

**An Industry Oriented Mini Project Report**

**On**

**Helmet Detection using Machine Learning and Automatic License Plate Recognition**

Submitted in partial fulfillment of the  
Academic Requirement for the Award of  
Degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING  
(DATA SCIENCE)**

Submitted By

<b>R. HARSHITHA</b>	<b>22R01A67H6</b>
<b>SHAIK SHARIF</b>	<b>22R01A67J3</b>
<b>V.PRANAY VARSHAM</b>	<b>22R01A67J5</b>
<b>V.ARUN KUMAR</b>	<b>22R01A67J6</b>

Under the esteemed guidance of

**Dr. A. Nirmal Kumar**

**(Head of Department, CSE(Data Science))**



**CMR INSTITUTE OF TECHNOLOGY**

**(UGC AUTONOMUS)**

(Approved by AICTE, Affiliated to JNTUH, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

**[www.cmrithyderabad.edu.in](http://www.cmrithyderabad.edu.in)**

2024 – 2025

**CMR INSTITUTE OF TECHNOLOGY**

(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

[www.cmrithyderabad.edu.in](http://www.cmrithyderabad.edu.in)



This is to certify that a Micro Project entitled with: **“Helmet Detection using Machine Learning and Automatic License Plate Recognition”** is being

Submitted By

<b>R. HARSHITHA</b>	<b>22R01A67H6</b>
<b>SHAIK SHARIF</b>	<b>22R01A67J3</b>
<b>V.PRANAYVARSHAN</b>	<b>22R01A67J5</b>
<b>V.ARUN KUMAR</b>	<b>22R01A67J6</b>

To JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree of B.Tech in CSE(Data Science) and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted have any other University forward of any other degree or diploma.

**Signature of Guide**  
**Mr. A.Rajesh**  
(Asst Prof, CSE(DS))

**Signature of Project Coordinator**  
**Mr. M. Raj Kumar**  
(Asst Prof, CSE(DS))

**Signature of HOD**  
**Dr. A. Nirmal Kumar**  
(HOD, CSE(DS))

**EXTERNAL EXAMINER**

**IOMP**

**CSE(DATA SCIENCE)**

## **ACKNOWLEDGEMENT**

We are extremely grateful to **Dr. M. Janga reddy, Director, Dr. G. Madhusudhanarao, Principal** and **Dr. A. Nirmal Kuamr, Head of Department**, Dept of Computer Science and Engineering(Data Science) CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to Mini Project Coordinator and internal guide **Dr. A. Nirmal Kumar**, Department of Computer Science and Engineering(Data Science), CMR Institute of Technology for their constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

<b>R. HARSHITHA</b>	<b>22R01A67H6</b>
<b>SHAIK SHARIF</b>	<b>22R01A67J3</b>
<b>V.PRANAY VARSHAN</b>	<b>22R01A67J5</b>
<b>V.ARUN KUMAR</b>	<b>22R01A67J6</b>

## ABSTRACT

Motorcycle accidents have been rapidly growing through the years in many countries. In India more than 37 million people use two wheelers. Therefore, it is necessary to develop a system for automatic detection of helmet wearing for road safety. Therefore, a custom object detection model is created using a Machine learning based algorithm which can detect Motorcycle riders. On the detection of a Helmetless rider, the License Plate is extracted and the License Plate number is recognized using an Optical Character Recognizer. This Application can be implemented in real-time using a dataset as input. In recent years, the number of motorcycle-related accidents has been rising rapidly, particularly in countries with high two-wheeler usage like India, where more than 37 million people use motorcycles and scooters for daily commute. One of the leading causes of fatalities and serious injuries in such accidents is the non-compliance with safety gear regulations—specifically, not wearing helmets.

To address this pressing issue and enhance road safety, this project proposes the development of an intelligent system that can automatically detect whether motorcycle riders are wearing helmets. The system leverages advanced **Machine Learning (ML)** and **Computer Vision** techniques to detect and analyze riders in real-time video footage.

.

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>TABLE OF CONTENTS</b>	<b>iii</b>
<b>LIST OF FIGURES</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>v</b>
<b>1. Introduction</b>	<b>1</b>
<b>1.1 Objective of the project</b>	<b>1</b>
<b>2. Literature Survey</b>	<b>2</b>
<b>3. System Analysis</b>	<b>6</b>
<b>3.1 Existing System</b>	<b>5</b>
<b>3.2 Disadvantages of Existing System</b>	<b>6</b>
<b>3.3 Proposed System</b>	<b>6</b>
<b>3.4 Limitations of Advanced System</b>	<b>6</b>
<b>3.5 Advantages of Proposed System</b>	<b>7</b>
<b>4. System Design</b>	<b>16</b>
<b>4.1 Feasibility Study</b>	<b>16</b>
<b>4.1.1 Economical Feasibility</b>	<b>16</b>
<b>4.1.2 Operational Feasibility</b>	<b>16</b>
<b>4.1.3 Social feasibility</b>	<b>16</b>
<b>5. Implementation</b>	<b>24</b>
<b>6. Testing</b>	<b>36</b>
<b>7. Screenshots</b>	<b>40</b>
<b>8. Conclusion</b>	<b>43</b>
<b>9. References</b>	<b>44</b>

\

## LIST OF FIGURES

Figure no.	Particulars	Page no.
3.3	SDLC(Umbrella Model)	07
4.2	Class Diagram	15
4.3	Use Case Diagram	16
4.4	Sequence Diagram	17
4.5	Collaboration Diagram	18
4.6	Component Diagram	19
4.7	Deployment Diagram	34
4.8	Active Diagram	34
4.9	Data Flow Diagram	34
7.0	OutPuts	39,40,41,42,42

**LIST OF TABLES**

<b>Table no.</b>	<b>Particulars</b>	<b>Page no.</b>
6.7	Acceptance Testing	38,39

## **1. INTRODUCTION:**

The main safety equipment of motorcyclist is the helmet. The helmet protects the motorcyclist against accidents. Although the helmet use is mandatory in many countries, there are motorcyclists that do not use it or use it incorrectly. Over the past years many works have been carried out in traffic analysis, including vehicle detection and classification, and helmet detection. Intelligent traffic systems were implemented using computer vision algorithms, such as: background and foreground image detection to segment the moving objects in scene and image descriptors to extract features. Computational intelligence algorithms are used too, like machine learning algorithms to classify the objects. Machine learning (ML) is the field of Artificial Intelligence in which a trained model works on its own using the inputs given during training period. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions and are also used in the applications of object detection. 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 on 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.

