neurons

Deep Learning models are able to automatically learn features from the data, which makes them well-suited for tasks such as image recognition, speech recognition, and natural language processing. The most widely used architectures in deep learning are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

Feedforward neural networks (FNNs) are the simplest type of ANN, with a linear flow of information through the network. FNNs have been widely used for tasks such as image classification, speech recognition, and natural language processing.

Convolutional Neural Networks (CNNs) are a special type of FNNs designed specifically for

image and video recognition tasks. CNNs are able to automatically learn features from the images, which makes them well-suited for tasks such as image classification, object detection, and image segmentation.

1.1.2 How Deep Learning Works

Computer programs that use deep learning go through much the same process as the toddler learning to identify the dog. Each algorithm in the hierarchy applies a nonlinear transformation to its input and uses what it learns to create a statistical model as output. Iterations continue until the output has reached an acceptable level of accuracy. The number of processing layers through which data must pass is what inspired the label deep**.**

In traditional machine learning, the learning process is supervised, and the programmer has to be extremely specific when telling the computer what types of things it should be looking for to decide if an image contains a dog or does not contain a dog. This is a laborious process called feature extraction, and the computer's success rate depends entirely upon the programmer's ability to accurately define a feature set for dog. The advantage of deep learning is the program builds the feature set by itself without supervision. Unsupervised learning is not only faster, but it is usually more accurate.

Initially, the computer program might be provided with training data -- a set of images for which a human has labeled each image dog or not dog with metatags. The program uses the information it receives from the training data to create a feature set for dog and build a predictive model. In this case, the model the computer first creates might predict that anything in an image that has four legs and a tail should be labeled dog. Of course, the program is not aware of the labels four legs or tail. It will simply look for patterns of pixels in the digital data. With each iteration, the predictive model becomes more complex and more accurate.

Unlike the toddler, who will take weeks or even months to understand the concept of dog, a computer program that uses deep learning algorithms can be shown a training set and sort through millions of images, accurately identifying which images have dogs in them within a few minutes.

To achieve an acceptable level of accuracy, deep learning programs require access to immense amounts of training data and processing power, neither of which were easily available to programmers until the era of big data and cloud computing. Because deep learning programming can create complex statistical models directly from its own iterative output, it is able to create accurate predictive models from large quantities of unlabeled, unstructured data. This is important as the internet of things (IoT) continues to become more pervasive because most of the data humans and machines create is unstructured and in labeled. in human brain approximately 100 billion neurons all together this is a picture of an individual

neuron and each neuron is connected through thousands of their neighbors. The question here is how do we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for-output value and in between, there may be lots of neurons interconnected in the hidden layer.

1.2 Deep Learning Architecture

Deep Neural Network:

It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.

Deep Belief Network (DBN):

It is a class of Deep Neural Network. It is multi-layer belief networks. Steps for performing DBN: a. Learn a layer of features from visible units using Contrastive Divergence algorithm. b. Treat activations of previously trained features as visible units and then learn features of features. c. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved

Recurrent (perform same task for every element of a sequence) **Neural Network:**

Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

1.2.1 What are the components of a deep learning network?

The components of a deep neural network are the following.

Input layer:

An artificial neural network has several nodes that input data into it. These nodes make up the input layer of the system.

Hidden layer:

The input layer processes and passes the data to layers further in the neural network. These hidden layers process information at different levels, adapting their behavior as they receive new information. Deep learning networks have hundreds of hidden layers that they can use to analyze a problem from several different angles.

Output layer:

The output layer consists of the nodes that output the data. Deep learning models that output "yes" or "no" answers have only two nodes in the output layer. On the other hand, those that output a wider range of answers have more nodes.

1.3Deep Learning methods

Various methods can be used to create strong deep learning models. These techniques include learning rate decay, transfer learning, training from scratch and dropout.

Learning rate decay:

The learning rate is a hyperparameter -- a factor that defines the system or set conditions for its operation prior to the learning process – that controls how much change the model experiences in response to the estimated error every time the model weights are altered. Learning rates that are too high may result in unstable training processes or the learning of a suboptimal set of weights. Learning rates that are too small may produce a lengthy training process that has the potential to get stuck.

the learning rate decay method -- also called learning rate annealing or adaptive learning rates -- is the process of adapting the learning rate to increase performance and reduce training time. The easiest and most common adaptations of learning rate during training include techniques to reduce the learning rate over time.

Transfer learning:

This process involves perfecting a previously trained model; it requires an interface to the internals of a preexisting network. First, users feed the existing network new data containing previously unknown classifications. Once adjustments are made to the network, new tasks can be performed with more specific categorizing abilities. This method has the advantage of requiring much less data than others, thus reducing computation time to minutes or hours.

Training from scratch:

This method requires a developer to collect a large labeled data set and configure a network architecture that can learn the features and model. This technique is especially useful for new applications, as well as applications with a large number of output categories. However, overall, it is a less common approach, as it requires inordinate amounts of data, causing training to take days or weeks.

Dropout:

This method attempts to solve the problem of overfitting in networks with large amounts of parameters by randomly dropping units and their connections from the neural network during training. It has been proven that the dropout method can improve the performance of neural networks on supervised learning tasks in areas such as speech recognition, document classification and computational biology.

1.4 History of Deep Learning

Feature extraction is another aspect of deep learning. It is used for pattern recognition and image processing. Feature extraction uses an algorithm to automatically construct meaningful "features" of the data for purposes of training, learning, and understanding. Normally a data scientist, or a programmer, is responsible for feature extraction. The history of deep learning can be traced back to 1943, when Walter Pitts and Warren McCulloch created a computer model based on the neural networks of the human brain. They used a combination of algorithms and mathematics they called "threshold logic" to mimic the thought process. Since that time, Deep Learning has evolved steadily, with only two significant breaks in its development. Both were tied to the infamous Artificial Intelligence winters

The 1960s

Henry J. Kelley is given credit for developing the basics of a continuous Back Propagation Model in 1960. In 1962, a simpler version based only on the chain rule was developed by Stuart Dreyfus. While the concept of back propagation (the backward propagation of errors for purposes of training) did exist in the early 1960s, it was clumsy and inefficient, and would not become useful until 1985.The earliest efforts in developing deep learning algorithms came from Alexey Grigoryevich Ivakhnenko (developed the Group Method of Data Handling) and Valentin Grigor′evich Lapa (author of Cybernetics and Forecasting Techniques) in 1965. They used models with polynomial (complicated equations) activation functions, that were then analyzed statistically. From each layer, the best statistically chosen features were then forwarded on to the next layer (a slow, manual process).

The 1970s

During the 1970's the first AI winter kicked in, the result of promises that couldn't be kept. The impact of this

lack of funding limited both DL and AI research. Fortunately, there were individuals who carried on the research without funding. The first "convolutional neural networks" were used by Kuniko Fukushima. Fukushima designed neural networks with multiple pooling and convolutional layers. In 1979, he developed an artificial neural network, called Neocognitron, which used a hierarchical, multilayered design. This design allowed the computer the "learn" to recognize visual patterns. The networks resembled modern versions, but were trained with a reinforcement strategy of recurring activation in multiple layers, which gained strength over time. Additionally, Fukushima's design allowed important features to be adjusted manually by increasing the "weight" of certain connections

The 1980s and 1990s

1989, Yann LeCun provided the first practical demonstration of backpropagation at Bell Labs. He combined convolutional neural networks with back propagation onto read “handwritten” digits. This system was eventually used to read the numbers of handwritten checks. This time is also when the second AI winter (1985-90s) kicked in, which also effected research for neural networks and deep learning. Various overly-optimistic individuals had exaggerated the “immediate” potential of Artificial Intelligence, breaking expectations and angering investors. The anger was so intense, the phrase Artificial Intelligence reached pseudoscience status. Fortunately, some people continued to work on AI and DL, and some significant advances were made. recurrent neural networks was developed in 1997, by Sepp Hochreiter and Juergen Schmidhuber.

The 2000-2010

Around the year 2000, The Vanishing Gradient Problem appeared. It was discovered “features” (lessons)

formed in lower layers were not being learned by the upper layers, because no learning signal reached these layers. This was not a fundamental problem for all neural networks, just theones with gradient-based learning methods. The source of the problem turned out to be certain activation functions. A number of activation functions condensed their input, in turn reducing the output range in a somewhat chaotic fashion. This produced large areas of input mapped over an extremely small range. In these areas of input, a large change will be reduced to a small change in the output, resulting in a vanishing gradient. Two solutions used to solve this problem were layer-by-layer pre-training and the development of long short-term memory.

In 2001, a research report by META Group (now called Gartner) described he challenges and opportunities of data growth as three-dimensional. The report described the increasing volume of data and the increasing speed of data as increasing the range of data sources and types. This was a call to prepare for the onslaught of Big Data, which was just starting.

In 2009, Fei-Fei Li, an AI professor at Stanford launched ImageNet, assembled a free database of more than 14 million labeled images. The Internet is, and was, full of unlabeled images. Labeled images were needed to “train” neural nets. Professor Li said, “Our vision was that big data would change the way machine learning works. Data drives learning.”

The 2011-2020

By 2011, the speed of GPUs had increased significantly, making it possible to train convolutional

neural networks “without” the layer-by-layer pre-training. With the increased computing speed, it became obvious deep learning had significant advantages in terms of efficiency and speed. One example is Alex Net, a convolutional neural network whose architecture won several international competitions during 2011 and 2012. Rectified linear units were used to enhance the speed and dropout.

Also in 2012, Google Brain released the results of an unusual project known as The Cat Experiment. The free-spirited project explored the difficulties of “unsupervised learning.” Deep learning uses “supervised learning,” meaning the convolutional neural net is trained …

The 2011-2020

By 2011, the speed of GPUs had increased significantly, making it possible to train convolutional neural networks “without” the layer-by-layer pre-training. With the increased computing speed, it became obvious deep learning had significant advantages in terms of efficiency and speed. One example is Alex Net, a convolutional neural network whose architecture won several international competitions during 2011 and 2012. Rectified linear units were used to enhance the speed and dropout. Also in 2012, Google Brain released the results of an unusual project known as The Cat Experiment. The free-spirited project explored the difficulties of “unsupervised learning.” Deep learning uses “supervised learning,” meaning the convolutional neural net is trained …

1.5 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.

1.5.1 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.

1.5.2 Installation and Setup

To install an aconda go tohttps://www.anaconda.com/distribution/

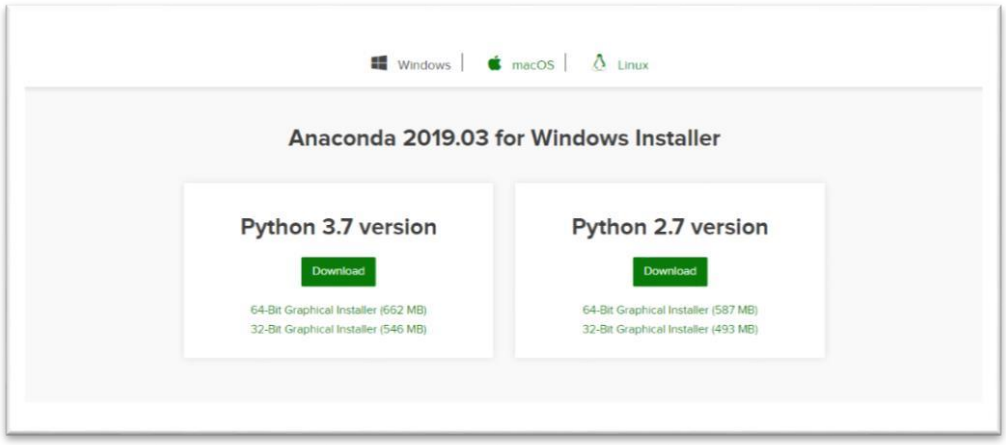

Fig:1 Anaconda Installation

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.

