



**Medi-Caps Institute of Technology and
Management**

HELMET DEFAULTER DETECTION

A Minor Project Report Submitted to

Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal

In partial fulfilment of the degree

Of

Bachelor of Engineering

In

Computer Science & Engineering

Submitted To:

Akanksha Gaur
Assistant Professor

Submitted By:

Parth Khopkar(0812CS151076)
Rishabh Jaiswal(0812CS151095)
Shruti Varadpande(0812CS151110)

**Department of Computer Science and Engineering
June, 2018**



**Medi-Caps Institute of Technology and
Management**

HELMET DEFAULTER DETECTION

A Minor Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

In partial fulfilment of the degree

Of

Bachelor of Engineering

In

Computer Science & Engineering

Submitted To:

Akanksha Gaur
Assistant Professor

Submitted By:

Parth Khopkar(0812CS151076)
Rishabh Jaiswal(0812CS151095)
Shruti Varadpande(0812CS151110)

**Department of Computer Science and Engineering
June, 2018**



Medi-Caps Institute of Technology and Management

A.B. Road, Pigdamber, Indore – 453 331 (M.P.)
Ph.: (0731) 4259700, 4259500, Fax : 4259501
Website: www.medicaps.ac.in
e-mail: info@medicaps-institute.ac.in

CERTIFICATE

This is to certify that **Parth Khopkar (0812CS151076), Rishabh Jaiswal (0812CS151095) and Shruti Varadpande (0812CS151110)** have completed their Minor project work titled “**Helmet Defaulter Detection**” as per the curriculum and submitted a satisfactory report on this project as a part of fulfilment towards the degree of **Bachelor of Engineering (Computer Science & Engineering)** from **Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal.**

Signature: _____

Name: _____

(Project Guide)

Signature: _____

Name: _____

(Internal Examiner)



Medi-Caps Institute of Technology and Management

A.B. Road, Pigdamber, Indore – 453 331 (M.P.)

Ph.: (0731) 4259700, 4259500, Fax : 4259501

Website: www.medicaps.ac.in

e-mail: info@medicaps-institute.ac.in

CERTIFICATE

This is to certify that **Parth Khopkar (0812CS151076), Rishabh Jaiswal (0812CS151095) and Shruti Varadpande (0812CS151110)** have completed their Minor project work titled “**Helmet Defaulter Detection**” as per the curriculum and submitted a satisfactory report on this project as a part of fulfilment towards the degree of **Bachelor of Engineering (Computer Science & Engineering)** from **Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal.**

Signature: _____

Name: _____

(External Examiner)

ACKNOWLEDGEMENT

The most awaited moment of successful completion of an endeavour is always a result of persons involved implicitly or explicitly in it. The successful completion of all the phases of our project is the result of dedicated efforts put in by many people and this report would be incomplete without giving due credits to them. This acknowledgment is but a small token of gratitude in recognition of their help in our endeavours.

We sincerely thank our project guide **Ms. Akanksha Gaur** for providing us her precious time, considering our doubts, mostly arose at odd times and providing us the solutions that always takes us out from all the chaos. It has been an honour and pleasure to work under her. Not just the technical knowledge but a lot we have learnt from her calm and composed attitude for which we will remain indebted to mam throughout our life.

We express a deep sense of gratitude to our project coordinator **Ms. Akanksha Gaur** for her invaluable support and constant encouragement which was of great support to bring this work to present shape.

We would like to give our sincere thanks to **Mr. Suresh Jain, HOD of Computer Science Department** from whom we have learned a lot.

Last but not the least; we would like to thanks our colleagues, friends and our parents who were a constant and willing source of encouragement and inspiration for us throughout the project.

Parth Khopkar

Rishabh Jaiswal

Shruti Varadpande

CONTENT

1. Introduction	1
1.1 Objective.....	2
1.2 Scope.....	2
1.3 Problems Domain.....	3
1.4 Solution Domain.....	3
1.5 Platform Specification.....	3
1.5.1 Hardware.....	3
1.5.2 Software	5
2. System Requirement Analysis.....	6
2.1 Information Gathering.....	7
2.1.1 Functional Requirement.....	7
2.1.2 Non-Functional Requirement.....	8
2.2 System Feasibility.....	10
2.2.1 Operational.....	10
2.2.2 Technical.....	10
2.2.3 Economical.....	10
3. System Design.....	11
3.1 Use Case Diagram.....	12
3.2 Class Diagram.....	13
3.3 Sequence Diagram.....	14
3.4 Collaboration Diagram.....	16
3.5 Activity Diagram.....	17
3.6 Data Flow Diagram.....	18
4. Design.....	20
4.1 Data Design.....	21
4.1.1-4 ER Diagrams & Data Dictionary.....	21
4.2 System Architecture.....	23
4.2.1 Structure Chart.....	24
5. Implementation.....	25
5.1 Implementation Of Modules.....	26
5.1 Results.....	29
6. Testing.....	30
6.1 Testing Objective.....	31
6.2 Testing Scope.....	31
6.3 Testing Principles.....	31
6.4 Testing Methods Used.....	31
6.5 Test Cases.....	33
6.6 Sample Test Data and Results.....	34
7. Limitation.....	36
8. Future scope.....	36

9. Conclusion.....	36
10. Bibliography and References.....	37

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1	Use Case Diagram	12
2	Class Diagram	13
3	Sequence Diagram	14
4	Collaboration Diagram	16
5	Activity Diagram	17
6	Data Flow Diagram (LEVEL-0)	18
7	Data Flow Diagram (LEVEL-1)	19
8	E-R Diagram for train.csv	21
9	E-R Diagram for Detected Person Frame	22
10	Structure Chart	24
11	LabelImg GUI	26
12	Tensorboard Visualization	27
13	GUI for Person Detection Module	28
14	White Box Testing through Jupyter Notebook	33
15	Sample Video Stream	34
16	Sample Video Stream Detection	34
17	Detected Person	35
18 (a)	Detected Helmet	35
18 (b)	No Helmet Detected	35

LIST OF TABLES

Table No.	Table Name	Page No.
1	Data Dictionary for train.csv	22
2	Data Dictionary for extracted frame (person detected)	23

ABSTRACT

The traffic of the cities is increasing day by day and the number of accidents that happen have also increased to a great limit. Indore has recorded maximum number of road accident deaths every year since 2011. And most of these accidents have a two-wheeler vehicle. People suffer from fatal injuries during these accidents because of not wearing the helmet. And this therefore leads to a lot of deaths and severe head damages. Indore Police is trying to have a compulsory helmet rule compliance but it is not working that efficiently and a lot of manual labour and efforts are required.

Therefore, this project aims to help ease the process of detecting defaulters of the compulsory helmet rule for the Traffic Police. Almost all major cities have traffic surveillance cameras installed at the traffic signals, intersections and other places of interest. Currently these systems are only used to record the footage which is later viewed by the officials to determine defaulters. This results in waste of man hours. Our project aims to delegate this job to computer systems by automatically finding the defaulters by using Machine Learning. This can be increased to the scope of detecting the license plate number if a person is found guilty by not wearing the helmet. This way, this introduction of technology would help both the traffic police as well as will ensure the safety of the citizens.

INTRODUCTION

1. INTRODUCTION

Indore is a thriving city in the middle of the country and has a population exceeding 2.5 million. Known as Chota Mumbai, the city is one of the highest rated start-up destinations in India. Being the most populous city in Madhya Pradesh, a state with high population density, Indore has fared well on developmental indicators. As the case happens to be, the swift progress comes with a cost of rapid urbanization problems. One such case is that of the problems faced by the city in its management of traffic, which is humongous given the vehicle count of the city exceeding 2 million and another 1.5 lakh vehicles entering the city on a daily basis for regular chores. With the area of the city spread across 390 km², it results in sustained traffic problems. The problem is bound to rise with the density pressure on the city exceeding day by day.

One of the identified issues present in the field, one of the major concerns is that people do not generally wear helmet. Helmets are often found in the hands of pillion riders and they are put on only after spotting traffic police personnel. Even though we have very high number of road accidents and fatalities relating to it, every Indian on the road considers himself special and an expert level driver and believes that he/she does not need a helmet. Another misconception is that, helmet is not required in low speeds. Even after 1500-2000 challans being issued in a day at times, the situation doesn't seem to be changing for good. On the contrary, the policemen have to bear the brunt of enraged people not willing to pay the fines. To reduce the fatalities caused by accidents, it is necessary to have a strict check on the regular usage of helmets as well as quality of helmets begin used by the riders.

1.1 Objective

The problem of traffic safety for can be solved efficiently if we make our detection systems automated. This will enable us to detect defaulters more easily and efficiently. This project aims to solve the problems using existing resources available within the traffic infrastructure. Thus, we propose to use traffic camera surveillance to detect helmet defaulters. We intend to use Machine Learning to train computer systems to detect the defaulters and automatically generate a challan.

1.2 Scope

The scope of this project is widespread and focuses primarily on the problems faced by the traffic police department. It involves the solution of the traffic police problem which involves the detection of the

presence or absence of helmet on the head of driver on a 2-wheeler. The process of this detection also leads in generating the automatic e-challan which helps the police in tracking and maintaining the record of defaulters easily. This definitely benefits the police as well as the citizens of the city. It acknowledges the safety of the citizens and eases out the tiresome manual job of the police department.

1.3 Problem Domain

The problem domain of this project consists of the following points –

- The existing system is not robust enough to detect the people not wearing helmet. It is either done manually by the traffic policemen or the technology used is not up to the mark.
- More number of people prefer not to wear a helmet as they feel wearing it is tiresome.
- This hinders the safety of the people riding 2-wheelers.
- Not wearing helmets can lead to severe head injuries in the road accidents.

1.4 Solution Domain

The solution domain of this project is multifold and solves many problems through one concept or software. We have primarily used Machine Learning to solve this problem. It solves the problem of imposing the compulsory helmet wearing rule for 2-wheeler drivers. This ensures the safety of the people and also makes the process easy for traffic policemen. The whole system of helmet wearing compliance gets easy and the scope of generation of e-challan also becomes possible.

1.5 Platform Specification

Hardware

The specifications of the hardware used are –

1. GPU (Graphical Processing Unit) - A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing, and their highly parallel structure makes them more efficient than general-purpose CPUs for algorithms where the processing of large blocks of data is done in parallel.

The GPU that we have used in this project is the Nvidia GT750 with 2GB RAM. It helps in boosting the parallel image processing. For better performance and faster training time GPUs with

upto 12 GB of RAM can be utilized.

2. RAM - Random-access memory is a form of computer data storage that stores data and code currently being used. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. The machine we used for the training process has 8 GB RAM.
3. Surveillance Camera – The surveillance camera or a web cam is needed to capture the input footage to perform the testing and training on. It can be a camera with any normal specification which captures a low-quality image or video or can depend upon the budget and feasibility of the project.