### **1.1 Objective of the project:**

Motorcycle accidents have been rapidly growing through the years in many countries. In India more than 37 million people use two wheelers. Therefore, it is necessary to develop a system for automatic detection of helmet wearing for road safety. Therefore, a custom object detection model is created using a Machine learning based algorithm which can detect Motorcycle riders. On the detection of a Helmetless rider, the License Plate is extracted and the License Plate number is recognized using an Optical Character Recognizer. This Application can be implemented in real-time using a dataset as input.



## **2. LITERATURE SURVEY:**

### **1. “Robust Real-time Object Detection”**

This paper describes a visual object detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of a new image representation called the “Integral Image” which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features and yields extremely efficient classifiers [6]. The third contribution is a method for combining classifiers in a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. A set of experiments in the domain of face detection are presented. The system yields face detection performance comparable to the best previous systems [18, 13, 16, 12, 1]. Implemented on a conventional desktop, face detection proceeds at 15 frames per second.

### **2. “Histogram of oriented gradients for human detection”**

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

### **3. “Rich feature Hierarchy for Accurate object Detection”**

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012---achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity

convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset.

#### **4.“Fast R-CNN”**

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9x faster than R-CNN, is 213x faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3x faster, tests 10x faster, and is more accurate.

#### **5.“YOLO9000: Better, Faster, Stronger”**

We introduce YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories. First we propose various improvements to the YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL VOC and COCO. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP, outperforming state-of-the-art methods like Faster RCNN with ResNet and SSD while still running significantly faster. Finally we propose a method to jointly train on object detection and classification. Using this method we train YOLO9000 simultaneously on the COCO detection dataset and the ImageNet classification dataset. Our joint training allows YOLO9000 to predict detections for object classes that don't have labelled detection data. We validate our approach on the ImageNet detection task. YOLO9000 gets 19.7 mAP on the ImageNet detection validation set despite only having detection data for 44 of the 200 classes. On the 156 classes not in COCO, YOLO9000 gets 16.0 mAP. But YOLO can detect more than just 200 classes; it predicts detections for more than 9000 different object categories. And it still runs in real-time.

#### **6.“YOLOv3: An Incremental Improvement”**

We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At 320x320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57.9 mAP@50 in 51 ms on a Titan X, compared to 57.5 mAP@50 in 198 ms by RetinaNet, similar performance but 3.8x faster.

### **7.“SSD: Single Shot MultiBox Detector”**

We present a method for detecting objects in images using a single deep neural network. Our approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. Our SSD model is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stage and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, MS COCO, and ILSVRC datasets confirm that SSD has comparable accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. Compared to other single stage methods, SSD has much better accuracy, even with a smaller input image size. For 300×300 input, SSD achieves 72.1% mAP on VOC2007 test at 58 FPS on a Nvidia Titan X and for 500×500 input, SSD achieves 75.1% mAP, outperforming a comparable state of the art Faster R-CNN model.

### **8.“Robust real-time unusual event detection using multiple fixedlocation monitors,”**

We present a novel algorithm for detection of certain types of unusual events. The algorithm is based on multiple local monitors which collect low-level statistics. Each local monitor produces an alert if its current measurement is unusual and these alerts are integrated to a final decision regarding the existence of an unusual event. Our algorithm satisfies a set of requirements that are critical for successful deployment of any large-scale surveillance system. In particular, it requires a minimal setup (taking only a few minutes) and is fully automatic afterwards. Since it is not based on objects' tracks, It is robust and works well in crowded scenes

where tracking-based algorithms are likely to fail. The algorithm is effective as soon as sufficient low-level observations representing the routine activity have been collected, which usually happens after a few minutes. Our algorithm runs in real-time. It was tested on a variety of real-life crowded scenes. A ground-truth was extracted for these scenes, with respect to which detection and false-alarm rates are reported.

#### **9.“The safety helmet detection for atm’s surveillance system via the modified hough transform,”**

The automatic teller machine (ATM) plays an important role in the modern economical activity. It provides a fast and convenient way to process economical transactions between banks and their customers. Unfortunately, it also provides a convenient way for criminals to get illegal money. For the safety reason, each ATM is with the surveillance system to record customer's face information. However, when criminals use the ATM to withdraw illegal money, they usually hide their faces with something (e.g. safety helmets) to avoid that the surveillance system records their face information. That will make the surveillance system decrease their efficiency. We propose a circle/circular arc detection method based upon the modified Hough transform, and apply it to the detection of safety helmet for the surveillance system of the ATM. Since the safety helmet location will be in the set of the obtained possible circles/circular arcs (if any exists). We use geometric features to verify if any safety helmet exists in the set. The proposed method can be applied to the surveillance systems of ATMs and banks, and it can provide the early warning to save-guards when any "customer" tries to avoid his/her face information from surveillance, such as withdrawing money with the safety helmet. That will make the surveillance system more useful. A real ATM image is used to see the performance of proposed method.

#### **10.“Motorcycle detection and tracking system with occlusion segmentation,”**

This paper proposes a vision-based motorcycle monitoring system to detect and tracking motorcycles. The system proposes an occlusion detection and segmentation method. The method uses the visual length, visual width, and Pixel Ratio to detect the classes of the motorcycle occlusions and segment the motorcycle from each occlusive class. Because the motorcycle riders must put on their helmets, the helmet detection or search method is used to make sure whether the helmet/motorcycle exists or not. Experiments obtained by using complex road scenes are reported, which demonstrate the validity of the method in terms of robustness, accuracy, and time responses.