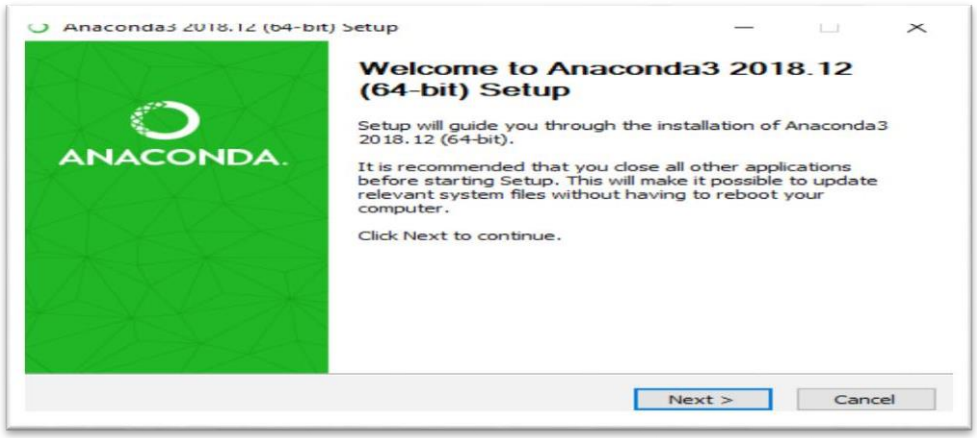

Fig:2 Anaconda setup

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.

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

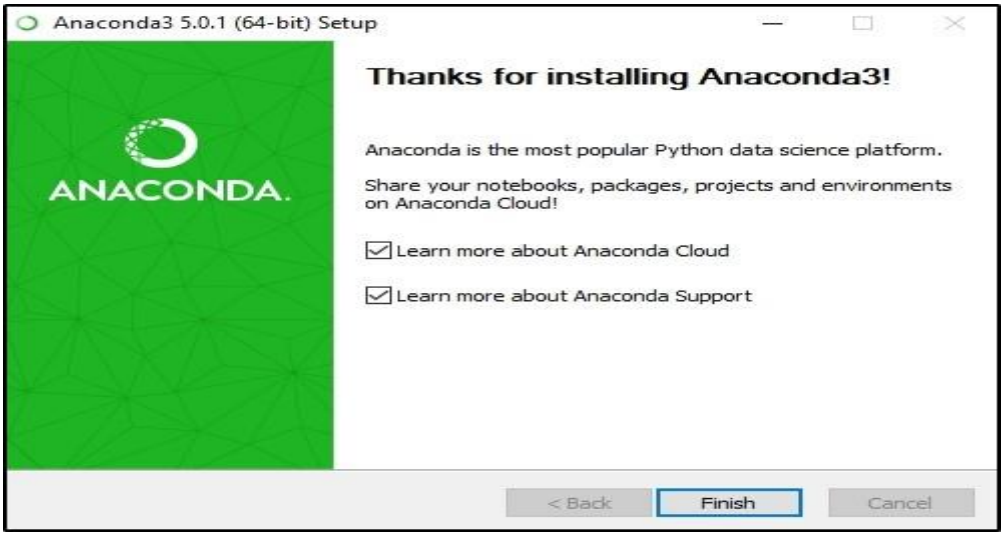

Fig:3 Anaconda download

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

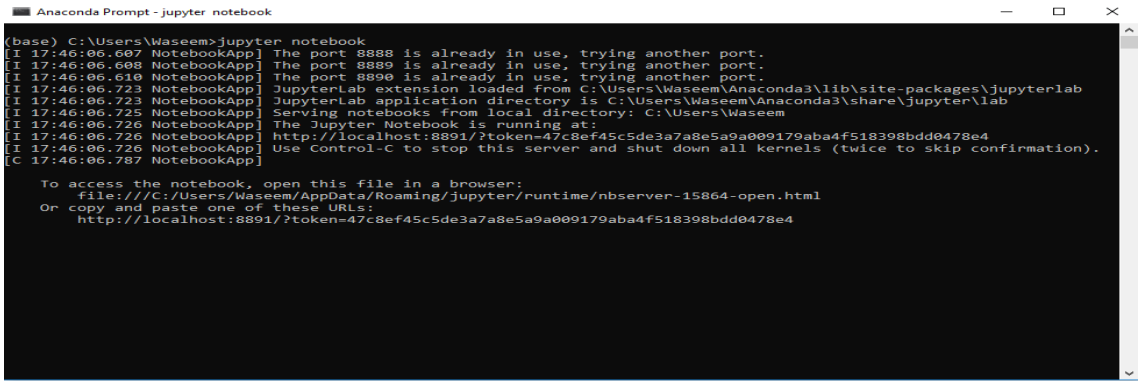

Fig :4 jupyter notebook

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

Fig:5 jupyter window

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

1.5.3 Anaconda Navigator

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

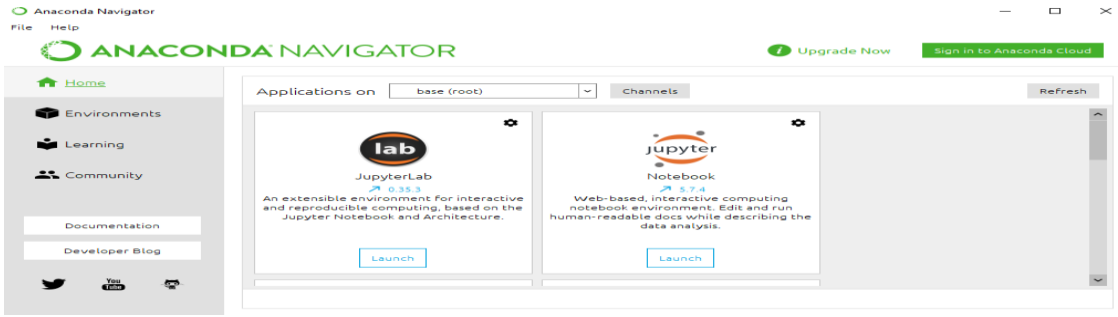

Fig:6 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

1.6 jupyter notebook

Project Jupyter startedasaspin-offfromIPythonprojectin2014.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 productsunderJupyterprojectareintendedtosupportinteractivedatascienceandscientificcomputing
The project Jupyter consists of various products described as under

- **IPy kernel**−This is a package that provides I Python 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 I Python notebook. This is a web based inter face to I Python kernel and kernel so many other programming languages.
- **Jupyter kernels** −Kernel is the execution environment of a programming language

  For Jupyter products.
- **Qtconsole**− Arich Qt-based console for working with Jupyter kernels
- **nbconvert**− Converts Jupyter note book files in other formats
- **Jupyter Lab**−Web based integrated interface for note book, editors, consoles etc.
- **Nb viewer**−HTML viewer for note book fil

1.6.1 Introduction to Jupyter Notebook

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

1.6.2 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** in your browser and choose 'Try Jupyter with Python 'option.

Fig:7 jupyter notebook

This will open home page of **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 I Python terminal, will be displayed. you can execute any Python expression in it

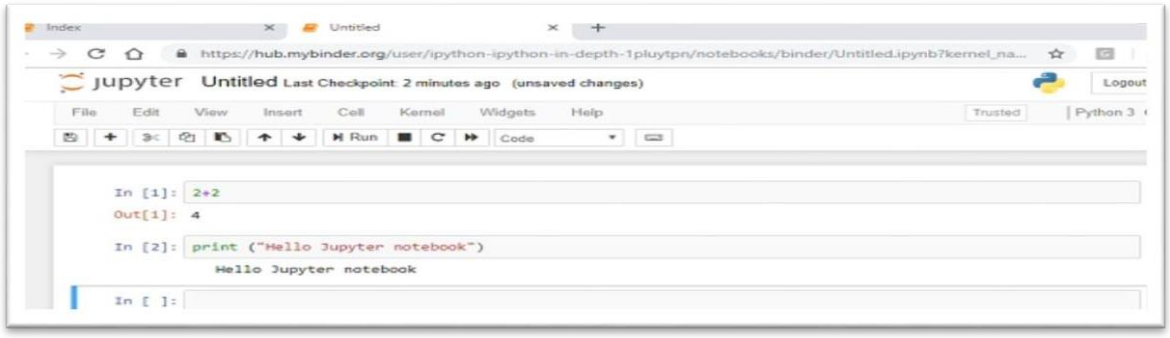

Fig:8 jupyter code

Jupyter notebooks are documents for technical and data science content. This tutorial provides an overview of Jupyter notebooks, their components, and how to use them.

We will explore notebooks using Datacamp Workspace, a hosted notebook service that provides all the functionality of Jupyter notebooks, along with functionality for connecting to databases, real-time collaboration, and publishing your work.

1.6.3 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 dashboar

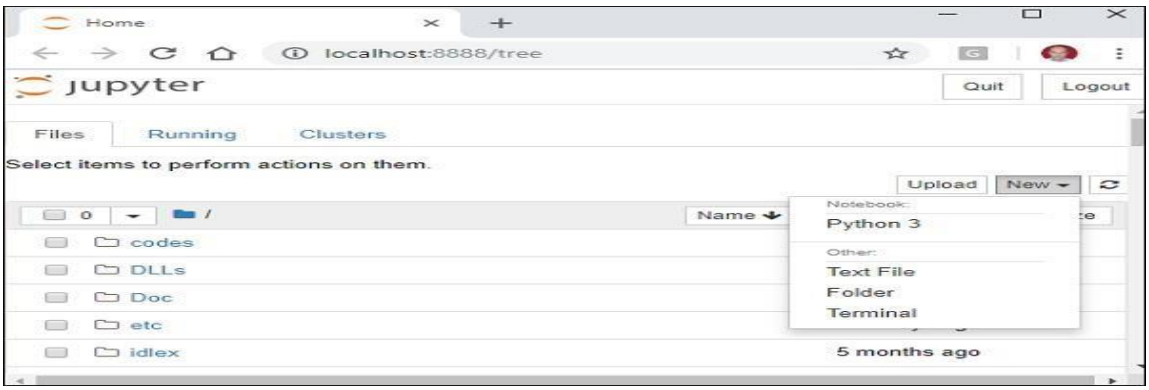

Fig:9 Jupyter Files

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 executeanyPythonexpressioninit.TheresultwillbedisplayedintheOutcell.

1.7.Tools

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

Operations using NumPy

Using NumPy, a developer can perform the following operations−

Mathematical and logical operations on arrays.

Fourier transforms and routines for shape manipulation.

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 Mat Lab, a popular platform for technical computing. How ever, 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)

1.7.2 Windows

Anaconda (from 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/) is available as a free as well as commercial distribution with full SciPy stack for Windows, Linux, and Mac.

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/)

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-matplotlib I python I python-notebook python-pandas python-sympy python-nose

For Fedora

Sudo yum install NumPy SciPy python-matplotlib I python 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. Trace back (mostrecent call last):

File "<pyshell#0>", line 1, in

<module>import numpy

Import Error: No module named' numpy'

Alternatively, NumPy package is imported using the following syntax–

Import numpy as np

1.7.3Pandas

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 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 Data Frame 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 datasets.

• Label-based slicing, indexing and sub setting of large datasets.

• 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) is a free Python distribution for SciPy stack. It is also available for Linux and Mac.

- *(https://www.enthought.com/products/canopy/)*isavailableasfreeaswellascommercial distributionwithfullSciPystack 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/)

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-matplotlib ipython ipython notebook Python-pandas python-sympy python-nose

For Fedora Users

Sudo yum install numpy scipy python-matplotlib ipython python-pandas sympy Python-nose atlas-devel

1.7.4Matplotlib

Matplotlib is one of the most popular Python packages used for data visualization. Itis a cross-platform library for making 2D plots from data in arrays. Mat plotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented AP Ithat helps in embedding plots in applications using Python GUI toolkits such as PyQt, Wx Python otT kinter. It can be used in Python and I Python shells, Jupyter notebook and web application servers also. 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. 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 use 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 forPython3.4) Redistributable packages need to be installed. If you are using Python 2.7 on a Mac, execute the following command

CHAPTER 2

SYSTEM IMPLEMENTATION

\

2.System Implementation

2.1 Frontend

2.1.1 About HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Advantages of HTML:

HTML is a simple language to learn and understand. HTML is the first and most important language that anyone learning web development will encounter. There are no complicated tags, and there is no case sensitivity with HTML.