Software

TensorFlow is majorly used in this project. It has a lot of dependencies involved in it.

TensorFlow - TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

The project depends on the following libraries and software:

- Protobuf 2.6 - The Tensorflow Object Detection API uses Protocol Buffers to configure model and training parameters. Before the framework can be used, the Protobuf libraries must be compiled.
- Python-tk - The Tkinter module (“Tk interface”) is the standard Python interface to the Tk GUI toolkit.
- Pillow 1.0 - Pillow is a “friendly” fork of the Python Imaging Library. The goal is to see if any improvement can be made to the packaging situation (*) by opening up development to the “public”.
- Lxml - lxml is a Pythonic, mature binding for the libxml2 and libxslt libraries. It provides safe and convenient access to these libraries using the ElementTree API.
- tfSlim - TF-Slim is a lightweight library for defining, training and evaluating complex models in TensorFlow. Components of tf-slim can be freely mixed with native tensorflow, as well as other frameworks, such as tf.contrib.learn.
- Jupyter Notebook - Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.
- Matplotlib - matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

- **COCO API** - COCO is a large image dataset designed for object detection, segmentation, person keypoints detection, stuff segmentation, and caption generation. This package provides Matlab, Python, and Lua APIs that assists in loading, parsing, and visualizing the annotations in COCO.

Apart from this, the Normal Operation System (Windows or Linux is required).

SYSTEM REQUIREMENT ANALYSIS

2. SYSTEM REQUIREMENT ANALYSIS

2.1 Information Gathering

2.1.1 Functional Requirement

Functional requirement of this project has various aspects. It includes the following specifications –

1. Data – The data that is entered in the system is in the form of images and video. It depends if the data is historical or live data is streamed into the system. The data is mostly unstructured if the images are stored or a stored video is processed. A live stream video is directly put as an input to the software and the helmet is detected.
2. Operations – The operations that are performed in this project are very synchronized and systematic. There are not many major operations that are performed by the software. The detection of helmet takes place in majorly only two steps. First step is when the system is trained to detect the helmet on a person's head. For that, the system is trained with around 50-60 images after which the next step i.e. testing is performed. The testing is generally performed on the lesser number of images. Therefore, we take around 5-10 images and see if the system detects the helmet correctly.
3. Work Flow – The work flow of the system is very simple and understandable. It involves of the steps in which first we enter the stored images or the live stream data into the system or the software. After that, the software detects the people on 2 wheelers in those images and then checks whether the rider is wearing a helmet or not and accordingly takes a snapshot. That snapshot is stored in the result data set which is later used for reference. The concerned authorities then validate the result data set to check if the detection is done accurately or not. Therefore, the last step is the validation of the result data.
4. Output – The output of the system consists of the result data set which describes whether the person has worn helmet or not. It is shown by the boxes that are made around the helmet if it is present in the image or the video. This can also lead to the license plate detection of the vehicle in the similar manner and the information of the driver can be derived from the same process.
5. Entry of data – The data can be entered by the concerned authority who is in charge of the helmet detection scheme. Or by a contractor who is given the job of this by the government.
6. Regulatory Requirements – By the build of the boxes in the data stream around the helmets. It becomes easy to meet regulatory requirements of this project and then the necessary actions can take place.

7. Administrative Functions - The rights of the administrative functions have with the authority who is going to regulate this. It has the right to control the capture the footage.
8. Authentication – The authentication of the system is checked by the developer while performing the testing on the data set. The authenticity the input image or video is checked by the live cameras that get it.
9. External Interface – The external interface of this project consists of only one interface which is used to show the input video or image on a GUI.
10. Certification Requirements – The only certification that is required is by the traffic police department.
11. Legal or Regulatory Requirements – The legal permission to capture the traffic footage in order to detect the presence of helmet is required. And the regulation of the traffic police department is required to access the footage and pass it through the system.

2.1.2 Non-functional Requirements

Non-functional requirements of the system consist of –

1. Performance – The performance of the system is quite efficient and it yields 99% accuracy.
2. Scalability – The scalability of this project extends till a limit the hardware can handle the input data. As the traffic increases, the input of the data will increase as well. Therefore, the system specification needs to be increased in such cases by equipping it with better hardware.
3. Availability – Availability of such software exists, but it depends on how efficient the throughput is and how fast the results are fetched.
4. Reliability – The software is quite reliable and fetches the correct results. The only point of failure can occur if there comes a similar object which look like helmet and then the system might detect it as helmet.
5. Maintainability – It does not require maintenance as such until the hardware resources get exhausted.
6. Security – TensorFlow is an open source module and hence can cause bugs and errors. However, good development practices can assist with defending against attempts from users looking to expose data or bring down an app.
7. Environmental – The system does not cause any major harm to the environment.
8. Data Integrity – The data remains consistent throughout the training and testing of the system. Only the type and size of the images and video changes. Other than that, the authenticity of the data is preserved.

9. Usability – The usability of the system is high in the traffic police department. The system is primarily made for the ease of the manual work of the traffic policemen in the helmet detection.

2.2 System Feasibility

2.2.1 Operational

Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. This project solves the problem of the helmet detection problem of the traffic police department as the work till now is happening more or less manually and a complete software based automatic manual has not been proposed till yet. The already existing cameras at the traffic signals and at various place can be taken advantage of to collect the live footage of the traffic and can be fed as input data to the system and the function of the helmet detection can be performed by the system. This saves a lot of work of the traffic police men and leads to the 99% accuracy.

2.2.2 Technical

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. In this project, we have used Machine Learning as our main tool to solve the helmet detection problem. Machine Learning is itself very stable and has a lot many options to solve a particular problem. We have used the TensorFlow software in the Python language to train and test the images. TensorFlow is a very well-established software with wide variety of modules. They can be used to solve the various aspects of this problem. The modules of the TensorFlow can also be upgraded to a certain level according to the requirement of the user. Therefore, it solves a major part of the problem.

2.2.3 Economical

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software.

In this project, the major cost that is incurred is in the hardware parts required to build the system. As the system is upgraded and the input of the data is scaled up, then a much higher GPU and RAM would

be required. That cost would add up. Also, if the traffic increases, the input data will increase up too leading to a million images and videos every day. That will require huge data centers or the cloud space to store the historical as well as the processed results. This will also add up to the cost of the system and need to be purchases once the system comes into working. This cost can be maintained by the fine collected by helmet defaulters. That can pay both the developers as well as maintain the cost of the software development team. And the other financial needs can be funded by the government as they have separate section in treasury for the traffic development.

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Use Case diagram

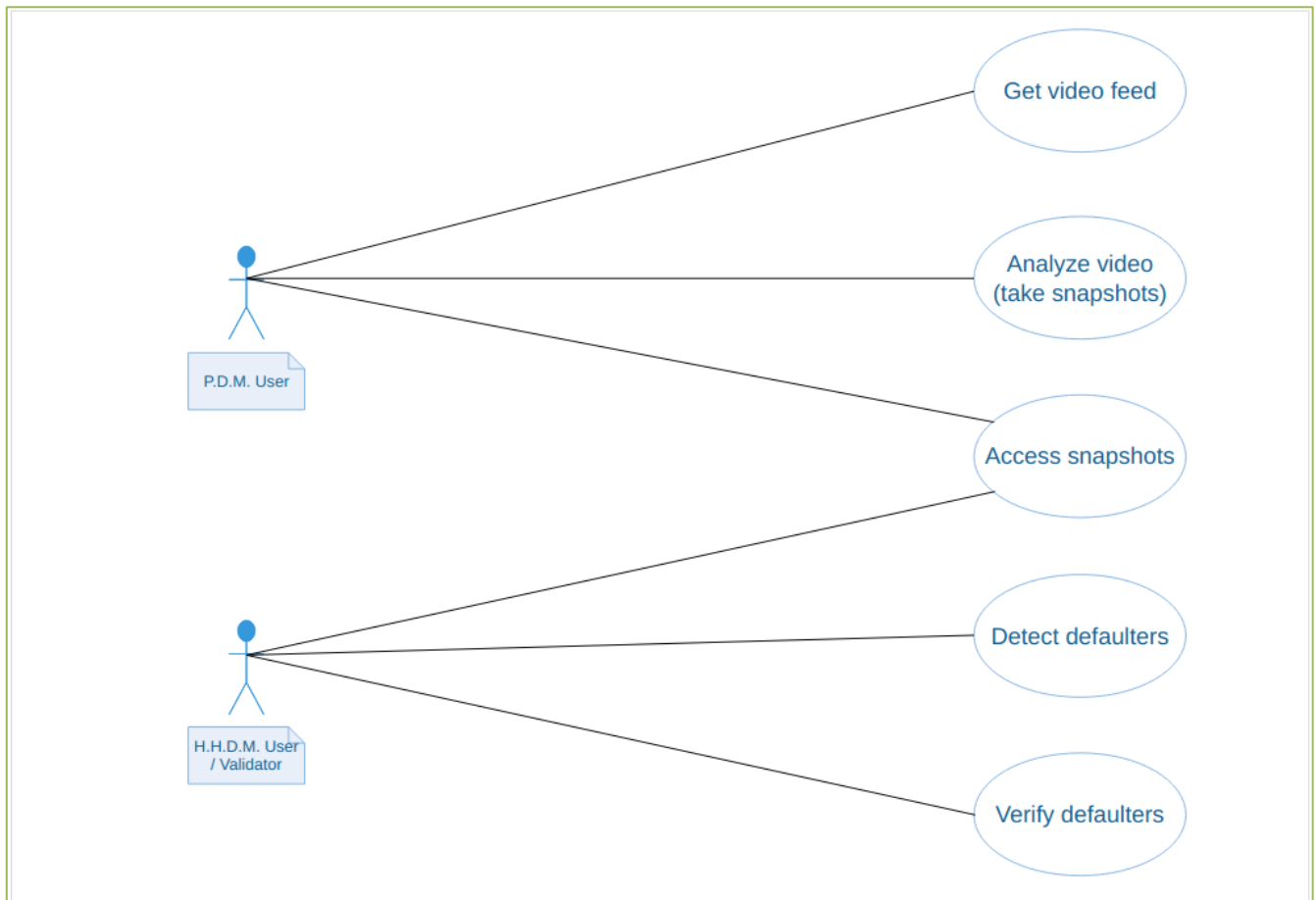


Fig. 1 - Use-Case Diagram

Description –

1. There are two types of user (actors) for the given system – Person Detection Module (P.D.M.) User and Head-Helmet Detection Module (H.H.D.M.) User.
2. Both the users have access to snapshots generated by P.D.M.
3. Role of P.D.M. User is to:-
 - a. Feed video data to P.D.M.,
 - b. Analyse the processed video, and
 - c. Take snapshots and remove redundancy of data (snapshots).
4. Role of H.H.D.M. User is to:-
 - a. Feed snapshots to H.H.D.M.,
 - b. Use the module to detect defaulters, and
 - c. Validate defaulters indicated by the module.