### **3. SYSTEM ANALYSIS**

#### **3.1 Existing System**

. Existing system monitors the traffic violations primarily through CCTV recordings, where the traffic police have to look into the frame where the traffic violation is happening, zoom into the license plate in case rider is not wearing helmet. But this requires lot of manpower and time as the traffic violations frequently and the number of people using motorcycles is increasing day-by-day. What if there is a system, which would automatically look for traffic violation of not wearing helmet while riding motorcycle/moped and if so, would automatically extract the vehicles' license plate number. The License plate extraction code extracts only from the motor bikes which has a rider who is not wearing helmet and discards the License plate of the motor bikes whose rider has helmet. The OCR model is able to detect and recognize the License plates present in an image with an accuracy up to 85 percent.

#### **3.2 Disadvantages of Existing System:**

1.Accuracy is less.

#### **3.3 Proposed System**

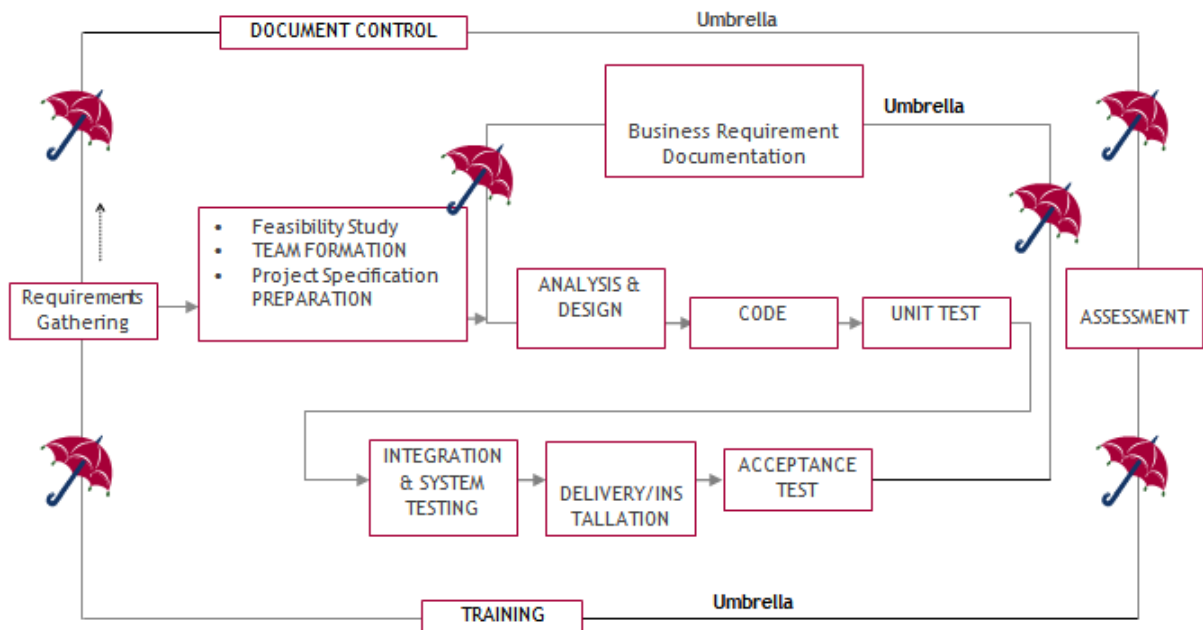
In This Paper, we a system to detect moving objects using a KNN classifier over the motorcyclist's head to classify helmet. These models were proposed based on statistical information of images and had a limitation to the level of accuracy that could be achieved. With the evolution of neural networks and deep learning models there was further improvement in the accuracy of classification. Introduced a convolutional neural network (CNN) based method for object classification and detection. Use a CNN for classification of helmeted and non-helmeted riders. Although they use CNN, their helmet detection accuracy is poor with limitations to helmet color and multiple riders on a single motorcyclist. 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. YOLO is a state-of-the-art, real-time object detection system. YOLOv3 is extremely fast and accurate and is a huge improvement over the previous YOLO versions.

### 3.4 Advantages of Proposed System:

1.Accuracy is more.

### PROCESS MODEL USED WITH JUSTIFICATION

#### 3.4.1 SDLC (Umbrella Model):



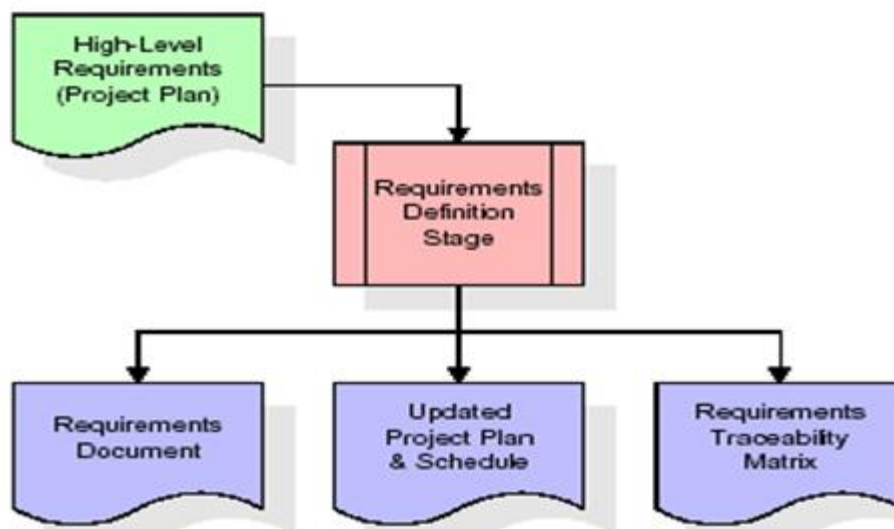
The requirements gathering process takes as its input SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

#### Stages in SDLC:

- ◆ Requirement Gathering
- ◆ Analysis
- ◆ Designing
- ◆ Coding
- ◆ Testing
- ◆ Maintenance