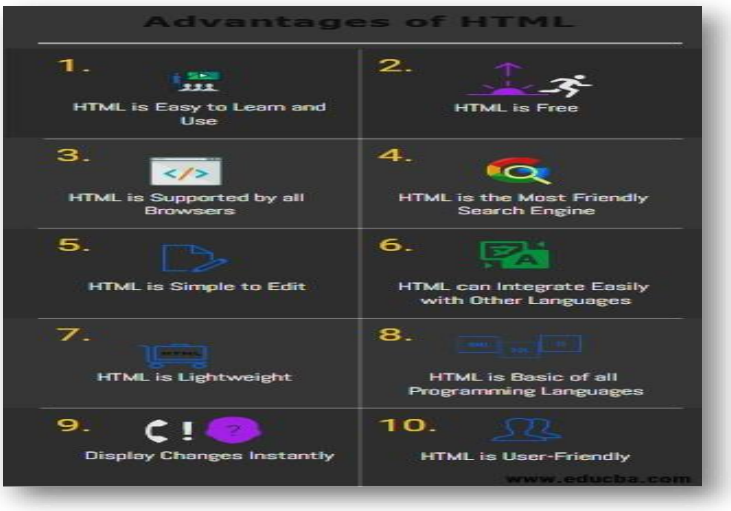

Fig 10: Advantages of HTML

One of the biggest advantages of using HTML is that it is free, and no special software is required. HTML does not require any plugins, so one should not have to deal with them when working with any software

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<meta http-equiv="X-UA-Compatible" content="IE=edge">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<title>HLDP</title>
	<link rel="stylesheet" href="{{url_for('static',filename='dist/css/output.css')}}">
	<script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite/1.6.3/flowbite.min.js"></script>
	<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-player.js"></script>
	<style>
	#loading {
		border: 16px solid #f3f3f3; /* Light grey */
		border-top: 16px solid #3498db; /* Blue */
		border-radius: 50%;
		width: 120px;
		height: 120px;
		animation: spin 2s linear infinite;
		display: none;
		margin-bottom: 20px;
		}
		#loader{
			display: none;
		}
		#load{
			display: flex;
			flex-direction: column;
			align-items: center;
		}
		@keyframes spin {
		0% { transform: rotate(0deg); }
		100% { transform: rotate(360deg); }
	  }
```

```
        #main {
                background-color: transparent;
                display: flex;
                flex-direction: column;
                align-items: center;
        }
        #head{
                color:#fff;
                margin-bottom:30px;
        }
  #content{
     display: flex;
     flex-direction: row;
     justify-content: space-between;
  }
        #player{
                margin-top:20px;
        }
        .bt{
                margin-bottom: 20px;
        }
        .bts{
                appearance: button;
                backface-visibility: hidden;
                background-color: #405cf5;
                border-radius: 6px;
                border-width: 0;
                box-shadow: rgba(50, 50, 93, .1) 0 0 0 1px inset,rgba(50, 50, 93, .1) 0 2px 5px 0,rgba(0, 0, 0,
.07) 0 1px 1px 0;
                box-sizing: border-box;
                color: #fff;
                cursor: pointer;
                font-family: -apple-system,system-ui,"Segoe UI",Roboto,"Helvetica Neue",Ubuntu,sans-serif;
                font-size: 100%;
```

```
                height: 44px;
                line-height: 1.15;
                margin: 12px 0 0;
                outline: none;
                overflow: hidden;
                padding: 0 25px;
                position: relative;
                text-align: center;
                text-transform: none;
                transform: translateZ(0);
                transition: all .2s,box-shadow .08s ease-in;
                user-select: none;
                -webkit-user-select: none;
                touch-action: manipulation;
                width: 100%;
                        }
        .bts:disabled {
        cursor: default;
        }
        .bts:focus {
        box-shadow: rgba(50, 50, 93, .1) 0 0 0 1px inset, rgba(50, 50, 93, .2) 0 6px 15px 0, rgba(0, 0, 0, .1) 0
2px 2px 0, rgba(50, 151, 211, .3) 0 0 0 4px;
        }
  #im{
     margin-right:40px;
     margin-bottom:10px;
  }
```

2.1.2 Tailwind CSS:

Tailwind CSS is a popular utility-first CSS framework that provides a set of pre-defined CSS classes that can be used to quickly style HTML elements. Instead of writing custom CSS code, Tailwind CSS provides pre-defined classes for common styles such as margins, padding, typography, colors, and more.

One of the benefits of using Tailwind CSS is that it helps developers write CSS code faster and more efficiently, as they don't have to remember the specific syntax for each style. Developers can simply add classes to their HTML elements to apply the desired styles.

Tailwind CSS is highly customizable, and developers can modify the default styles or create their own custom styles by configuring the framework using a configuration file.

Advantages of Tailwind CSS:

Tailwind CSS is often used in modern web development projects as it is easy to use, flexible, and can help developers write clean and maintainable code. It is also compatible with popular frontend frameworks such as React, Vue, and Angular.

Fig 11: Advantages of Tailwind CSS

One of the main advantages of using Tailwind CSS is it offers great room for customization. Though it comes with an in-built default configuration, you can override it from the config file. From the config file, you can customize all the elements, be it colors, spacing sizes, styling, themes, etc.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HLDP</title>
        <link rel="stylesheet" href="{{url_for('static',filename='dist/css/output.css')}}">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite/1.6.3/flowbite.min.js"></script>
    <style>
      .image-grid{
        --gap:16px;
        --num-cols:4;
        --row-height:300px;
```

```
    box-sizing: border-box;
    padding: var(--gap);

    display:grid;
    grid-template-columns: repeat(var(--num-cols) , 1fr);
    grid-auto-rows: var(--row-height);
    gap:var(--gap);
}
.image-grid > img {
    width:100%;
    height:100%;
    object-fit: cover;
}
.image-grid-col-2{
    grid-column: span 2;
}
.image-grid-row-2{
    grid-row: span 2;
}
@media screen and (max-width:1024px)
{
    .image-grid{
        --num-cols:2;
        --row-height:200px;
    }
}
#head{
    color:#fff;
    margin:30px;
        }
#cent{
    display: flex;
    flex-direction: column;
    align-items: center;
```

```
        width:100%;
        height:100vh;
    }
  </style>
</head>
<body class="bg-gray-800">
  <nav class="bg-white border-gray-200 px-2 sm:px-4 py-2.5 rounded dark:bg-gray-900">
            <div class="container flex flex-wrap items-center justify-between mx-auto">
            <a href="#" class="flex items-center">
                <img src="{{url_for('static',filename='dist/css/sample.jpg')}}" class="h-6 mr-3 sm:h-9" alt="HNPD Logo" />
                <span class="self-center text-xl font-semibold whitespace-nowrap dark:text-white">HNPD</span>
            </a>
            <div class="flex md:order-2">
                <button data-collapse-toggle="navbar-cta" type="button" class="inline-flex items-center p-2 text-sm text-gray-500 rounded-lg md:hidden hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-gray-200 dark:text-gray-400 dark:hover:bg-gray-700 dark:focus:ring-gray-600" aria-controls="navbar-cta" aria-expanded="false">
                <span class="sr-only">Open main menu</span>
                <svg class="w-6 h-6" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip-rule="evenodd"></path></svg>
            </button>
            </div>
            <div class="items-center justify-between hidden w-full md:flex md:w-auto md:order-1" id="navbar-cta">
            <ul class="flex flex-col p-4 mt-4 border border-gray-100 rounded-lg bg-gray-50 md:flex-row md:space-x-8 md:mt-0 md:text-sm md:font-medium md:border-0 md:bg-white dark:bg-gray-800 md:dark:bg-gray-900 dark:border-gray-700">
                <li>
                <a href="/" class="block py-2 pl-3 pr-4 text-white bg-blue-700 rounded md:bg-
```

```
transparent md:text-blue-700 md:p-0 dark:text-white" aria-current="page">Home</a>
                        </li>
                        <li>
                        <a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100
md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400
dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">About</a>
                        </li>
                        <li>
                        <a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100
md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400
dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video
Upload</a>
                        </li>
                        <li>
                        <a href="/stream" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100
md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400
dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video
Live</a>
                        </li>
        <li>
                                <a href="/image_upload" class="block py-2 pl-3 pr-4 text-gray-700 rounded
hover:bg-gray-100 md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white
dark:text-gray-400 dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-
gray-700">Image Upload</a>
                        </li>
                </ul>
                </div>
                </div>
        </nav>
   <div id="cent">
      <h1 id="head">Number Plates In The Video</h1>
      <div class="image-grid">
         {% for name in image_name %}
               <img src="{{name}}"  alt="number_plate" id="im">
```

```
        {% endfor %}
      </div>
   </div>
</body>
</html>
```

2.1.3 About JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. JavaScript was invented by Brendan Eichi in 1995, and became an ECMA standard in 1997.ECMA262 is the official name of the standard. ECMA Script is the official name of the language. These standards for JavaScript are the ECMA Script Language Specification (ECMA-262) and the ECMA Script Internationalization API specification (ECMA-402). The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402. And in cases where some proposals for new ECMA Script features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features.

Advantages of JavaScript

Since JavaScript is an 'interpreted' language, it reduces the time required by other programming languages like Java for compilation. JavaScript is also a client-side script, speeding up the execution of the program as it saves the time required to connect to the server.

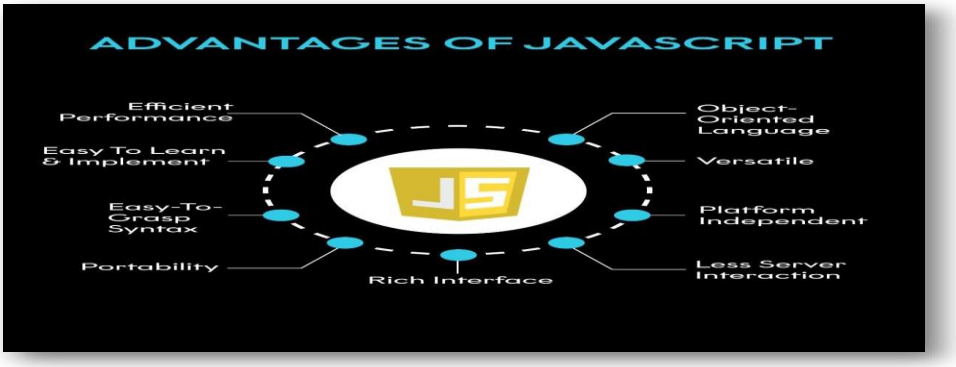

Fig 12: Advantages of JavaScript

JavaScript is easy to understand and learn. The structure is simple for the users as well as the developers. It is also very feasible to implement, saving developers a lot of money for developing dynamic content for the web.