3.2 Class diagram

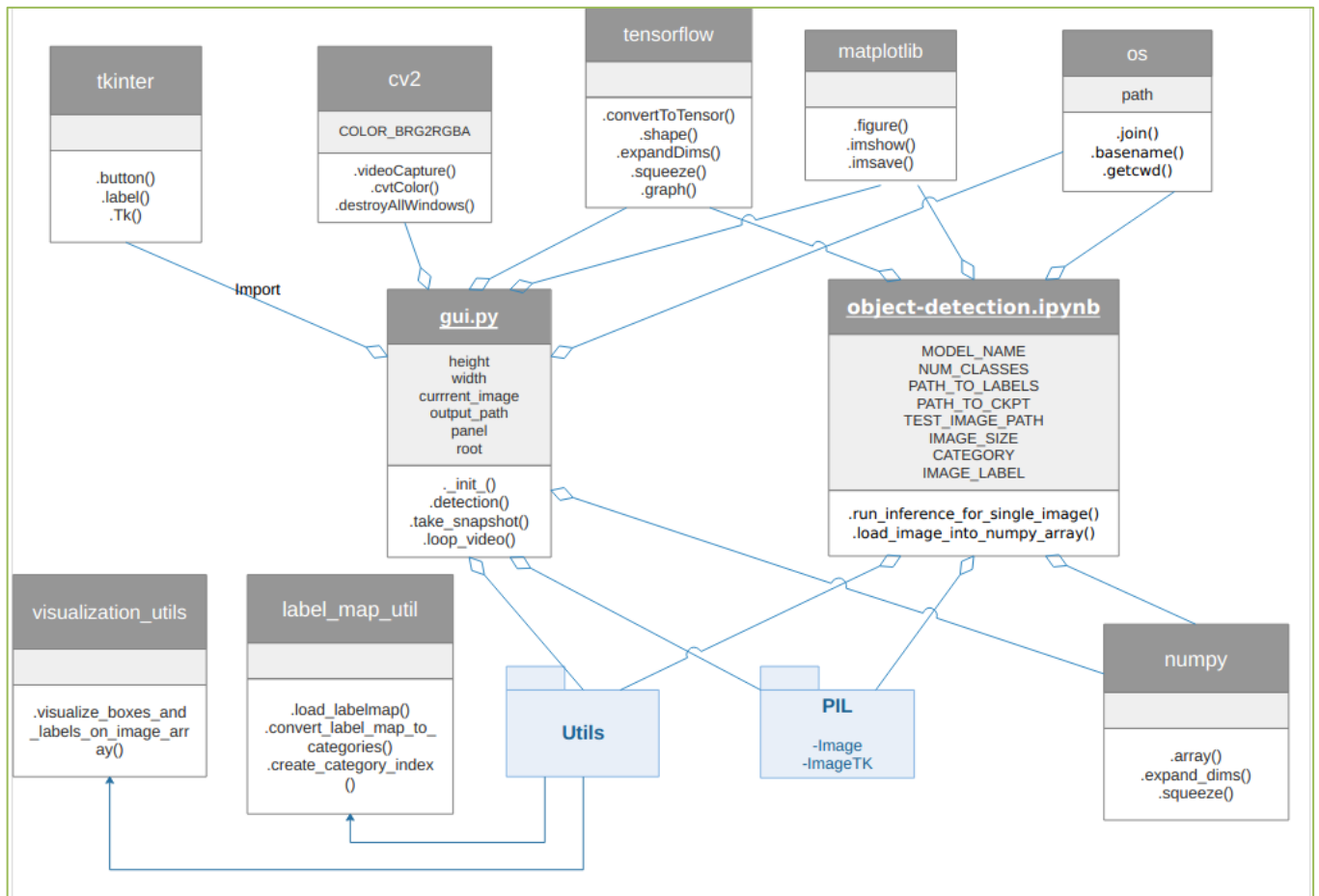


Fig.2 - Class Diagram

Description –

1. The system consists of 10 significant classes, 2 of which act as main modules. There are many more dependencies other than these classes which are somehow involved in the system working.
2. There are 2 packages namely Utils and PIL.
3. Utils consists of 2 significant classes – visualization_utils and label_map_util.
4. PIL consists of 2 significant classes – Image and ImageTK.
5. The two main modules are:-
 - a. gui.py – It is the Person Detection Module. It is aggregated by (i.e. it imports) following classes/packages:
 - i. Utils
 - ii. tkinter
 - iii. cv2 (OpenCV)
 - iv. tensorflow
 - v. os

- vi. numpy
- vii. matplotlib.pyplot
- viii. PIL

b. object_detection.ipynb – It is the Head-Helmet Detection Module. It is aggregated by (i.e. it imports) following classes/packages:

- i. tensorflow
- ii. utils
- iii. os
- iv. numpy
- v. matplotlib.pyplot
- vi. PIL

3.3 Sequence diagram

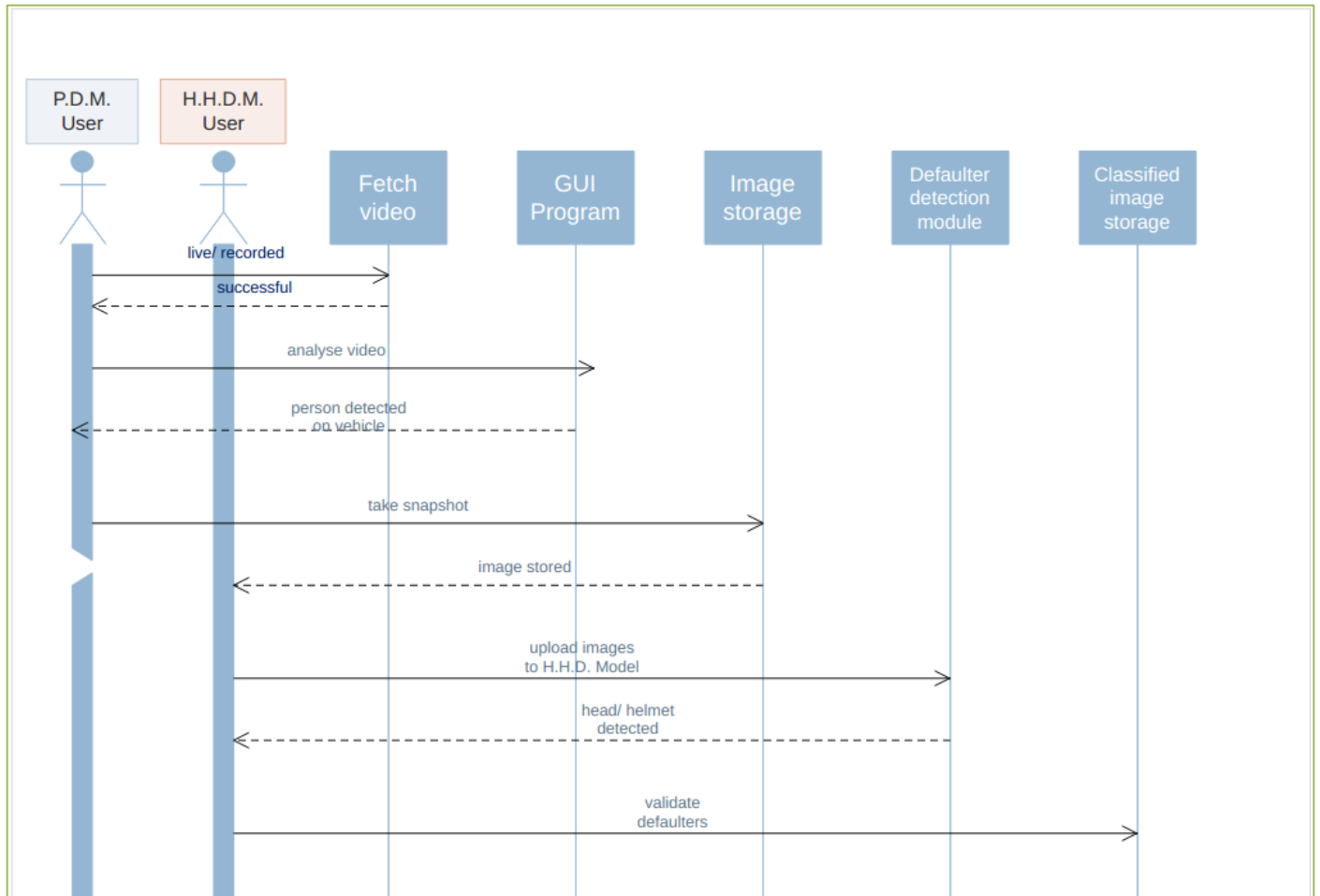


Fig. 3 - Sequence Diagram

Description –

1. The system working can be classified into five activities.
2. The role of P.D.M. User ends after the third activity after which role of H.H.D.M. User also starts.
3. The sequence of activities are enumerated as:-
 - a. Live/recorded video is fed to system by P.D.M. User.
 - b. After successful fetching of raw video data, the GUI program runs and processes video for analysis i.e. to detect a person on vehicle.
 - c. When detected, a cropped snapshot is produced as an output and stored.
 - d. The stored image data is fed to H.H.D. Module to detect defaulters.
 - e. The detection output is then validated by the H.H.D.M. User.