**Requirements Gatheringstage:**

the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product

components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

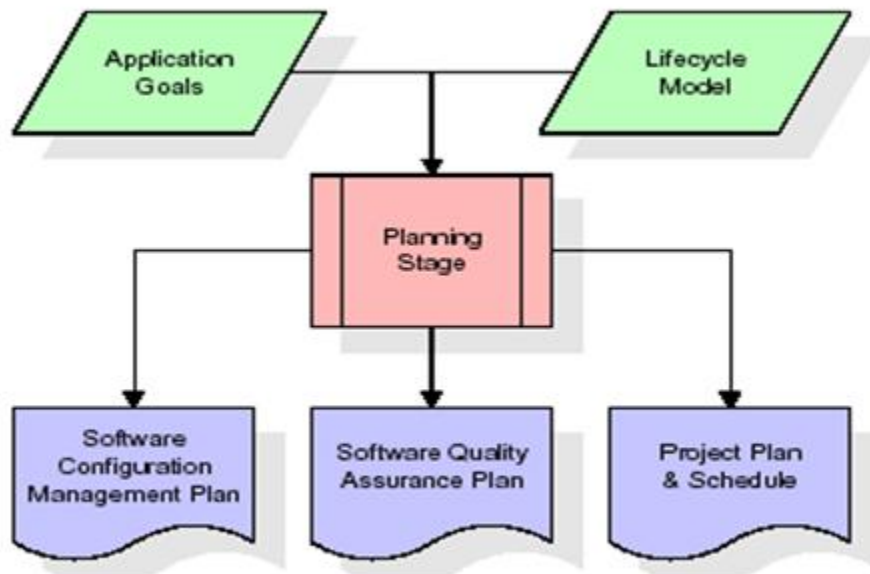
The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- ◆ Feasibility study is all about identification of problems in a project.
- ◆ No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- ◆ Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

### **Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

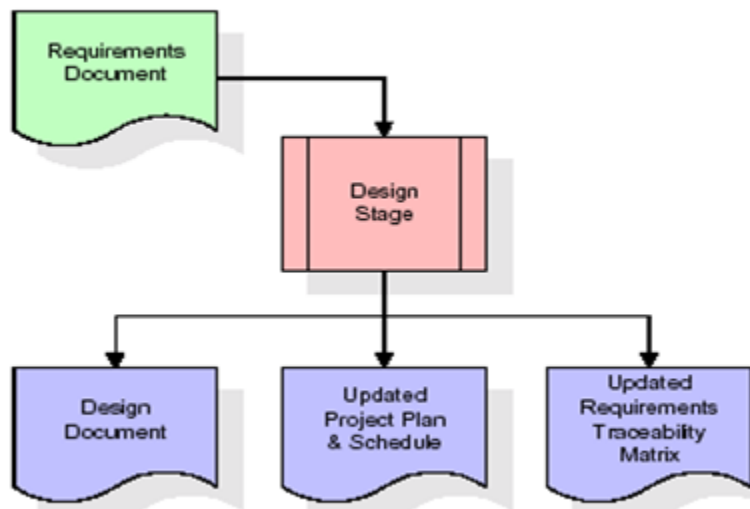




The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

### **Designing Stage:**

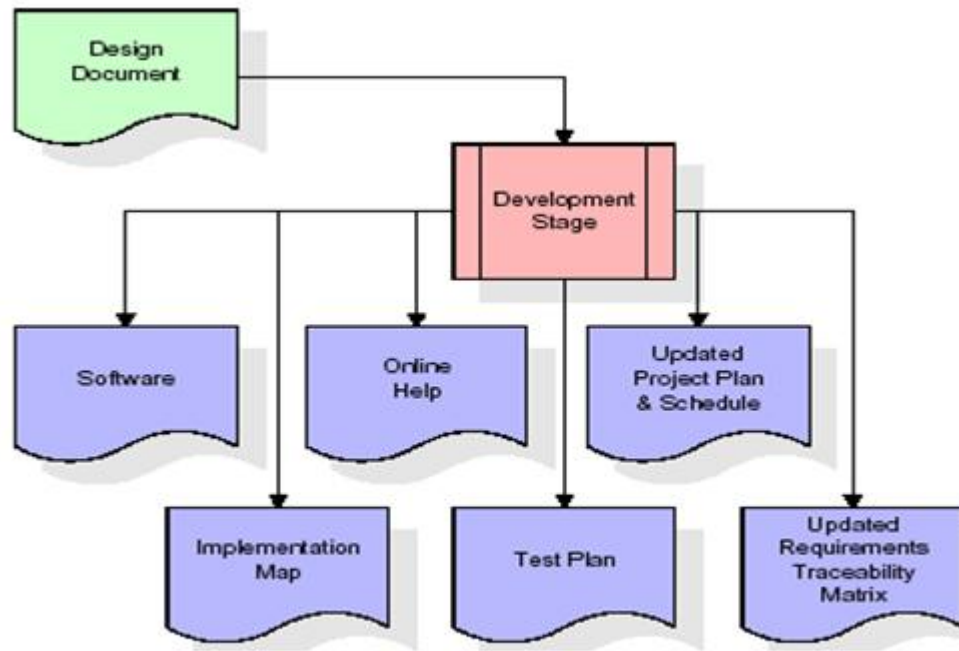
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