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<meta http-equiv="X-UA-Compatible" content="IE=edge">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<title>HLDP</title>
	<link rel="stylesheet" href="{{url_for('static',filename='dist/css/output.css')}}">
	<script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite/1.6.3/flowbite.min.js"></script>
	<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-player.js"></script>
	<style>
	#loading {
		border: 16px solid #f3f3f3; /* Light grey */
		border-top: 16px solid #3498db; /* Blue */
		border-radius: 50%;
		width: 120px;
		height: 120px;
		animation: spin 2s linear infinite;
		display: none;
		margin-bottom: 20px;
		}
		#loader{
			display: none;
```

```
        }
        #load{
                display: flex;
                flex-direction: column;
                align-items: center;
        }
        @keyframes spin {
        0% { transform: rotate(0deg); }
        100% { transform: rotate(360deg); }
    }
#main {
        background-color: transparent;
        display: flex;
        flex-direction: column;
        align-items: center;
}
#head{
        color:#fff;
        margin-bottom:30px;
}
#player{

        margin-top:20px;
```

```
}
.bt{
    margin-bottom: 20px;
}
.bts{
    appearance: button;
    backface-visibility: hidden;
    background-color: #405cf5;
    border-radius: 6px;
    border-width: 0;
    box-shadow: rgba(50, 50, 93, .1) 0 0 0 1px inset,rgba(50, 50, 93, .1) 0 2px 5px 0,rgba(0, 0, 0, .07) 0 1px 1px 0;
    box-sizing: border-box;
    color: #fff;
    cursor: pointer;
    font-family: -apple-system,system-ui,"Segoe UI",Roboto,"Helvetica Neue",Ubuntu,sans-serif;
    font-size: 100%;
    height: 44px;
    line-height: 1.15;
    margin: 12px 0 0;
    outline: none;
    overflow: hidden;
    padding: 0 25px;
```

```
        position: relative;
        text-align: center;
        text-transform: none;
        transform: translateZ(0);
        transition: all .2s,box-shadow .08s ease-in;
        user-select: none;
        -webkit-user-select: none;
        touch-action: manipulation;
        width: 100%;
            }
    .bts:disabled {
    cursor: default;
    }
    .bts:focus {
    box-shadow: rgba(50, 50, 93, .1) 0 0 0 1px inset, rgba(50, 50, 93, .2) 0 6px 15px 0, rgba(0, 0, 0, .1) 0 2px 2px 0, rgba(50, 151, 211, .3) 0 0 0 4px;
    }
    </style>
    <script type="text/javascript">
  function loading(){
    document.getElementById('loading').style.display  = "block";
    document.getElementById('loader').style.display  = "block";
            document.getElementById('content').style.display = "none" ;
```

```
    }
  </script>
</head>
<body class="bg-gray-800">
    <nav class="bg-white border-gray-200 px-2 sm:px-4 py-2.5 rounded dark:bg-gray-900">
        <div class="container flex flex-wrap items-center justify-between mx-auto">
        <a href="/" class="flex items-center">
            <img src="{{url_for('static',filename='dist/css/sample.jpg')}}" class="h-6 mr-3 sm:h-9" alt="HNPD Logo" />
            <span class="self-center text-xl font-semibold whitespace-nowrap dark:text-white">HNPD</span>
        </a>
        <div class="flex md:order-2">
            <!-- <button type="button" class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm px-5 py-2.5 text-center mr-3 md:mr-0 dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800">Get started</button> -->
            <button data-collapse-toggle="navbar-cta" type="button" class="inline-flex items-center p-2 text-sm text-gray-500 rounded-lg md:hidden hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-gray-200 dark:text-gray-400 dark:hover:bg-gray-700 dark:focus:ring-gray-600" aria-controls="navbar-cta" aria-expanded="false">
<span class="sr-only">Open main menu</span>
<svg class="w-6 h-6" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip-rule="evenodd"></path></svg>
        </button>
```

```
</div>

<div class="items-center justify-between hidden w-full md:flex md:w-auto md:order-1" id="navbar-cta">

<ul class="flex flex-col p-4 mt-4 border border-gray-100 rounded-lg bg-gray-50 md:flex-row md:space-x-8 md:mt-0 md:text-sm md:font-medium md:border-0 md:bg-white dark:bg-gray-800 md:dark:bg-gray-900 dark:border-gray-700">

<li>

<a href="/" class="block py-2 pl-3 pr-4 text-white bg-blue-700 rounded md:bg-transparent md:text-blue-700 md:p-0 dark:text-white" aria-current="page">Home</a>

</li>

<li>

<a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">About</a>

</li>

<li>

<a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video Upload</a>

</li>

<li>

<a href="/stream" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video Live</a>

</li>
```

```
                    <li>
                        <a href="/image_upload" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Image Upload</a>
                    </li>
                </ul>
                </div>
                </div>
        </nav>
        <div id="main" class="flex h-screen items-center justify-center bg-gray-400">
            <h2 class="text-4xl font-bold dark:text-white" id="head">Helmet and License Plate Detection using video upload</h2>
            <div id="loader">
                <div id="load">
                    <div id="loading"></div>
                    <h1 id="head">Number Plates Are Identifying Please Wait</h1>
                </div>
            </div>
            <div id="content">
                <form method='POST' enctype="multipart/form-data" onsubmit="loading()">
                    {{form.hidden_tag()}}

{{form.file(class_="block w-full text-sm text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-
```

```
gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 bt")}}
                                {{form.submit(class_="bts")}}
                        </form>
                </div>
                <div id="player">
                        <lottie-player src="https://assets7.lottiefiles.com/packages/lf20_c6jminso.json" background="transparent"  speed="1"  style="width: 300px; height: 300px;"  loop  autoplay></lottie-player>
                </div>
        </div>
</body>
</html>
```

2.1.4 Lotti files for SVG:

Lottie is a powerful animation library created by Airbnb that enables designers and developers to easily add high-quality animations to their applications. It supports a wide range of file formats, including SVG.

To use Lottie with SVG files, you will need to convert the SVG to a JSON file using a tool like Adobe After Effects or the Lottie Files website. Once you have the JSON file, you can then load it into your application using the Lottie library.

Here are the steps to convert an SVG file to a Lottie JSON file using the Lottie Files website:

Go to https://lottiefiles.com/convert.

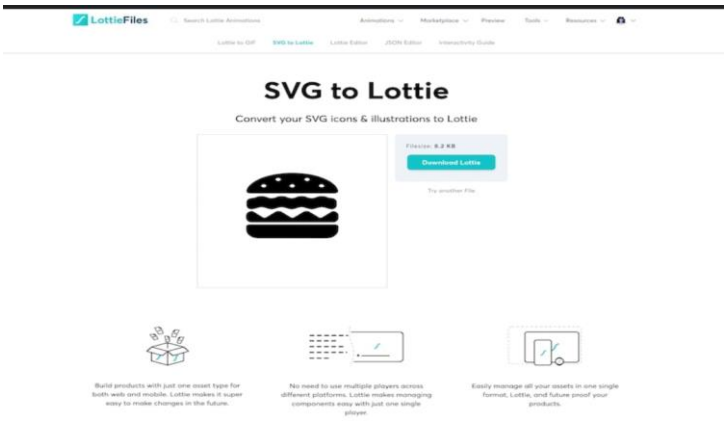

Fig 13: SVG to Lottie

Click on the "Choose File" button and select the SVG file you want to convert.

Choose the options you want, such as loop and autoplay settings.

Click the "Convert to Lottie JSON" button.

Download the resulting JSON file.

Once you have the JSON file, you can load it into your application using the Lottie library.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HLDP</title>
        <link rel="stylesheet" href="{{url_for('static',filename='dist/css/output.css')}}">
<script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite/1.6.3/flowbite.min.js"></script>
    <style>
        .image-grid{
            --gap:16px;
            --num-cols:4;
```

```
    --row-height:300px;
    box-sizing: border-box;
    padding: var(--gap);
    display:grid;
    grid-template-columns: repeat(var(--num-cols) , 1fr);
    grid-auto-rows: var(--row-height);
    gap:var(--gap);
}
.image-grid > img {
    width:100%;
    height:100%;
    object-fit: cover;
}
.image-grid-col-2{
    grid-column: span 2;
}
.image-grid-row-2{
    grid-row: span 2;
}
@media screen and (max-width:1024px)
{
    .image-grid{
        --num-cols:2;
        --row-height:200px;
    }
}
```

```
    #head{
        color:#fff;
        margin:30px;
}
    #cent{
        display: flex;
        flex-direction: column;
        align-items: center;
        width:100%;
        height:90vh;
    }
    #mi{
        width:80%;
        height:80%
    }
    #imm{
        width:100%;
        height:100%;
    }
  </style>
</head>
<body class="bg-gray-800">
  <nav class="bg-white border-gray-200 px-2 sm:px-4 py-2.5 rounded dark:bg-gray-900">
      <div class="container flex flex-wrap items-center justify-between mx-auto">
                                                  <a href="#" class="flex items-center">
                          <img
```

src="{{url_for('static',filename='dist/css/sample.jpg')}}" class="h-6 mr-3 sm:h-9" alt="HNPD Logo" />

HNPD

<!-- <button type="button" class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm px-5 py-2.5 text-center mr-3 md:mr-0 dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800">Get started</button> -->

<button data-collapse-toggle="navbar-cta" type="button" class="inline-flex items-center p-2 text-sm text-gray-500 rounded-lg md:hidden hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-gray-200 dark:text-gray-400 dark:hover:bg-gray-700 dark:focus:ring-gray-600" aria-controls="navbar-cta" aria-expanded="false">

Open main menu

<svg class="w-6 h-6" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip-rule="evenodd"></path></svg>

</button>

<ul class="flex flex-col p-4 mt-4 border border-gray-100 rounded-lg bg-gray-50 md:flex-row md:space-x-8 md:mt-0 md:text-sm md:font-medium md:border-0 md:bg-white dark:bg-gray-800 md:dark:bg-gray-900 dark:border-gray-700">

Home

<a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent

```
md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700
dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">About</a>
      </li>
      <li>
      <a href="/" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent
md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-700
dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video Upload</a>
     </li>
     <li>
     <a href="/stream" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100 md:hover:bg-
transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400 dark:hover:bg-gray-
700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Video Live</a>
     </li>
     <li>
     <a href="/image_upload" class="block py-2 pl-3 pr-4 text-gray-700 rounded hover:bg-gray-100
md:hover:bg-transparent md:hover:text-blue-700 md:p-0 md:dark:hover:text-white dark:text-gray-400
dark:hover:bg-gray-700 dark:hover:text-white md:dark:hover:bg-transparent dark:border-gray-700">Image
Upload</a>
    </li>
   </ul>
   </div>
   </div>
   </nav>
   <div id="cent">
<h2 class="text-4xl font-bold dark:text-white" id="head">Live Streaming</h2>
  <div id="mi">
        <img src="{{url_for('video')}}" alt="video" id="imm">
      </div>
```

```
    </div>
    <!-- <div class="container grid grid-cols-3 gap-2 mx-auto">
        {% for name in image_name %}
        <div class="w-full rounded">
            <img src="{{name}}" height="250px" width="250px" alt="number_plate">
        </div>
        {% endfor %}
    </div> -->
</body>
</html>
```

2.2 Backend

2.2.1 About Python

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Below are some facts about Python Programming Language:

1. Python is currently the most widely used multi-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like –

   Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
5. The biggest strength of Python is huge collection of standard library which can be used for the following:

   - Machine Learning
   - GUI Applications (like Kivy, Tkinter, PyQt etc.)

- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test framework

Advantages of Python

Python is one of the most popular programming languages in the world. In Stack Overflow's 2021 Developer Survey, 48% of respondents said they work with Python. When other respondents were asked which technology they had a desire to learn, Python ranked first as the most wanted technology among developers.

Fig 14: Advantages of Python

Python's large open-source community means Pythonistas can enjoy strong peer support and helpful documentation. If you ever run into a roadblock, you can always check out Python forums or meetups to get help from other Python developers. This community support can be especially helpful if Python is your first programming language

2.2.2 Flask

Flask is python-based web framework that can be easily installed using pip. Many tools, libraries, and predefined standards come with this framework, so it becomes easier to build a web application or APIs. It is considered a micro-framework. Flask is a lightweight and flexible Python web framework that provides developers with tools and libraries to build web applications quickly and efficiently. It is classified as a micro
web framework because it does not require particular tools or libraries, making it easy to get started with and customizable to fit your specific needs.

Flask provides features such as URL routing, templates, request/response handling, and integration with various web technologies. It also supports extensions and plugins that can be easily added to extend its functionality.

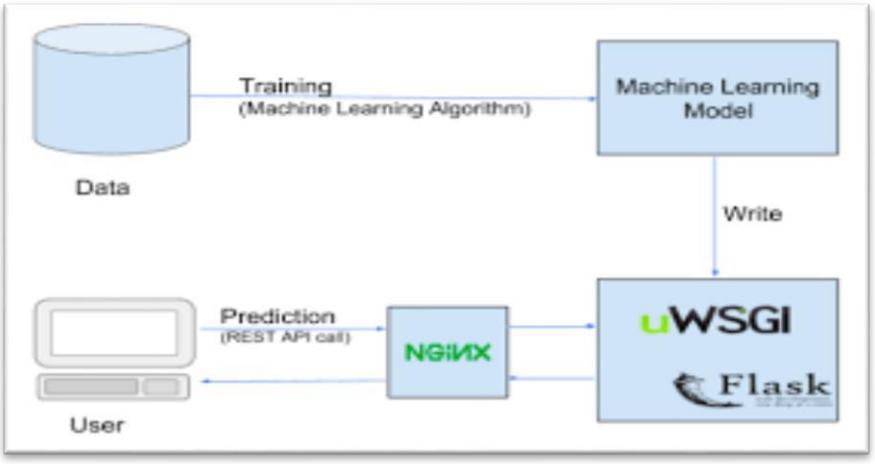

Fig 15: Flask Framework

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc.