3.4 Collaboration diagram

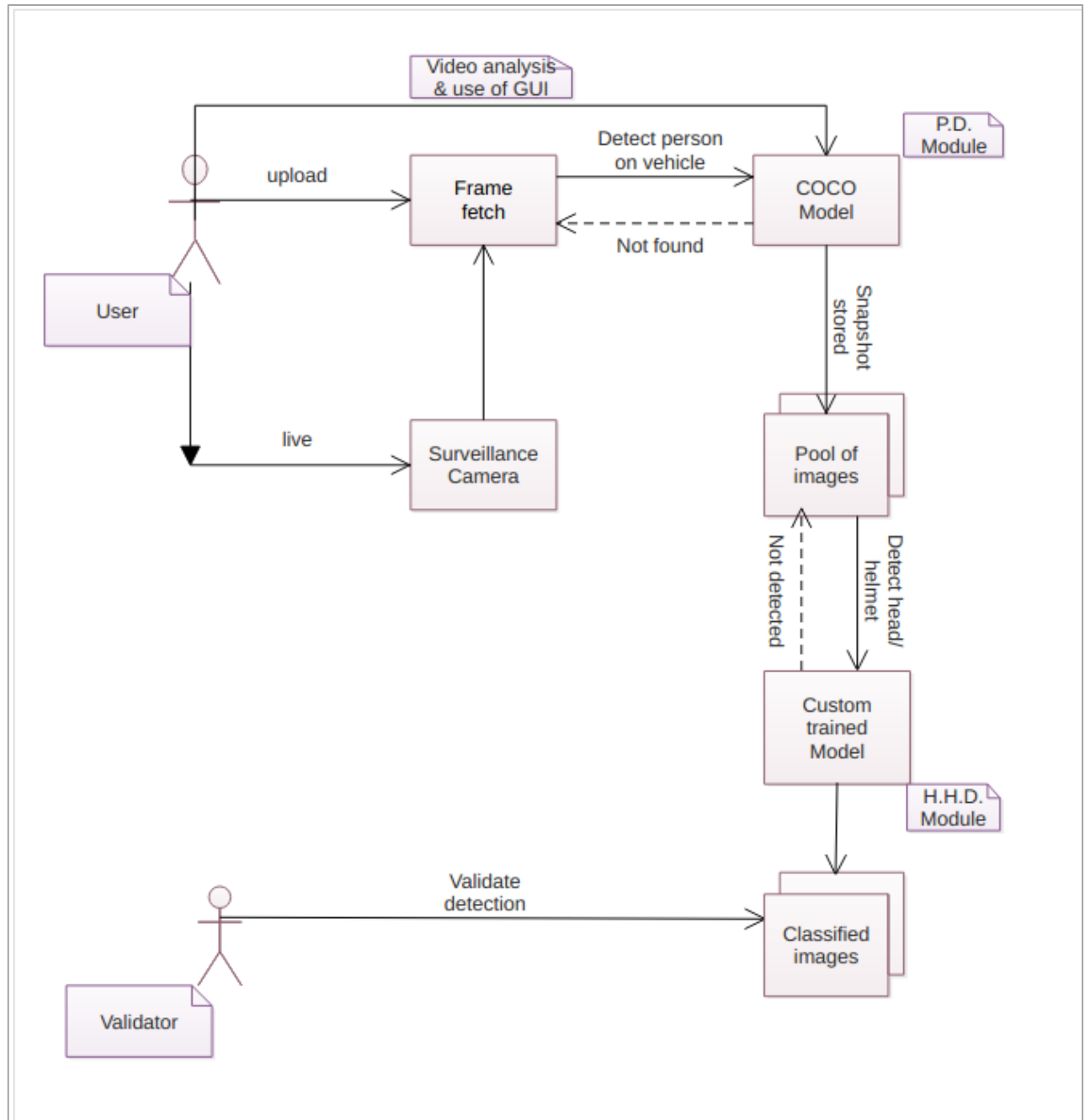


Fig. 4 - Collaboration Diagram

Description –

Given diagram represents the collaboration of various actors, modules, hardware and activities involved in the working of our Defaulter Detection System.

1. Actors are:
 - a. User

- b. Validator
- 2. Modules are:
 - a. COCO Model
 - b. Custom-trained Model
- 3. Hardware are:
 - a. Surveillance camera
 - b. Computer system
- 4. Data files are:
 - a. Pool of images
 - b. Classified images

3.5 Activity diagram

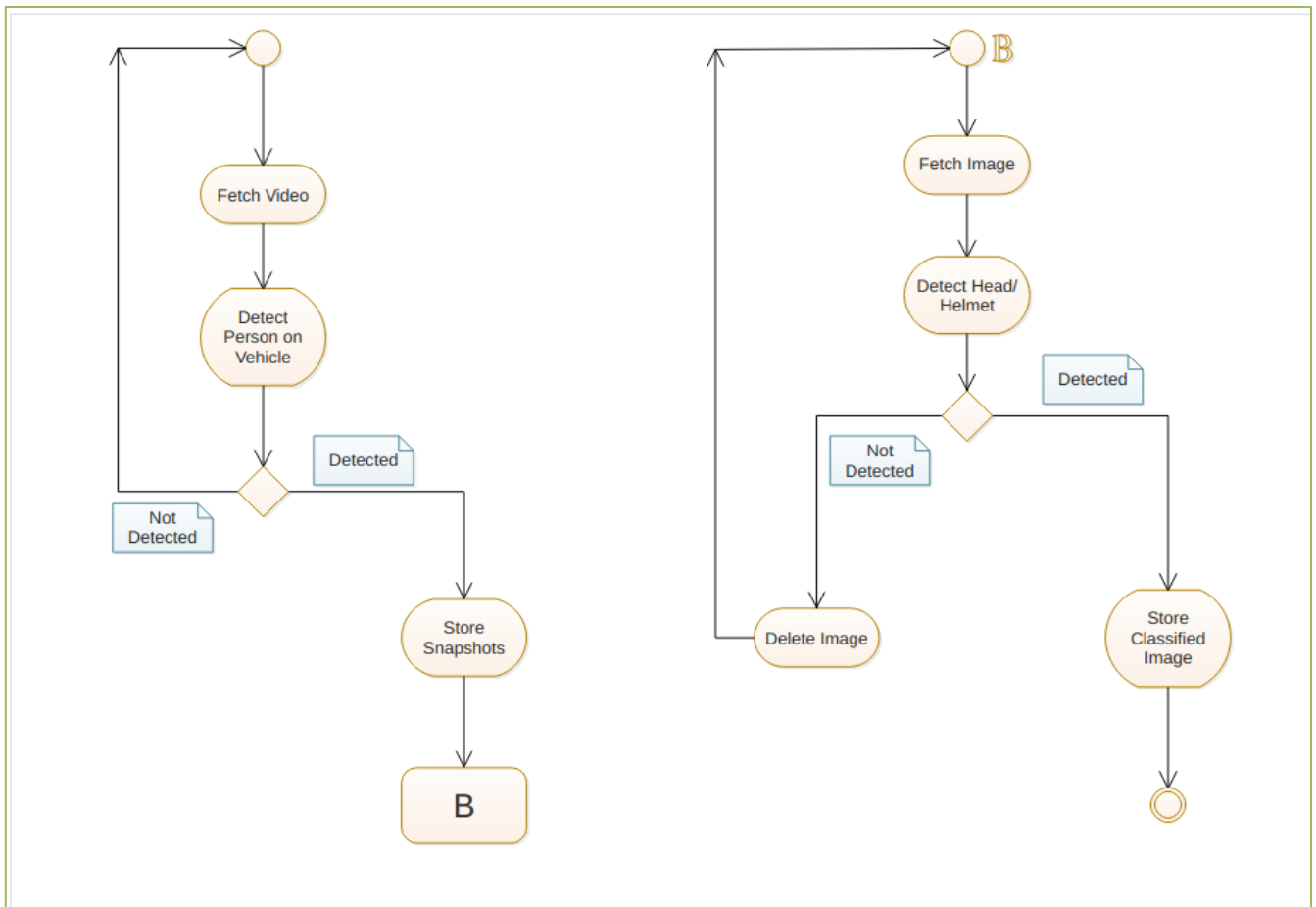


Fig.5 - Activity Diagram

Description –

The activity diagram given above is segregated in two parts according to the modules they come in. Initial activity starts at P.D. Module whereas as B denotes the initialization of H.H.D. Module.

3.6 Data Flow diagram

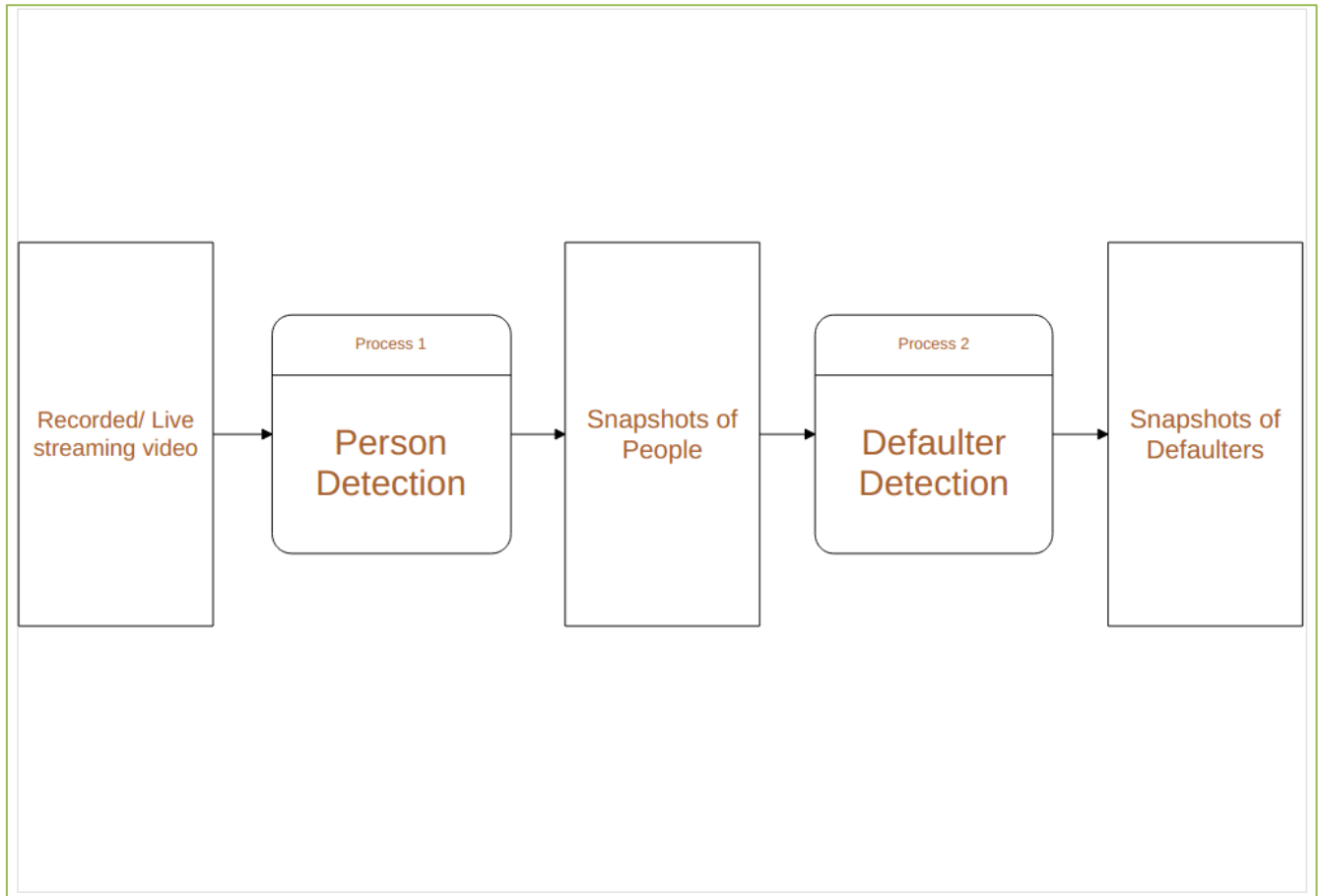


Fig.6 - Data Flow Diagram (LEVEL-0)

Description –

The above diagram shows basic processes involved in the Defaulter Detection System from input to output.