### **Development (Coding) Stage:**

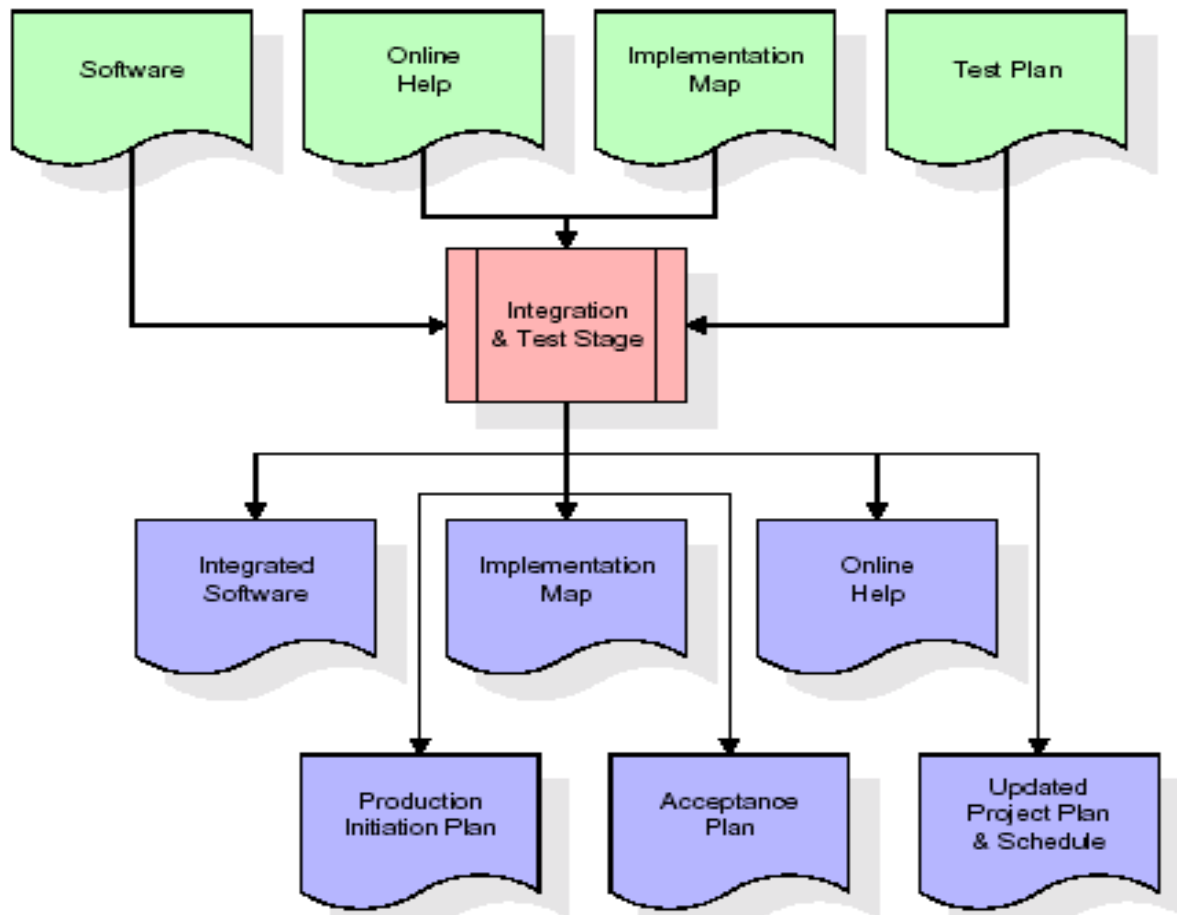
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

### **Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

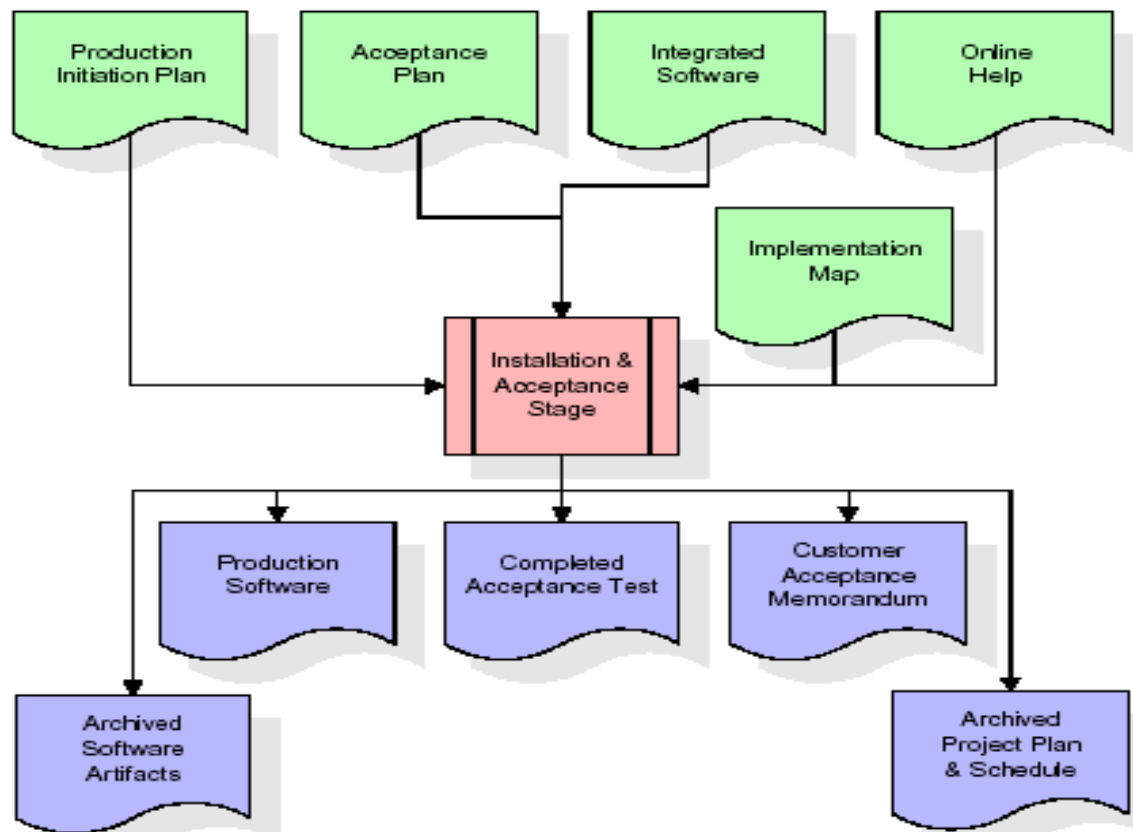


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

### **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

#### **Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

### Software Requirement Specification

#### Overall Description

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms *what* must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited

resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

## **4. SYSTEM DESIGN**

### **4.1 ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

#### **4.1.2 OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would

ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

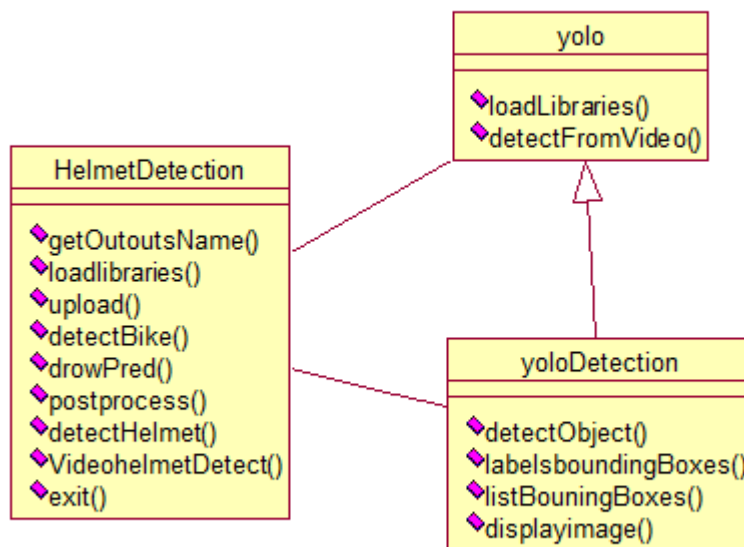
#### **4.1.3 TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**UML Diagrams:****4.2 Class Diagram:**

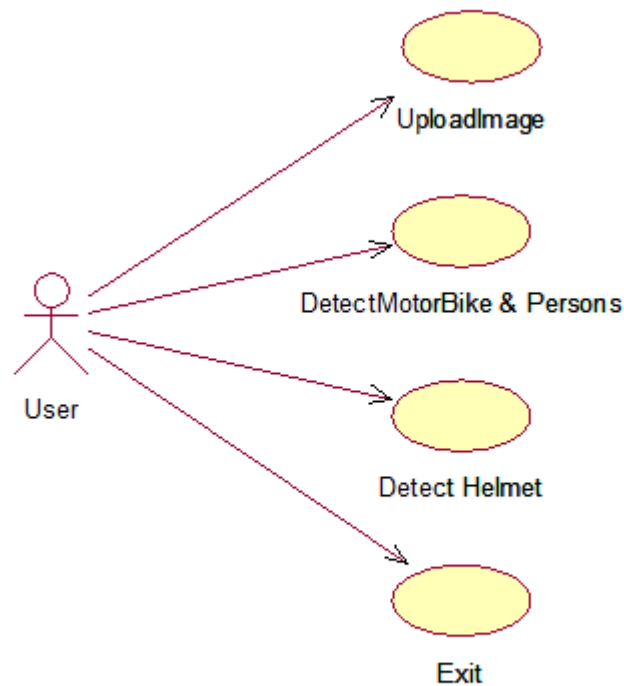
The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake.

**4.3 Use case Diagram:**

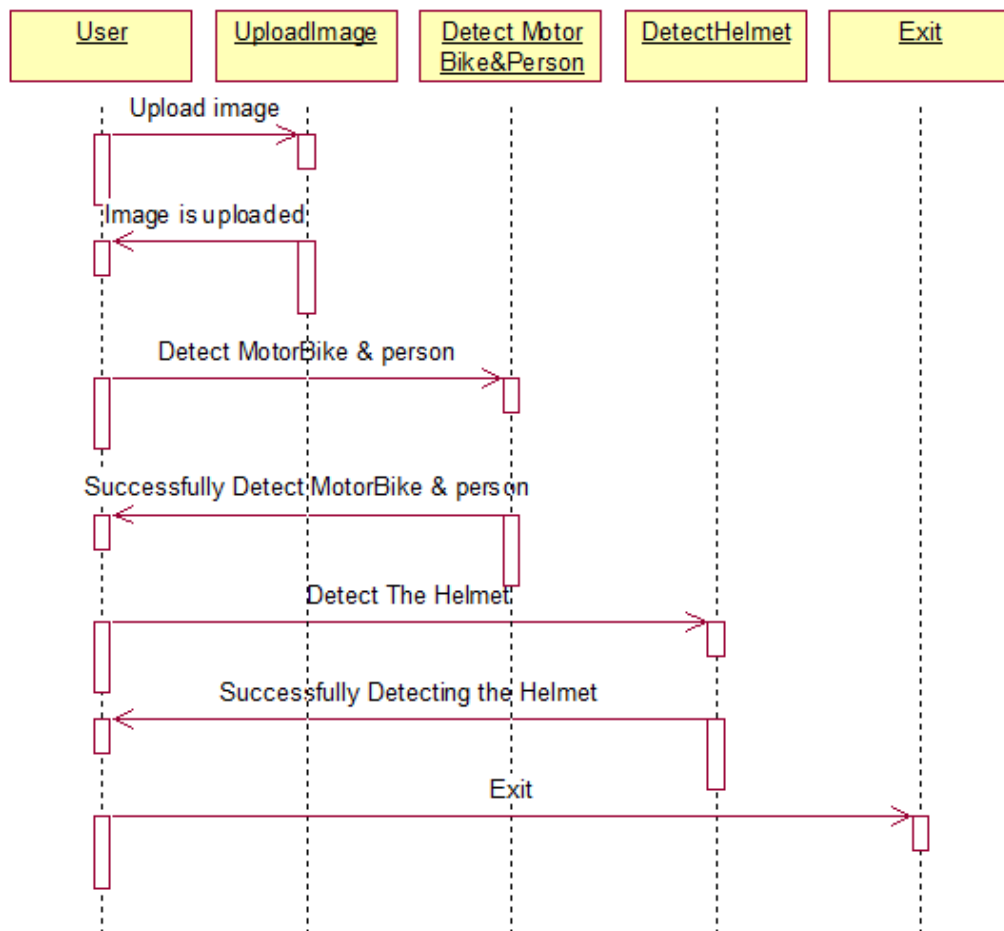
A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.