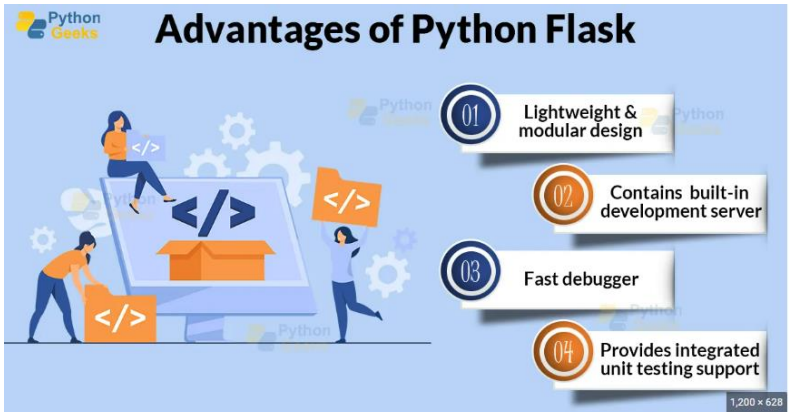

Fig 16: Advantages of Flask

Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. If you want to make an app that starts small, but has the potential to grow quickly and in directions you haven't completely worked out yet, then it's an ideal choice. Its simplicity of use and few dependencies enable it to run smoothly even as it scales up and up.

This is the core feature of Flask, and one of its biggest advantages. To paraphrase one of the principles of the Zen of Python, simplicity is better than complexity, because it can be easily rearranged and moved around.

Chapter 3

LITERATURE SURVEY

3. Literature Survey

3.1 Literature Survey

As the bikers in our country are increasing, the road mishaps 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”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 includehead 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.

CHAPTER 4

FLOW DIAGRAMS

4. Data Flow Diagram

4.1 Data Flow Diagram:

A data flow diagram (DFD) is a graphical representation of a system or process that shows how data moves within it. It is a tool used by system analysts and developers to model, analyze, and design information systems. A DFD consists of four main components:

Entities: These are external objects that interact with the system. Entities can be people, organizations, or other systems.

Processes: These are the activities that take place within the system. Processes can be manual or automated.

Data stores: These are the places where data is stored within the system. Data stores can be physical or virtual.

Data flows: These are the pathways through which data moves within the system. Data flows can be represented by arrows that connect entities, processes, and data stores.

.

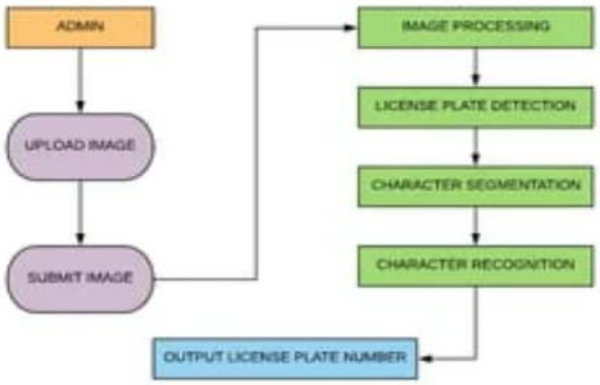

Fig 15: Data flow diagram

DFDs can be used to identify bottlenecks, redundancies, and other inefficiencies within a system. They can also be used to model complex systems and to communicate system requirements to stakeholders

Data Collection:

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.thisdata-setconsistsof13818samples

Data Preparation:

Data preparation involves the cleaning and exploration of data to find relationships among the features of the data. In the 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.

4.2 UML Diagrams:

4.2.1 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. The main components of a use case diagram are actors, use cases, and the relationships between them. Actors represent the different types of users or systems that interact with the system, while use cases represent the specific tasks or functions that can be performed within the system. Relationships between actors and use cases show the interactions and dependencies between them

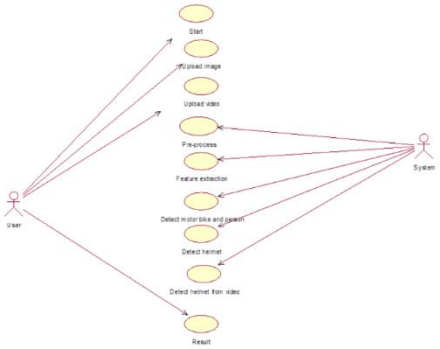

Fig 16 : Use Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is just a special kind of diagram and shares the same common properties as do all other diagrams, i.e. a name and graphical contents that are a projection into a model. Use case diagrams are useful for modeling the requirements and functionality of a system, and can help to identify potential issues and areas for improvement. They are also useful for communicating the requirements and functionality of a system to stakeholders and developers.

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.

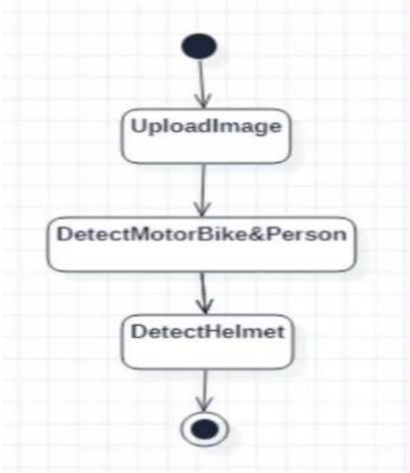

Fig 17: Activity Diagram

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

Sequence Diagram:

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

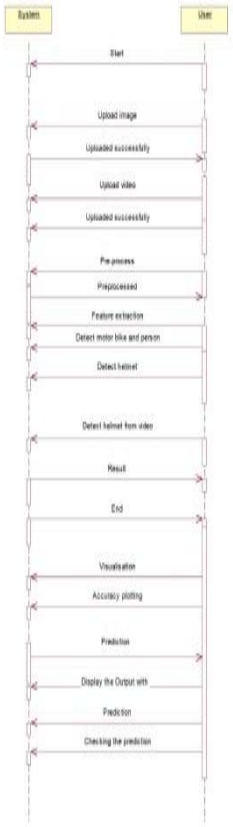

Fig 18: Sequence diagram

Unified Modelling Language (UML) is a modeling language in the field of software engineering which aims to set standard ways to visualize the design of a system. UML guides the creation of multiple types of diagrams such as interaction, structure and behavior diagrams. A sequence diagram is the most commonly used

interaction diagram. Interaction diagram – An interaction diagram is used to show the interactive behavior of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in 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

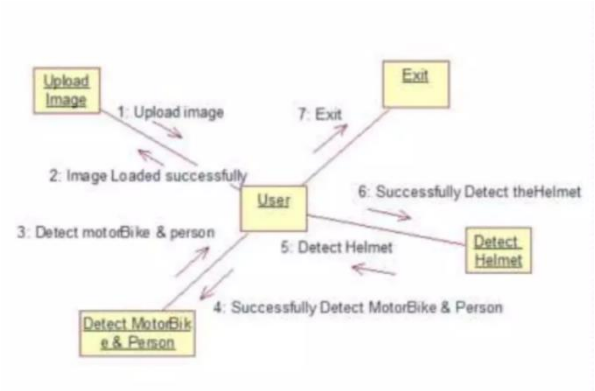

Fig 19: Collaboration diagram

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

CHAPTER 5

SYSTEM ANALYSIS

5. SYSTEM ANALYSIS

5.1ProblemStatement

As the bikers in our country are increasing ,the road mishaps 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..

5.2 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.

5.3 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.

5.4 Architecture of proposed system:

For detecting helmet and recognizing license plate. The whole process consists of five steps. For detection of the helmet, we first insert the photo of the motorcyclist and in the next step, we process the video and remove the background. The third step involves the segmentation of two-wheelers. Next step the system detects persons without a helmet. In the last step, we detect the license plate and display the license plate number from the image. Here the head is classified as wearing a helmet or not based on various features obtained from the segmented head region moving objects can be detected using adaptive background subtraction. ViBe background modelling algorithm can also be applied to detect moving objects. The Canny edge detection
algorithm is used to get segmented moving objects.

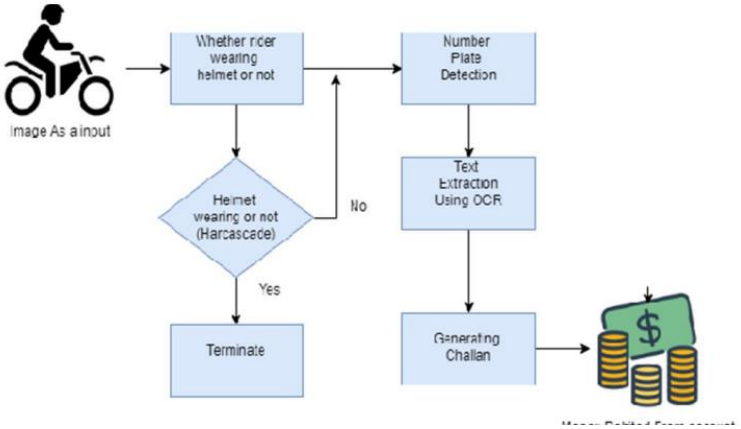

Fig 20: Architecture of proposed system:

5.5 Algorithm

5.5.1 YOLO ALGORITHM:

It deals with localizing a region of interest within an image and classifying this region like a typical image classifier. One image can include several regions of interest pointing to different objects. This makes object detection a more advanced problem of image classification.

YOLO (You Only Look Once) is a popular object detection model known for its speed and accuracy. It was first introduced by Joseph Redmon et al. in 2016 and has since undergone several iterations, the latest being YOLO v7.

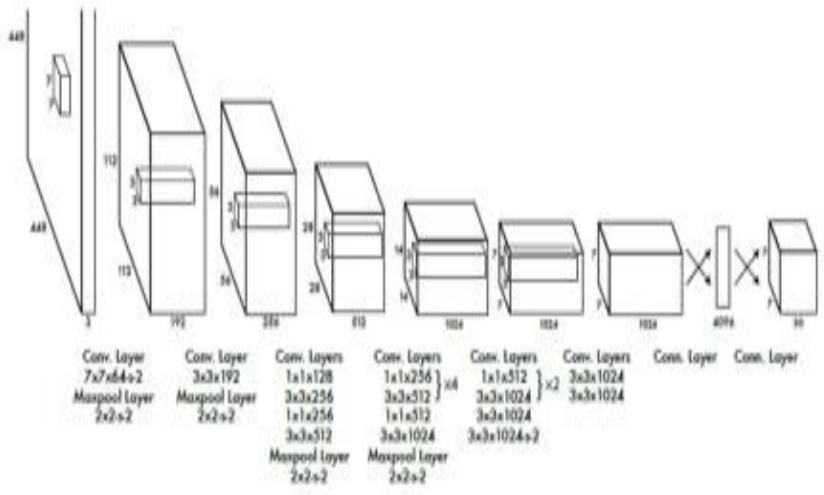

Figure 1: Yolo architecture

YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides

the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.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.

5.5.2 Convolutional Neural Network (CNN):

A convolutional neural network (CNN or convnet) is a subset of machine learning. It is one of the various types of artificial neural networks which are used for different applications and data types. A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice. This makes them highly suitable for computer vision (CV) tasks and for applications where object recognition is vital, such as self-driving cars and facial recognition. The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

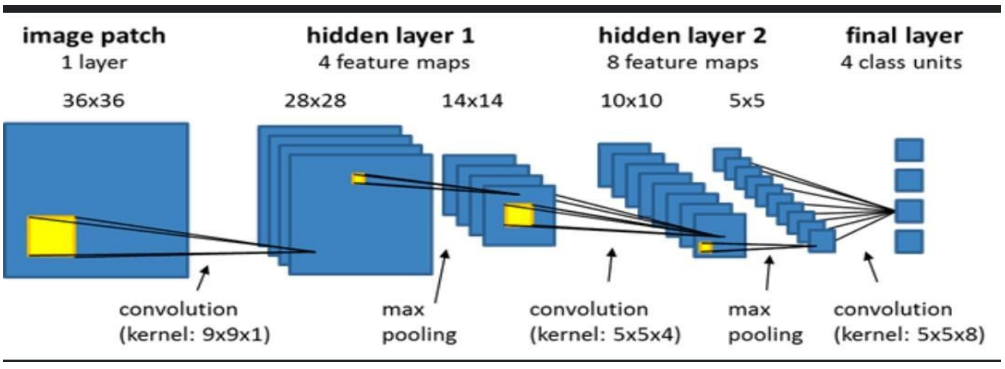

Fig 22: Convolutional Neural Network

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptions are usually fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.)