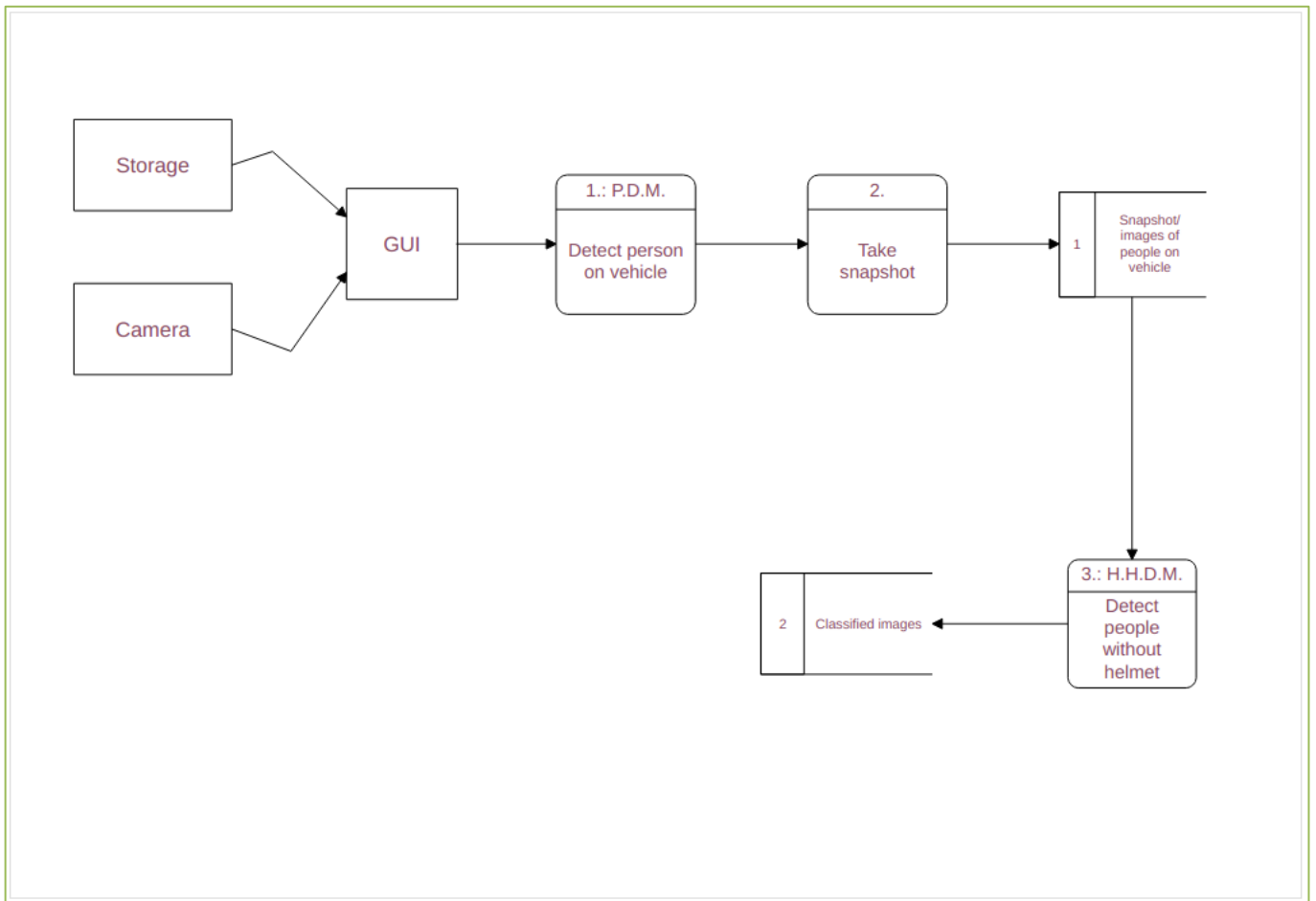


Fig.7 - Data Flow Diagram (LEVEL-1)

Description –

The above diagram shows elaborated view of processes, interfaces and data files involved in the Defaulter Detection System from input to output.

DESIGN

4. DESIGN

4.1 Data Design

Since the software is mostly based on Machine Learning through Neural Networks, it does not involve any database directly. However, some functionality involved here uses structured data to process images. The structure used here is called **Tensor**. Tensors are special kind of vectors used to handle image data. TensorFlow (here) uses tensors to process images and give comprehensive outputs such as detecting images and highlighting them with their names using bounding boxes.

We also use data in tabular format (**CSV** to be precise) to train the object-detection model which after training is implemented in the system to identify specific objects. We call this data as **Training data/dataset**.

4.1.1 E-R Diagram - Training Data Design

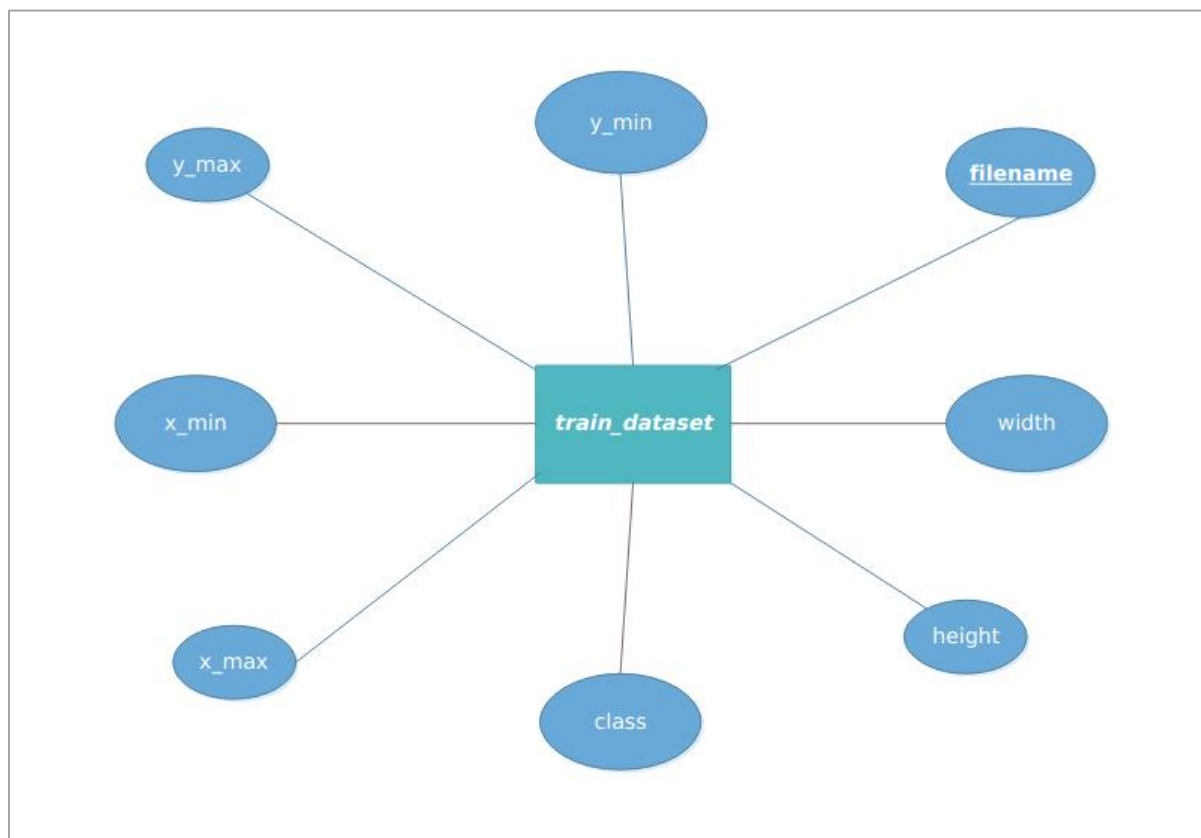


Fig. 8 - E-R Diagram for train.csv (Training Dataset)

4.1.2 Data Dictionary

Field	Type	Constraints
Filename (Primary Key)	String	File naming constraints
Width	Int	Width = 500
Height	Int	Height = 888
Class	String	Either 'head' or 'helmet'
Xmax	Int	\leq width
Xmin	Int	≥ 0
Ymax	Int	\leq height
Ymin	Int	≥ 0

Table 1: Data Dictionary for train.csv (Training Dataset)

The above shown E-R diagram corresponds to the data set of Custom-trained model (Mobile-Net).

4.1.3 E-R Diagram – Person on Two Wheeler Identification

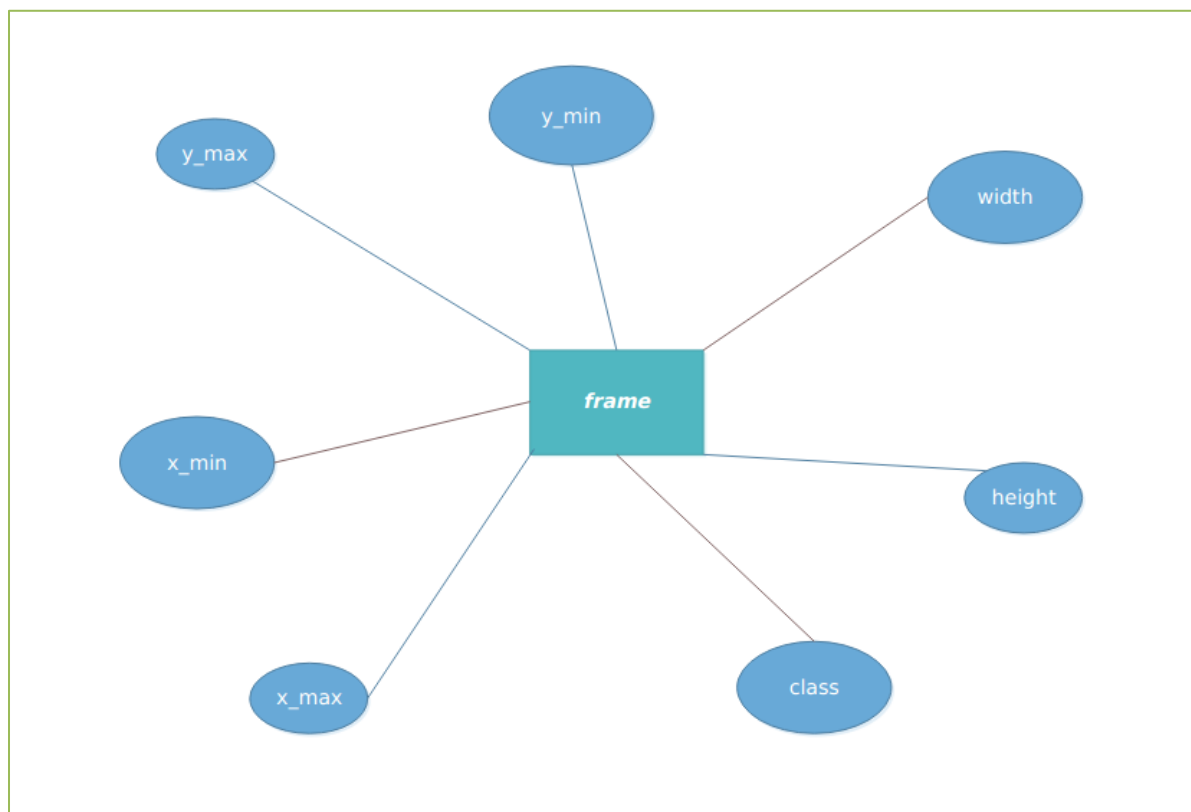


Fig. 9 - E-R Diagram for Extracted Frame of Detected Person

4.1.4 Data Dictionary

Field	Type	Constraints
Width	Int	Constant 'w'
Height	Int	Constant 'h'
Xmax	Int	$\leq w$
Xmin	Int	≥ 0
Ymax	Int	$\leq h$
Ymin	Int	≥ 0
Class	String	Any value from Labels list