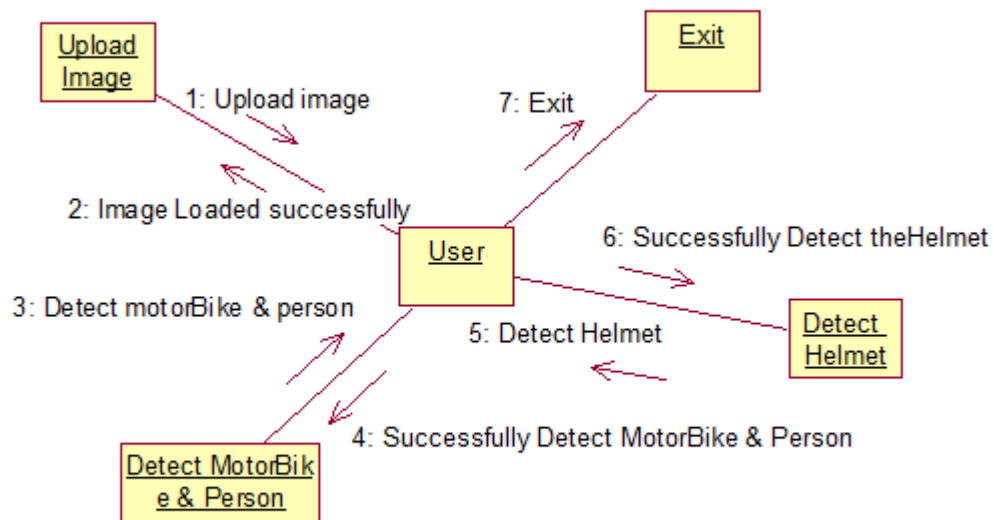
#### 4.4 Sequence Diagram:

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



#### 4.5 Collaboration diagram:

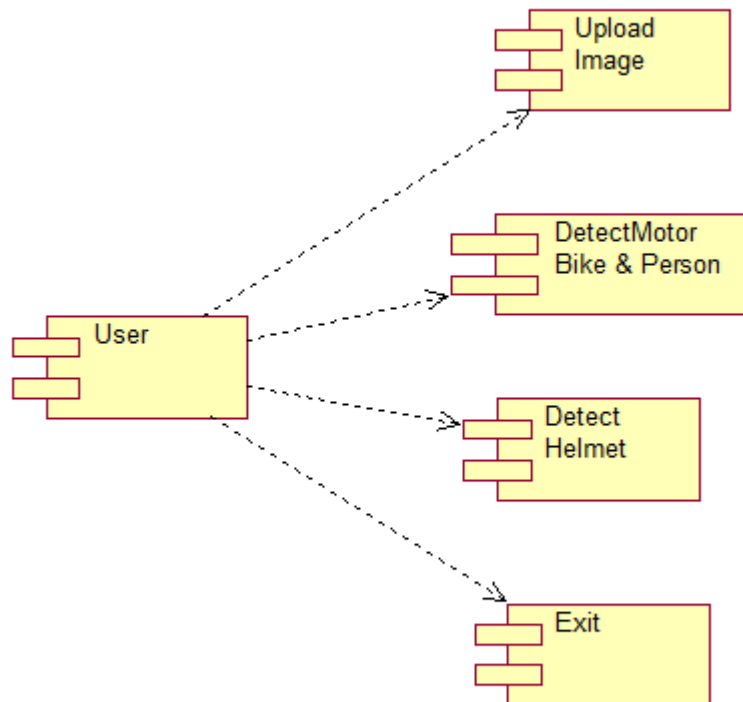
A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



#### 4.6 Component Diagram:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



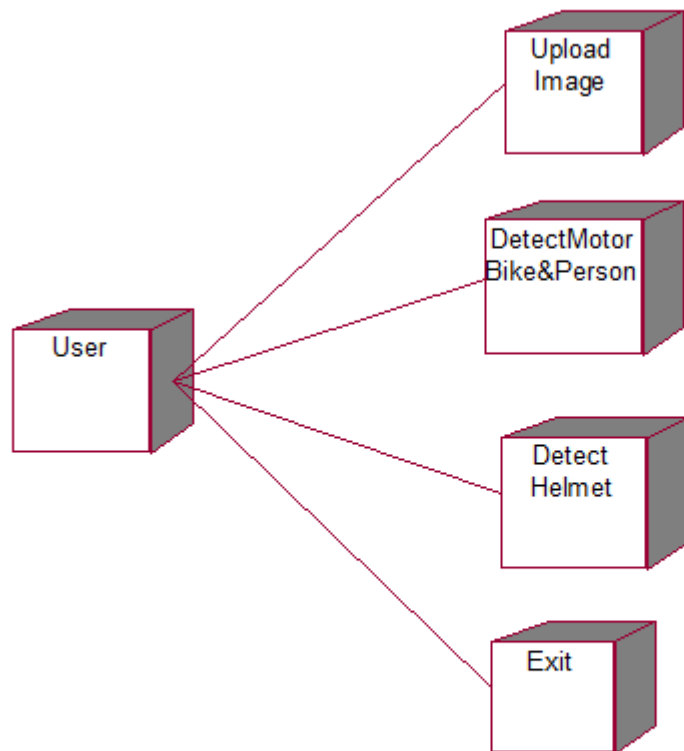
#### 4.7 Deployment Diagram:

A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster

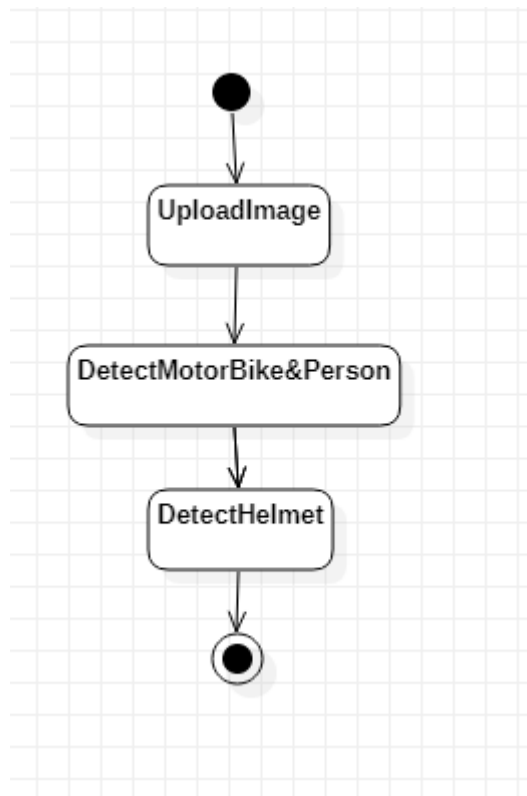
of

databaseservers.