Developing robust datasets also increases the probability that CNNs will learn thegeneralized principles that characterize a given dataset rather than the biases of a poorly-populated set[10] CNN take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters.

OpenCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

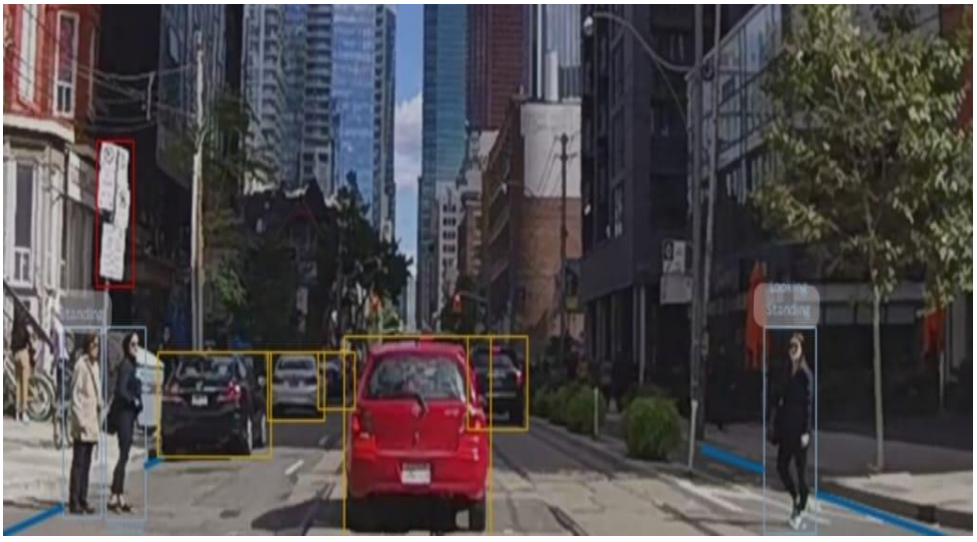

Fig 23: OpenCv

OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and machine learning software library. It was first developed by Intel in 1999 and has since been maintained by a community of developers. OpenCV provides a wide range of functions and algorithms to process and analyze images and videos, including object detection, face recognition, feature detection, image segmentation, and more.

OpenCV is written in C++ and can be used in several programming languages such as Python, Java, and MATLAB. It also supports multiple operating systems, including Windows, Linux, macOS, iOS, and Android.

TensorFlow:

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level KerasAPI, which makes getting started with TensorFlow and machine learning easy If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition**.**

TensorFlow is an open-source machine learning framework developed by Google. It was released in 2015 and has since become one of the most popular frameworks for building and training machine learning models.

TensorFlow provides a flexible and powerful platform for implementing and testing various machine learning algorithms, including deep neural networks, convolutional neural networks, and recurrent neural networks. It also includes tools for data pre-processing, model visualization, and distributed computing.

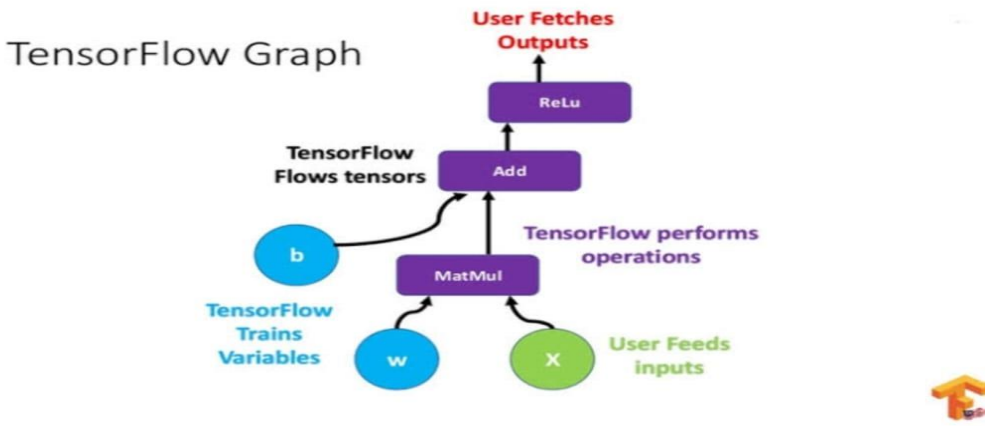

Fig 24: TensorFlow Graph

One of the key features of TensorFlow is its ability to build and train models using a symbolic computational graph. This allows for efficient computation and automatic differentiation, which is crucial for training deep neural networks. In addition to its core features, TensorFlow also provides a high-level API called Keras, which makes it easy to build and train neural networks with minimal code. TensorFlow can be used for a wide range of applications, including computer vision, natural language processing, and reinforcement learning

Optical Character Recognition (OCR):

The procedure of transforming images of written, printed, or handwritten text into device-encoded text is referred to as optical character recognition (OCR). Acquiring information, performing basic processing, and categorizing and separating characteristics are the three basic stages. Pattern recognition and feature detection are the two

fundamental methods of extracting features in OCR. By attempting to identify white pixel rows with black pixels in between, OCR involves identifying every single character. A neural network undergoes training to generate a generalized device>text translation. into the subsequent workshops. It may be critical to evaluate the data quality and boost the data even more. There are free online OCR libraries, like the Tesseract library, which can be leveraged to boost efficiency.

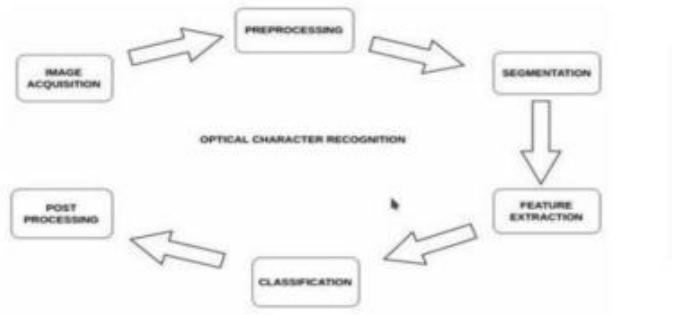

Fig 25: Optical Character Recognition

Optical Character Recognition (OCR) is a technology that involves the use of software algorithms to recognize and extract text from images or scanned documents. OCR software analyzes an image or a scanned document and then converts the text into an editable digital format that can be searched, edited, and stored electronically.
OCR technology has numerous applications, including document management, data entry automation, and image recognition. For example, OCR can be used to convert printed books and documents into electronic formats, which can then be easily searched, indexed, and stored. It can also be used to extract data from invoices, receipts, and other business documents, which can help automate data entry tasks.

Object Detection:

CCTV cameras present at traffic signals should detect helmet, and if any person found without helmet is checked into the database of RTO, a mail should be sent to him/her regarding fine.to prove the concept, a prototype model will be developed using laptop wed cam and a window based application.to detect and recognize number, a standard Indian number plate image will be uploaded

Object detection is a task in computer vision that involves detecting and localizing objects within an image or a video stream. In deep learning, object detection is typically performed using convolutional neural networks (CNNs) that are trained on large datasets of labeled images.

Fig 26: Object Detection

There are several popular deep learning models that are used for object detection, including:

Faster R-CNN: This model uses a region proposal network (RPN) to generate candidate object bounding boxes, which are then refined using a second network to produce final object detections.

Mask R-CNN: This model extends Faster R-CNN to include instance segmentation, which involves not only detecting objects but also segmenting them into individual pixels. YOLO (You Only Look Once): This model processes the entire image at once and outputs bounding boxes and class probabilities for all objects in a single pass through the network, making it very fast and efficient. SSD (Single Shot Detector): This model is similar to YOLO in that it detects objects in a single pass through the network, but it uses a series of convolutional filters to predict the location and size of objects at different scales. Object detection has many applications, including autonomous driving, security and surveillance, medical imaging, and robotics

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

This research paper introduces the integration of a helmet and registration plate detection system into a Flask website. For object detection, the system leverages a deep learning model based on the YOLO approach. That was designed to be simple to deploy and utilize. The website's design is simple and responsive because of the Tailwind CSS styling. The equipment could be modified according to individual branding demands and achieves superior precision when recognizing helmets and licences plates in photographs, videos, and live streaming data. The system can also be employed for monitoring traffic, road safety, and law enforcement. The

system that was recently put in place is a prototype that, with approval from the right people, can be expanded to process regular traffic video. A big database is set up for maintaining track of violators and their challan payment history, and a high-resolution camera is advised to maintain precision and accuracy. The system proposed in this

research offers an efficient solution for the detection of helmets and license plates, and the integration with a Flask web application and Tailwind CSS styling enhances the system's usability. The system's potential applications include road safety and traffic management. However, there is room for further improvement to increase the system's accuracy and add new features