Table 2: Data Dictionary for Extracted Frame of Detected Person

4.2 System Architecture

The Traffic Defaulter Detection System consists of two types of Object Detection Models – COCO trained Mobile_Net Model and Custom trained Mobile_Net Model. These models form the basis of our two main modules – P.D.M. and H.H.D.M. – respectively.

The users of these modules play an important role in system architecture by directly involving the execution of system. Structure chart for our system is shown on the next page.

4.2.1 Structure Chart

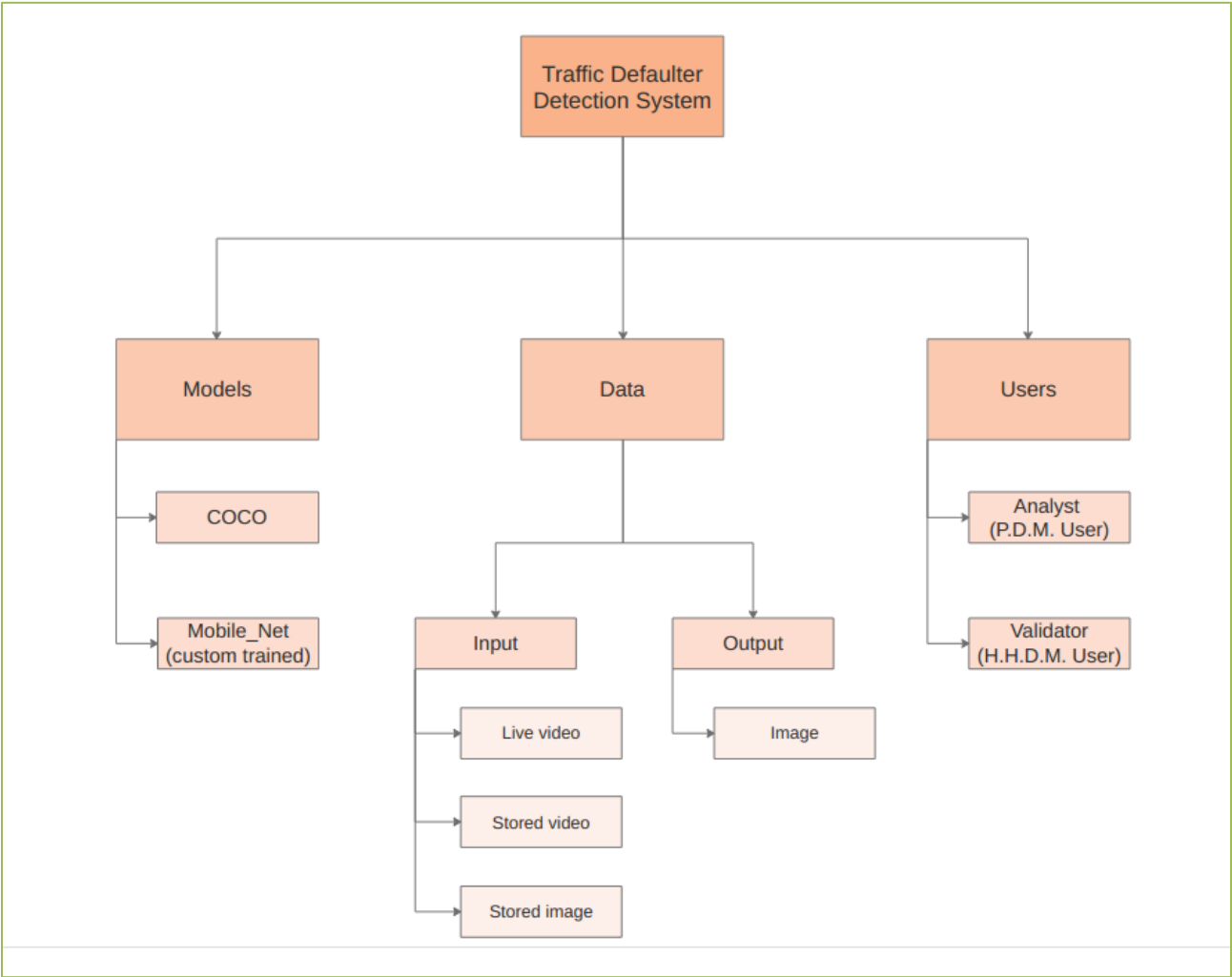


Fig. 10 - Structure Chart

IMPLEMENTATION

5. IMPLEMENTATION

5.1 Implementation of Modules

Programming modules –

1. XML to CSV Creator - This module will convert XML files generated by Label Img into CSV records.
2. TFRecord Generator – This module will convert the CSV files to TensorFlow records which will be used for training our model.
3. LabelImg – LabelImg is a graphical image annotation tool and label object bounding boxes in images.

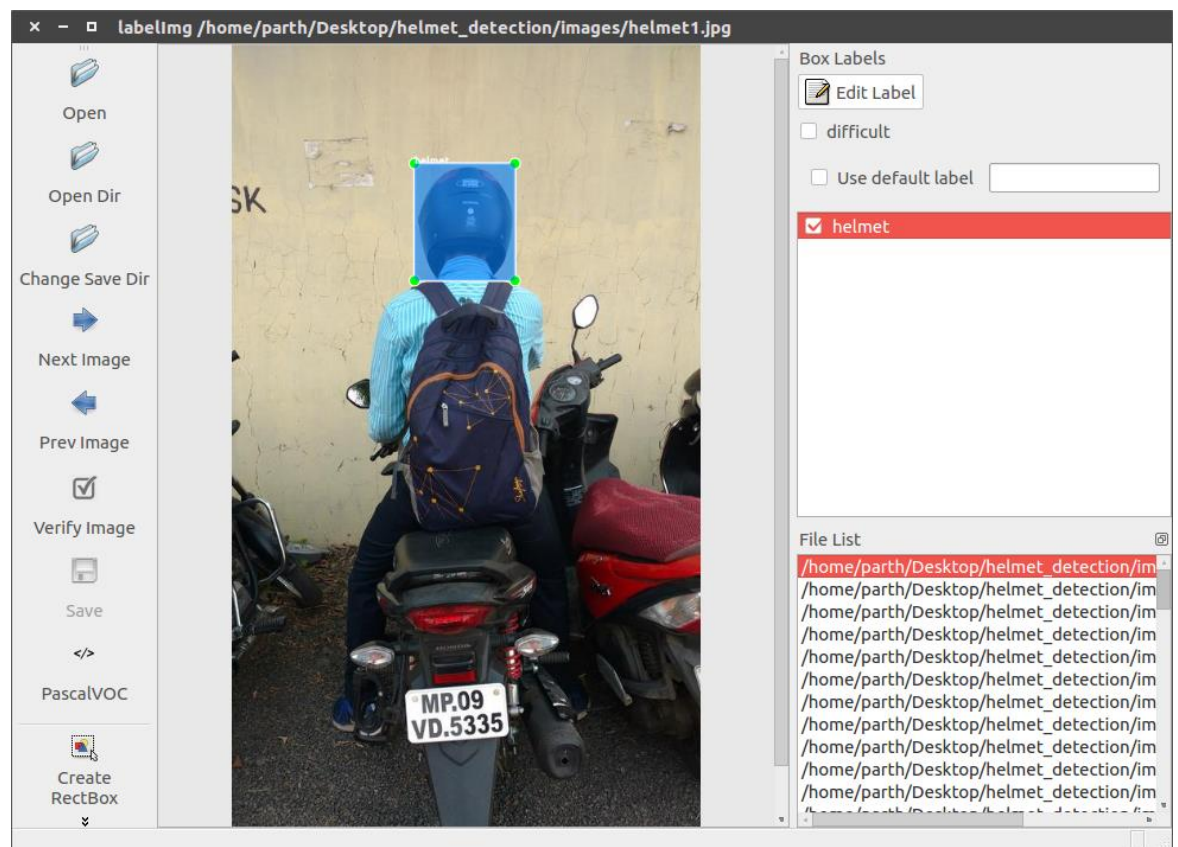


Fig. 11 – LabelImg GUI

4. Training Module – The training module takes the TF Records in “.record” format and a pre-trained model which is then passed to the Tensorflow Object Detection API. For this project we have used ssd_mobilenet_v1_coco pre-trained model along with its configuration file. The model is publicly available on Tensorflow GitHub repository as well as the Tensorflow website. We have taken advantage of transfer learning through this pre-trained model which made the training process much quicker and also allowed for less data.

5. Tensorboard – The Tensorboard module is an event viewer which enables us to monitor the training process.

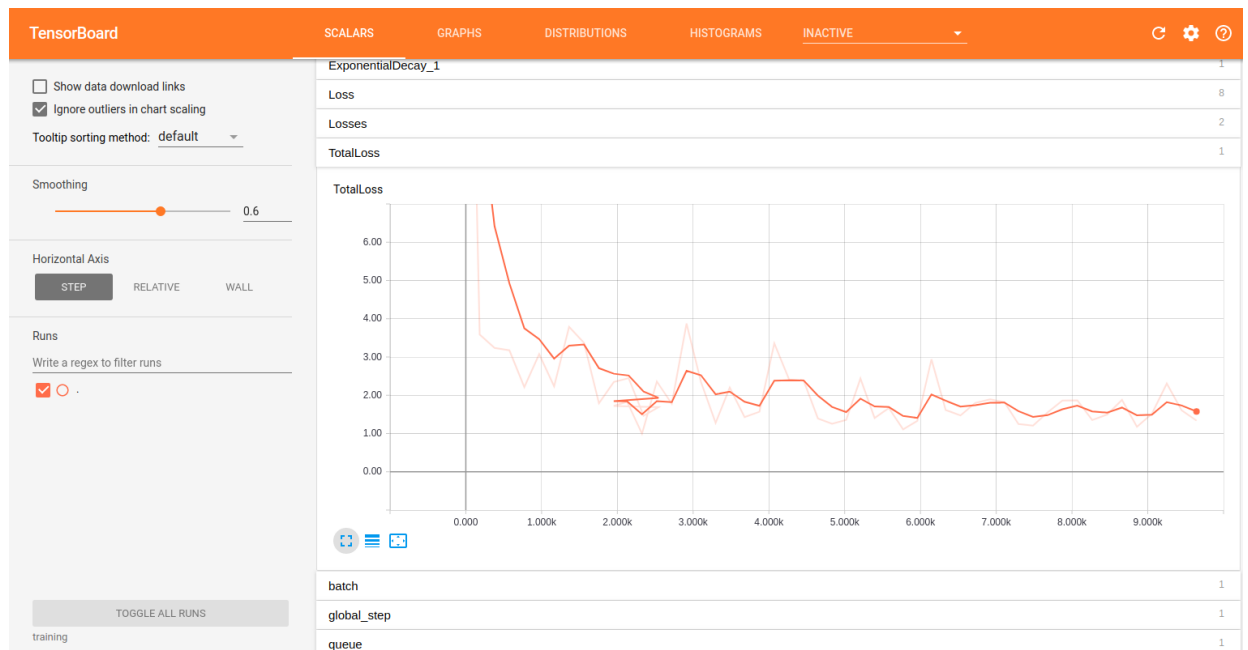


Fig. 12 – Tensorboard Visualisation

6. Detection Module – This is the module which will receive the feed from the surveillance camera and use the trained module to detect whether or not a rider is wearing a helmet. If not, the license plate number will be detected and a fine sent to the owner of the vehicle.
7. Data enhancement – The images are resized and renamed through this module which helps in cleaning the data in a better way and then the data set is fed to the training and testing module.

8. GUI Module – Tkinter is used to make the GUI for this project. It is an interactive toolkit for the GUI in Python.

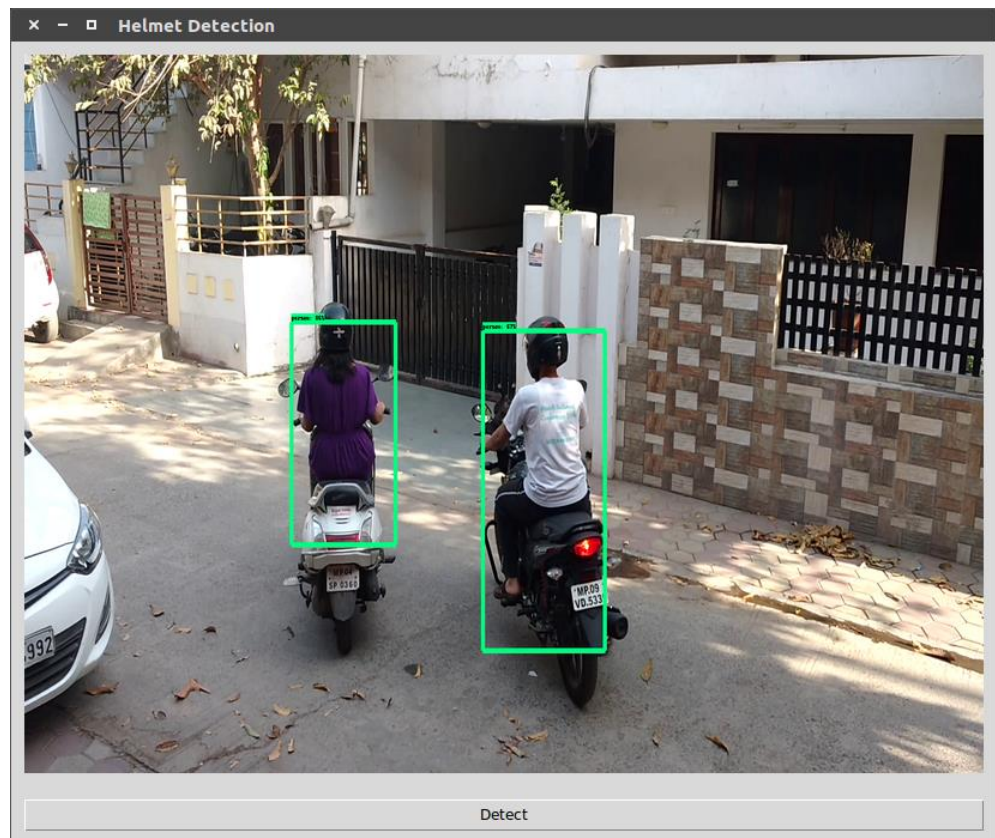


Fig. 13 – GUI for Person Detection Module

9. Open CV Module – OpenCV is a library of programming functions mainly aimed at real-time computer vision.

Networking module –

Surveillance Camera Network – The surveillance camera will provide a live feed to the detection module.

Database module

Training Images – A collection of around 100+ images which will be used for training the model to recognize riders wearing helmets.

5.2 Result

All the modules of the project are compiled together to give the final output. Although, each module has its individual results which are then fed into the next module.

1. The Networking Module first comes into play in which the camera is used to capture the images and video footage. The result of this is directly transferred to the database module where the further processing of data takes place.
2. The Database Module consists of the 100s of images and video footages that have been captured by the camera. They are stored in a folder out of which some images are used for training and some are used for testing. The images are also resized and renamed with the help of programming module which are also then saved in the same folder.
3. The Programming Module is the main working entity of the project. It has several inbuilt modules which help in carrying out the process in a systematic manner. The images are first converted from XML to CSV format by XML to CSV Creator. Then those CSV images are turned into TF format by TFRecord Generator. These images now become ready for training and testing. The images are resized and renamed in order to build a cleaner dataset. Once the images are in the TF format, LabelImg is used to draw the bounding boxes around the helmet which recognizes or shows the presence of helmet. Then the images are passed through the training module. The progress of the process can be seen through the Tensor Board. All these modules are then finally integrated through a GUI for which we have used Tkinter. If the data set on which the process to be performed is video or live camera, then we use the OpenCV module.

Combining all the modules mentioned above, the results can be generated according to the requirement. The results depend upon the type of input we give.

TESTING

6. TESTING

6.1 Testing Objective

There are various objectives for testing of this particular project. The major objectives are as follows

–

- To detect the various types of helmet having different shapes and sizes.
- To detect both head and helmet separately.
- To detect the helmet in various conditions like two people sitting on a bike or single person sitting on bike with their varied combinations of helmet wearing.
- To get the maximum accuracy.
- To ensure that it satisfies the System Requirement Specifications (SRS).

6.2 Testing Scope

The scope of testing of this project extends to a great number of possible situations in which the dataset is captured. There are mainly three types of datasets possible. They are –

- Image – The datasets can be in the form of the images of the people driving the vehicle with or without the helmet.
- Video – The datasets can be the recorded video files taken by the camera on which the testing is performed.
- Live Streaming – We can also perform the tests on a live streamed video on which the testing is performed and the helmet is detected there itself.

The testing is also performed to differentiate between the head and the helmet.

6.3 Testing Principles

While testing we tried to use a variety of test cases including – videos taken by us on the roads, images taken by us, images the model had encountered while training and a real live stream from a traffic camera. We tried to test our system for robustness and adaptability to various kinds of input which mimic the real life scenarios that the system is sure to encounter.

6.4 Testing Methods Used

Testing methodologies are approaches to testing, from unit testing through system testing and beyond. There is no formally recognized body of testing methodologies, and very rarely will you ever find a unified set of definitions. But here are some common methodologies which we have used in our project

- **Unit testing**-The act of testing software at the most basic (object) level. We perform this testing at the most basic level by comparing the pictures and checking if the object has been accurately identified.
- **Functional testing**-Functional testing takes a user story or a product feature and tests all of the functionality contained within that feature. In our project, we see and check the two basic functionalities of our system, i.e. training and testing. We check if these two functions are being performed correctly and systematically or not.
- **System Testing**-Testing the project as a collective system. We check if the system collectively performs both training and testing. And if the outputs of training are helpful to perform functions in the testing.
- **Performance Testing**-The performance of our system is checked by accurately resizing and renaming the input data sets before training and by validating if the GUI is helpful in giving the screenshots of results or not.
- **Black box Testing**

Black Box Testing is done considering only the inputs and outputs and without going into the internal working of the system. In our project, we compared the input and the output pictures and checked if the detection is accurate or not. The bounding boxes which are made with the help of COCO API are to be fitted accurately around the helmets of various head shapes. And the head and the helmet need to be detected separately. That is tested by simply looking at the input and output pictures.
- **White Box Testing**

White box testing is when the tester has access to the internal data structures and algorithms including the code. In this testing, we test our internal code with the help of accuracy that we get after running each module. The length and width of the bounding box is changed according to the requirements. We used Interactive Python Notebook – Jupyter Notebook for most of our White Box Testing. Since we had access to the code, we were able to optimize it on the fly as after each round of testing.