Source code

Object_detection.py

```
from flask import Flask,render_template,Response
from flask_wtf import FlaskForm
from wtforms import FileField , SubmitField
from werkzeug.utils import secure_filename
import os
from wtforms.validators import InputRequired
import cv2
import numpy as np
import imutils
from tensorflow.keras.models import load_model
from datetime import date
import math

app =  Flask(__name__)
app.config['SECRET_KEY'] = 'supersecretkey'
app.config['UPLOAD_FOLDER'] = 'static/files'
camera = cv2.VideoCapture(0)

weights0_path = './input/detect-person-on-motorbike-or-scooter/yolov3-obj_final.weights'
configuration0_path = './input/detect-person-on-motorbike-or-scooter/yolov3_pb.cfg'

probability_minimum = 0.5
threshold = 0.3

COLORS = [(0,255,0),(0,0,255)]

os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'
```

```
net = cv2.dnn.readNet("yolov3-custom_7000.weights", "yolov3-custom.cfg")
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)

network0 = cv2.dnn.readNetFromDarknet(configuration0_path, weights0_path)
layers_names0_all = network0.getLayerNames()
layers_names0_output = [layers_names0_all[i-1] for i in network0.getUnconnectedOutLayers()]
labels0 = open('./input/detect-person-on-motorbike-or-scooter/coco.names').read().strip().split('\n')

model = load_model('helmet-nonhelmet_cnn.h5')
print('model loaded!!!')

layer_names = net.getLayerNames()
output_layers = net.getUnconnectedOutLayersNames()

# for helmet
weights1_path = './input/helmet-detection-yolov3/yolov3-helmet.weights'
configuration1_path = './input/helmet-detection-yolov3/yolov3-helmet.cfg'

network1 = cv2.dnn.readNetFromDarknet(configuration1_path, weights1_path)
layers_names1_all = network1.getLayerNames()
layers_names1_output = [layers_names1_all[i-1] for i in network1.getUnconnectedOutLayers()]
labels1 = open('./input/helmet-detection-yolov3/helmet.names').read().strip().split('\n')

np.random.seed(42)
colours0 = np.random.randint(0,255,size=(len(labels0),3),dtype='uint8')
colours1 = np.random.randint(0,255,size=(len(labels1),3),dtype='uint8')

def helmet(imagee,l1):
    tempp = imagee
    blob1 = cv2.dnn.blobFromImage(tempp,1/255.0,(416,416),swapRB=True,crop=False)
    blob_to_show = blob1[0,:,:,:].transpose(1,2,0)
    network1.setInput(blob1)
```

```
    output_from_network1 = network1.forward(output_layers)
    temp = tempp
    height,width = tempp.shape[:2]
    cc = 0
    for result in output_from_network1:
        for detection in result:
            scores = detection[5:]
            class_current=np.argmax(scores)
            confidence_current=scores[class_current]
            if confidence_current>probability_minimum:
                box_current=detection[0:4]*np.array([width,height,width,height])
                x_center,y_center,box_width,box_height=box_current.astype('int')
                x_min=int(x_center-(box_width/2))
                y_min=int(y_center-(box_height/2))
                color = [int(c) for c in COLORS[class_current]]
                cv2.rectangle(temp, (x_min, y_min), (x_min + box_width, y_min + box_height), color, 7)
                cc = cc+1

    if cc>=1:
        return 1
    else:
        return 0


def check(imagee,l1,directory):
                tempp = imagee
                blob1 = cv2.dnn.blobFromImage(tempp,1/255.0,(416,416),swapRB=True,crop=False)
                blob_to_show = blob1[0,:,:,:].transpose(1,2,0)
                net.setInput(blob1)
                outs = net.forward(output_layers)
                temp = tempp
                height,width = tempp.shape[:2]
                for out in outs:
                    for detection in out:
```

```
                scores = detection[5:]
                class_id = np.argmax(scores)
                confidence = scores[class_id]
                if confidence > 0.4:
                    center_x = int(detection[0] * width)
                    center_y = int(detection[1] * height)
                    w = int(detection[2] * width)
                    h = int(detection[3] * height)
                    x = int(center_x - w / 2)
                    y = int(center_y - h / 2)
                    color = [int(c) for c in COLORS[class_id]]
                    if class_id==0:
                        helmet_roi = temp[max(0,y):max(0,y)+max(0,h)//4,max(0,x):max(0,x)+max(0,w)]
                    else:
                        x_h = x-60
                        y_h = y-350
                        w_h = w+100
                        h_h = h+100
                        cv2.rectangle(temp, (x, y), (x + w, y + h), color, 2)

class UploadFileForm(FlaskForm):
    file = FileField("File",validators=[InputRequired()])
    submit = SubmitField("Upload File")

def generate_frames():
    global camera
    camera = cv2.VideoCapture(0)
    count  = 0
    while True:
        success,frame=camera.read()
        if not success:
            break
        else:
            img = imutils.resize(frame,height=500)
```

```
parent = os.getcwd().replace("\\","/") + '/static/numberplates'
path = parent+'/'+'livestream'
image_names = os.listdir(path)
li = len(image_names)
COLORS = [(0,255,0),(0,0,255)]
layer_names = net.getLayerNames()
output_layers = net.getUnconnectedOutLayersNames()
height, width = img.shape[:2]
blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
outs = net.forward(output_layers)
confidences = []
boxes = []
classIds = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.3:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            classIds.append(class_id)
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

for i in range(len(boxes)):
    if i in indexes:
        x,y,w,h = boxes[i]
```

```
            color = [int(c) for c in COLORS[classIds[i]]]
            if classIds[i]==0:
                helmet_roi = img[max(0,y):max(0,y)+max(0,h)//4,max(0,x):max(0,x)+max(0,w)]
            else:
                x_h = x-60
                y_h = y-350
                w_h = w+100
                h_h = h+100
                cv2.rectangle(img, (x, y), (x + w, y + h), color, 7)
                if y_h>0 and x_h>0:
                    h_r = img[y_h:y_h+h_h , x_h:x_h +w_h]
                    c = helmet_or_nohelmet(h_r)
                    cv2.putText(img,['helmet','no-helmet'][c],(x,y-
100),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),2)
                    cv2.rectangle(img, (x_h, y_h), (x_h + w_h, y_h + h_h),(255,0,0), 10)
                    if c==1:
                        imagg = img[y:y+h,x:x+w]
                        today = date.today()
                        today = str(today)
                        path = path+'/'+today
                        if not os.path.exists(path):
                            os.mkdir(path)
                        cv2.imwrite('{}_{}.{}'.format(os.path.join(path,'img'), str(li), 'jpg'), imagg)

        cv2.waitKey(0)
        cv2.destroyAllWindows()
        ret,buffer=cv2.imencode('.jpg',img)
        frame=buffer.tobytes()

      yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

def helmet_or_nohelmet(helmet_roi):
    try:
```

```
        helmet_roi = cv2.resize(helmet_roi, (224, 224))
        helmet_roi = np.array(helmet_roi,dtype='float32')
        helmet_roi = helmet_roi.reshape(1, 224, 224, 3)
        helmet_roi = helmet_roi/255.0
        return int(model.predict(helmet_roi)[0][0])
    except:
            pass

@app.route('/',methods = ["GET","POST"])
@app.route('/home',methods = ["GET","POST"])
def home():
    form = UploadFileForm()
    global camera
    camera.release()
    if form.validate_on_submit():
        file = form.file.data
        directory = file.filename
        seperator = '.'
        directory = directory.split(seperator,1)[0]
        parent = os.getcwd().replace("\\","/") + '/static/numberplates'
        path = parent+'/'+directory
        if not os.path.exists(path):
            os.mkdir(path)
        file.save(os.path.join(os.path.abspath(os.path.dirname(__file__)),app.config['UPLOAD_FOLDER'],secure
_filename(file.filename)))

        cap =
cv2.VideoCapture(os.path.join(os.path.abspath(os.path.dirname(__file__)),app.config['UPLOAD_FOLDER'],se
cure_filename(file.filename)))

        COLORS = [(0,255,0),(0,0,255)]
        xii1 = 0

        layer_names = net.getLayerNames()
```

```
output_layers = net.getUnconnectedOutLayersNames()


fourcc = cv2.VideoWriter_fourcc(*"XVID")
writer = cv2.VideoWriter('output.avi', fourcc, 5,(888,500))

ret = True
xii = 1

while ret:

    ret, img = cap.read()
    if img is None or xii1==200:
        break
    img = imutils.resize(img,height=500)
    height, width = img.shape[:2]

    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

    net.setInput(blob)
    outs = net.forward(output_layers)

    confidences = []
    boxes = []
    classIds = []

    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.3:
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
```

```
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            classIds.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

for i in range(len(boxes)):
    if i in indexes:
        x,y,w,h = boxes[i]
        color = [int(c) for c in COLORS[classIds[i]]]
        if classIds[i]==0: #bike
            helmet_roi = img[max(0,y):max(0,y)+max(0,h)//4,max(0,x):max(0,x)+max(0,w)]
        else: #number plate
            x_h = x-60
            y_h = y-350
            w_h = w+100
            h_h = h+100
            if y_h>0 and x_h>0:
                h_r = img[y_h:y_h+h_h , x_h:x_h +w_h]
                c = helmet_or_nohelmet(h_r)
                if c==1:
                    imagg = img[y:y+h,x:x+w]
                    cv2.imwrite('{}_{}.{}'.format(os.path.join(path,'img'), str(xii), 'jpg'), imagg)
                    xii = xii + 1

writer.write(img)
xii1 = xii1+1
```

```
            if cv2.waitKey(1) == 27:
                break

        writer.release()
        cap.release()
        cv2.waitKey(0)
        cv2.destroyAllWindows()
        image_names = os.listdir(path)
        for i in range(len(image_names)):
            image_names[i] = '../static/numberplates/'+ directory+'/'+image_names[i]
        return render_template('images.html',image_name = image_names)
    return render_template('index.html',form = form)



@app.route('/image_upload',methods = ["GET","POST"])
def image_up():
    form = UploadFileForm()
    global camera
    camera.release()
    ans = ""
    if form.validate_on_submit():
        file = form.file.data
        directory = file.filename
        seperator = '.'
        directory = directory.split(seperator,1)[0]
        parent = os.getcwd().replace("\\","/") + '/static/numberplates'
        path = parent+'/'+directory
        if not os.path.exists(path):
            os.mkdir(path)
        file.save(os.path.join(os.path.abspath(os.path.dirname(__file__)),app.config['UPLOAD_FOLDER'],secure
_filename(file.filename)))
        pa  =
os.path.join(os.path.abspath(os.path.dirname(__file__)),app.config['UPLOAD_FOLDER'],secure_filename(file.
```

```
filename))
    img = cv2.imread(pa)

    image_input = img
    blob = cv2.dnn.blobFromImage(image_input,1/255.0,(416,416),swapRB=True,crop=False)
    blob_to_show = blob[0,:,:,:].transpose(1,2,0)
    network0.setInput(blob)
    output_from_network0 = network0.forward(layers_names0_output)

    h,w = image_input.shape[:2]
    height,width = image_input.shape[:2]

    l1 = 0
    for result in output_from_network0:
        for detection in result:
            scores = detection[5:]
            class_current=np.argmax(scores)
            confidence_current=scores[class_current]
            l1=scores[class_current]
            if confidence_current>=probability_minimum:
                box_current=detection[0:4]*np.array([w,h,w,h])
                x_center,y_center,box_width,box_height=box_current.astype('int')
                x_min=int(math.ceil(x_center-(box_width/2)))
                y_min=int(math.ceil(y_center-(box_height/2)))
                imagee = image_input[y_min:y_min+int(box_height),x_min:x_min+int(box_width)]
                if helmet(imagee,l1)==0:
                    check(imagee,l1,directory)
                    pa1 = './static/numberplates/'+str(directory)+'/img_box'+str(l1)+'.jpg'
                    l1 = l1 + 1
                    cv2.imwrite(pa1, imagee)

    image_names = os.listdir('./static/numberplates/'+directory+'/')
    for i in range(len(image_names)):
        image_names[i] = '../static/numberplates/'+ directory+'/'+image_names[i]
```

```
        cv2.waitKey(0)
        cv2.destroyAllWindows()



        return render_template('images.html',image_name = image_names)
    return render_template('image_upload.html',form = form)

@app.route('/stream',methods = ["GET","POST"])
def video_page():
    return render_template('video.html')

@app.route('/video',methods = ["GET","POST"])
def video():
    return Response(generate_frames(),mimetype='multipart/x-mixed-replace;boundary=frame')

if __name__ == '__main__':
    app.run(debug=True)
```

CHAPTER 6

REQUIREMENT AND SPECIFICATION

6.REQUIREMENT AND SPECIFICATION

6.1System Specifications

6.1.1 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

6.1.2 Software requirements

Operating System	:	Windows
Coding Language	:	Python 3.6.8

Chapter 7

Model Screens

7.1 Screenshots

A home page (or homepage) is the main web page of a website.[1] The term may also refer to the start page shown in a web browser when the application first opens.[2] Usually, the home page is located at the root of the website's domain or subdomain.

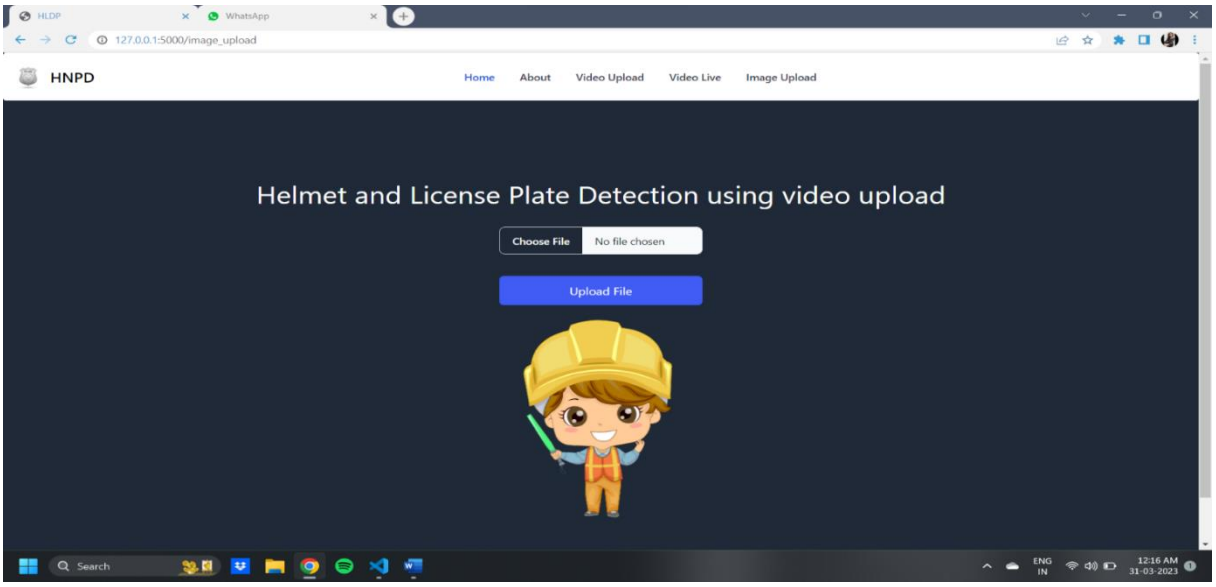

Fig 27: Home Screen

Case 01 Image upload:

After getting the input photographs from the surveillance tape.the CNN algorithm is used to locate the People within the image who are not wearing safety helmets, and outcomes are displayed as a number plate and the license number for those who were identified. their driving license number is shown here in a red box.