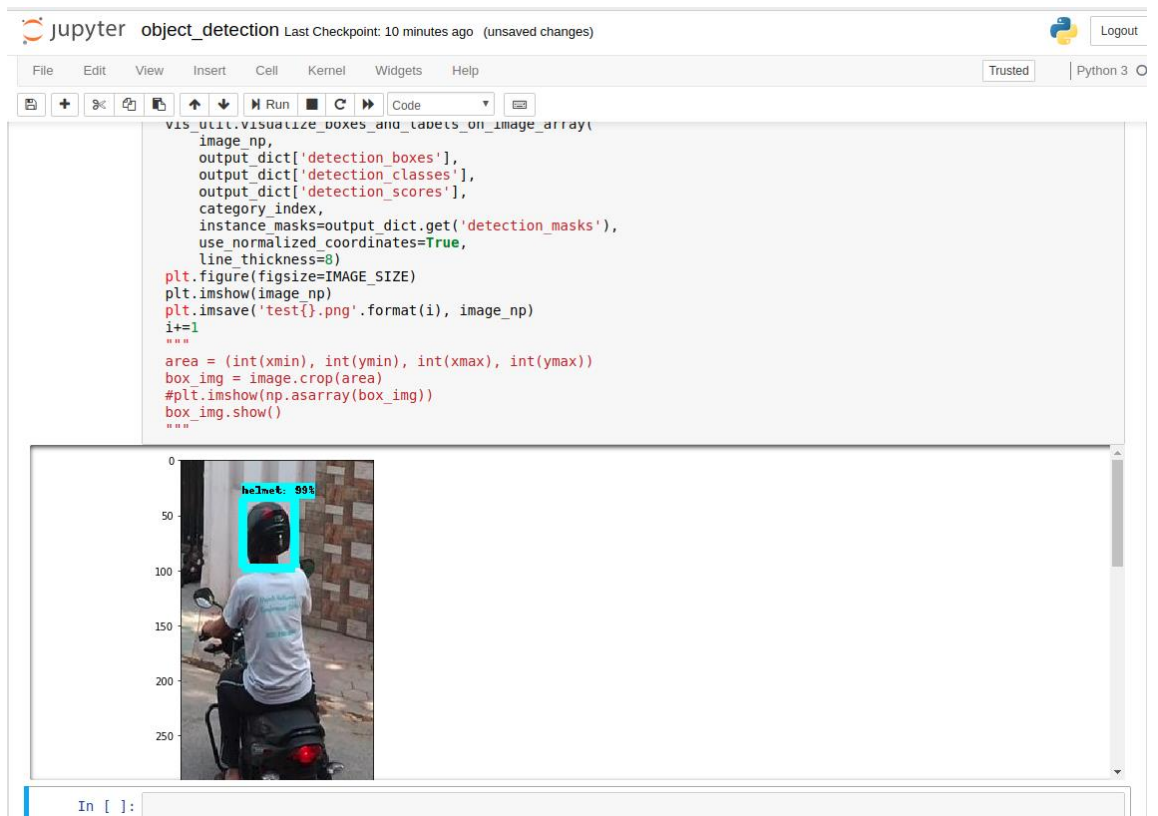


Fig. 14 – White Box Testing through Jupyter Notebook

6.5 Test Cases

There are various possibilities of test cases in our project. They are mentioned below –

- Sometimes the system considers the head and helmet to be the same because of the black hair of person which is similar in shape to that of the helmet. Therefore, various conditions of images need to be kept in mind performing the tests.
- There can be a single person driving the vehicle with him/her wearing the helmet.
- There can be a single person driving the vehicle with him/her not wearing the helmet.
- There can be two people sitting on a 2-wheeler with both of them wearing the helmet.
- There can be two people sitting on 2-wheeler with both of them not wearing the helmet.
- There can be two people sitting on 2-wheeler with one person wearing a helmet and the other not.

6.6 Sample Test Data and Result

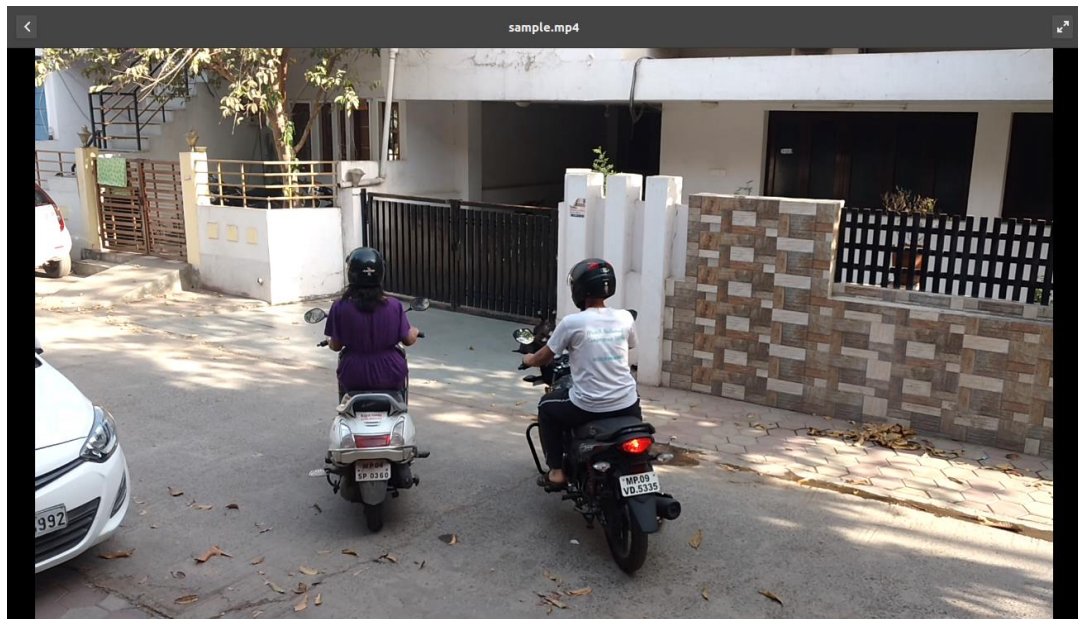


Fig. 15 – Sample Videostream

This is the sample data which we took to perform the testing on our system. In this image, two people are detected.

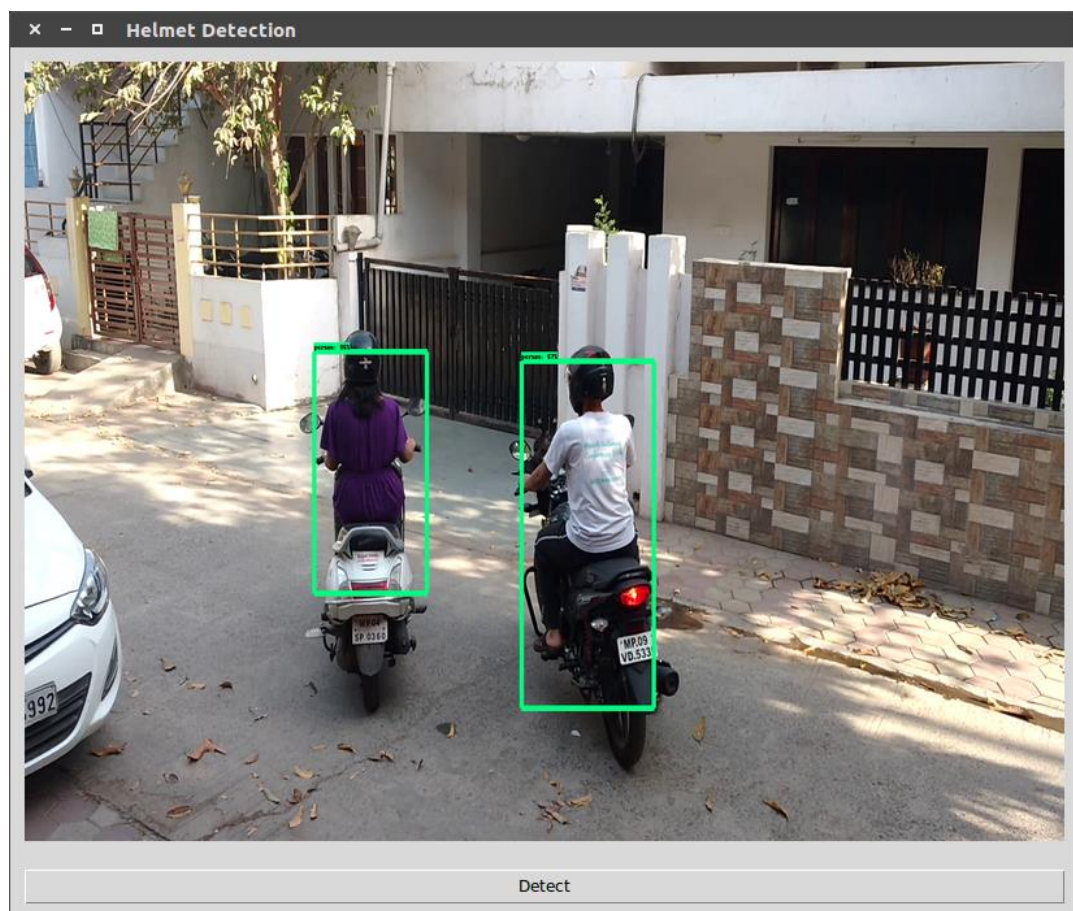


Fig. 16 – Sample Videostream Detection

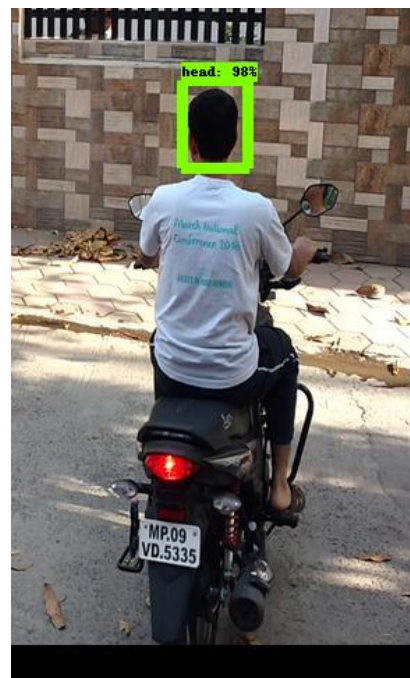


Fig.17 – Detected Person

After detecting a person, the system saves the portion of the image where the person is present and then runs helmet detection on this image as can be seen.



(a)



(b)

Fig. 18 - (a) Detected Helmet (b) No Helmet Detected

7. LIMITATIONS

- Specific hardware and software is used to carry out the mechanism of the system. If the input data set increases, the hardware specifications need also to be increased. Otherwise it will cause data congestion.
- A wide variety of data sets are required for the training data sets as there can be multiple variety of heads and helmets. The system should go through each possible image with different permutations and combinations of positions so that it gives no error while testing the data set.
- The quality of the image or video may be extremely low that it can become difficult to recognize the objects.
- The resizing of images can pose a problem sometimes if the dimensions are not given accurately.

8. FUTURE SCOPE

The future scope of this project can be extended to detecting the license plate of the vehicles as well. So that, when any person who is not wearing helmet is detected, his license plate number can be generated which will help in generating e-challan. This will completely reduce the manual work of the traffic police department as the challans are manually generated in the present scenario. Also, the scope of this concept can be extended to detecting the presence of people wearing car seat belts.

9. CONCLUSION

It is a tiresome job for the traffic policemen to check if people are wearing helmets or not. Specifically, in Indore, the systems have not been upgraded to a level where this process can become automatic. Also, the cameras that have been installed on various points are not being used to the full potential. Therefore, this system would help the government to make their system more technologically advanced and get benefits from it. We have just initiated a step to upgrade the system to a level which results in helping both the citizens as well as the authorities.

10. BIBLIOGRAPHY AND REFERENCES

- 1) <https://pythonprogramming.net/custom-objects-tracking-tensorflow-object-detection-api-tutorial/>
- 2) <https://github.com/tzutalin/labelImg>
- 3) https://github.com/tensorflow/models/tree/master/research/object_detection
- 4) <https://www.tensorflow.org/>
- 5) <https://docs.python.org/2/library/tkinter.html>