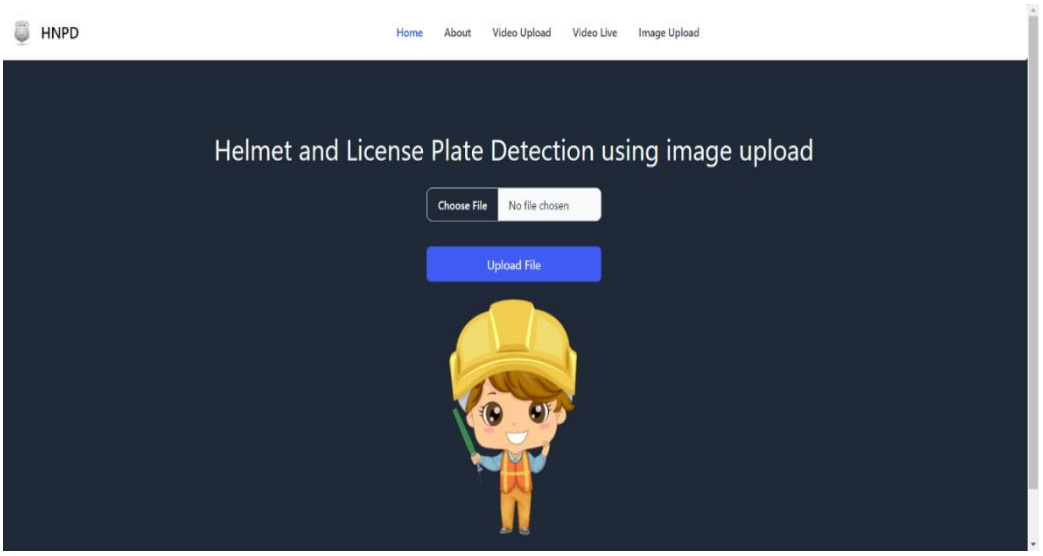

Fig 28: after uploaded image

The proposed model was able to accurately identify the Region of Interest (ROI) needed for image enhancement, and a batch size of 28 was used with a subdivision reduction to 16 for faster training.

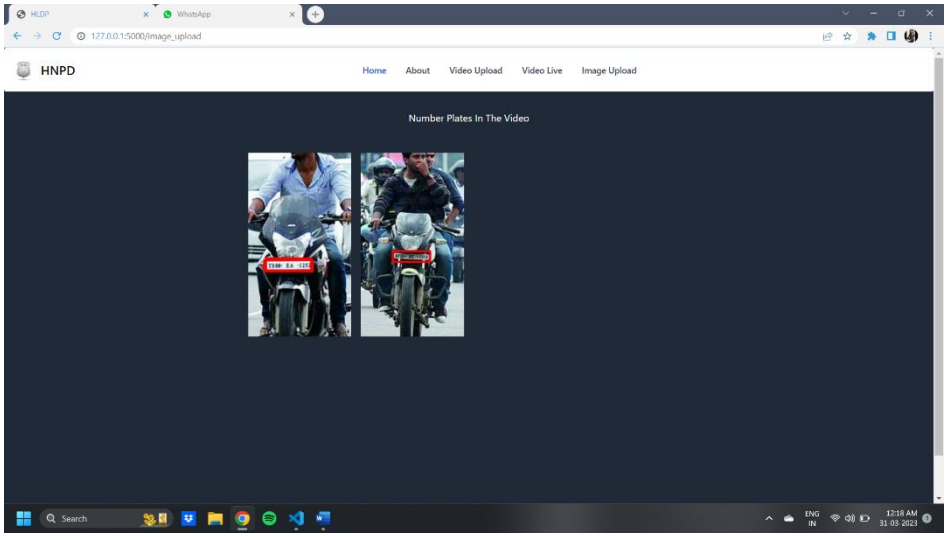

Fig 29: Output of image upload

license plate detection phase involved only one class and a fixed filter size of 18. The training process utilized previously trained weights from the darknet-53 convolutional layer

Case 02: Video upload

Here, there are videos who were recorded from CCTV footage. CNN algorithm is used for identifying photos, and in these uploaded videos the algorithms used recognize each frame in the video and take the frame where the two wheelers are not wearing helmets, restoring the output of the licenses number for each of the two-wheelers

in the video who are not wearing helmets.

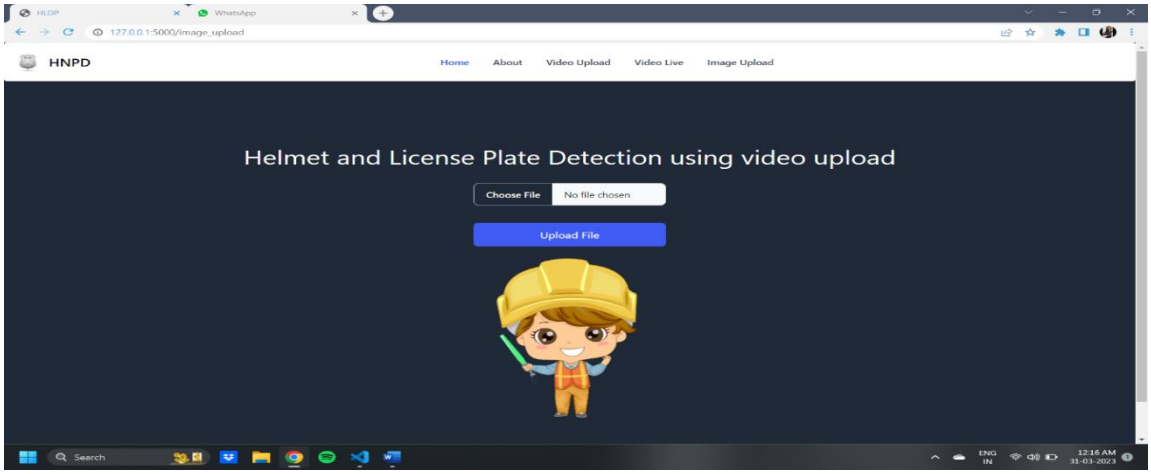

Fig 30: Video upload

following this procedure, the all the permission numbers to earn those who were two-wheelers are displayed in the red box who are not wearing helmets. The system underwent testing with different images and videos under varying environmental conditionsit was observed that the helmet detection model accurately detected helmets 94.2% of the time, while the license plate detection model was able to detect license plates with 97.5% accuracy. Additionally, the system underwent live streaming testing through a camera mounted on a two-wheeler, during which it displayed real-time results of helmet and license plate detection on the website.

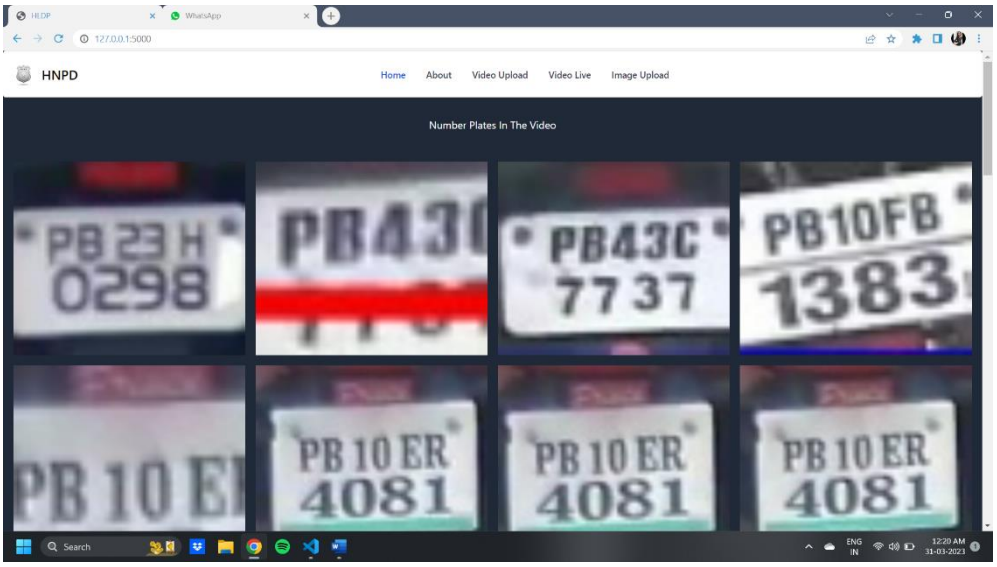

Fig 31: Output of video upload

that sounds impressive! It's great to hear that the YOLOv4 model is achieving such high accuracy in detecting helmets and license plates in various types of media, and that the web application is user-friendly and easy to use. The ability to support live streaming is particularly useful for real-time monitoring of traffic, which can help improve safety on the roads. Overall, it sounds like this application could be a valuable tool for traffic management

and safety.

Case 03: Live streaming

Live streaming refers to the act of broadcasting live video content over the internet in real-time. It has become increasingly popular in recent years as more people have access to high-speed internet and the technology required to broadcast live content has become more affordable and accessible.Live streaming can be used for a wide range of purposes, including entertainment, education, business, and social events. For example, musicians and performers can use live streaming to connect with fans and promote their work, while educators can use it to deliver online courses and webinars.

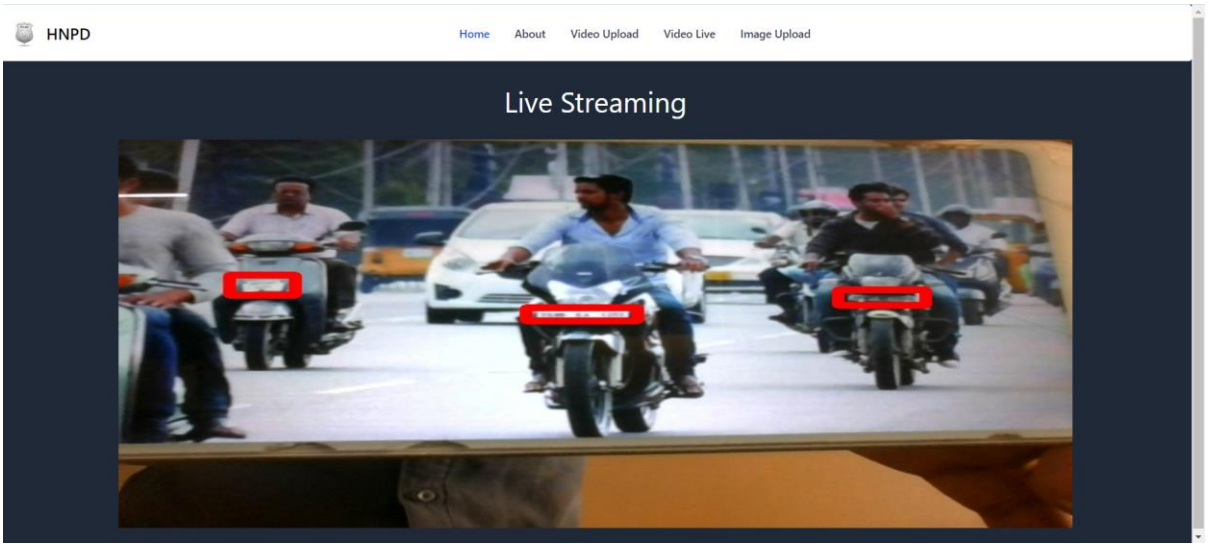

Fig 32: Output of Live streaming

There are a variety of platforms available for live streaming, including social media platforms like Facebook, YouTube, and Instagram, as well as dedicated streaming platforms like Twitch and Periscope. These platforms often provide tools and features to help users enhance their live streams, such as chat rooms, donation buttons, and analytics tracking.To start a live stream, you typically need a device with a camera and an internet connection. Some platforms also require special software or hardware to be installed, such as OBS Studio or a dedicated streaming encoder. Once you have everything set up, you can start broadcasting your live video to your audience in real-time.

Chapter 8

REFERENCES

8.REFERENCES

8.1 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.