#### **4.8 Activity Diagram:**

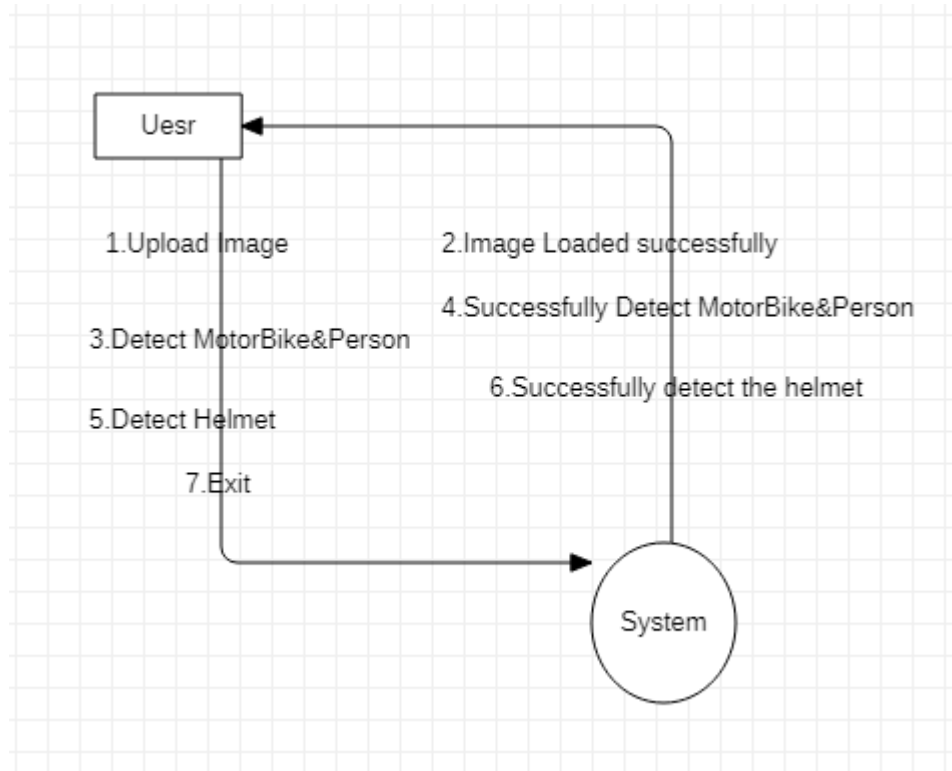
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



#### 4.9 Data Flow Diagram:

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



## 5. IMPEMETATION

### 5.1 Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

#### History of Python:

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

#### Why Python was created?

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a

language that was extensible. This led to design of a new language which was later named Python.

### **Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

### **Features of Python:**

#### **A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

#### **Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

#### **Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

#### **Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.



### **A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

### **Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using `import MySQLdb`.

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

### **Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

### **Applications of Python:**

#### **1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

```
a = 2
```

```
b = 3
```

```
sum = a + b
```

```
print(sum)
```

## 2. Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

## 3. Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

## 4. Great Community and Support

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

## 5.2 Sample Code:

### yolo.py

```
import numpy as np
import cv2 as cv
import subprocess
import time
import os
from yoloDetection import detectObject, displayImage
import sys

global class_labels
global cnn_model
global cnn_layer_names

def loadLibraries(): #function to load yolov3 model weight and class
labels
    global class_labels
    global cnn_model
    global cnn_layer_names
    class_labels = open('yolov3model/yolov3-
labels').read().strip().split('\n') #reading labels from yolov3 model
    print(str(class_labels)+" == "+str(len(class_labels)))
    cnn_model = cv.dnn.readNetFromDarknet('yolov3model/yolov3.cfg',
'yolov3model/yolov3.weights') #reading model
```

```
cnn_layer_names = cnn_model.getLayerNames() #getting layers
from cnn_model
cnn_layer_names = [cnn_layer_names[i[0] - 1] for i in
cnn_model.getUnconnectedOutLayers()] #assigning all layers

def detectFromImage(imagename): #function to detect object from images
    #random colors to assign unique color to each label
    label_colors =
(0,255,0)#np.random.randint(0,255,size=(len(class_labels),3),dtype='uint8')

    try:
        image = cv.imread(imagename) #image reading
        image_height, image_width = image.shape[:2] #converting
image to two dimensional array
    except:
        raise 'Invalid image path'
    finally:
        image, _, _, _, _ = detectObject(cnn_model,
cnn_layer_names, image_height, image_width, image, label_colors,
class_labels,indexno)#calling detection function
        displayImage(image,0)#display image with detected
objects label

def detectFromVideo(videoFile): #function to read objects from video

    #random colors to assign unique color to each label
    label_colors =
(0,255,0)#np.random.randint(0,255,size=(len(class_labels),3),dtype='uint8')

    indexno = 0
    try:

        video = cv.VideoCapture(videoFile)
        frame_height, frame_width = None, None #reading video
from given path
        video_writer = None
    except:
        raise 'Unable to load video'
    finally:
        while True:
            frame_grabbed, frames = video.read() #taking
each frame from video
            #print(frame_grabbed)
            if not frame_grabbed: #condition to check
whether video loaded or not
                break
            if frame_width is None or frame_height is None:
                frame_height, frame_width =
frames.shape[:2] #detecting object from frame
            frames, _, _, _, _ = detectObject(cnn_model,
cnn_layer_names, frame_height, frame_width, frames, label_colors,
class_labels,indexno)

            #displayImage(frames,index)
            #indexno = indexno + 1
            print(indexno)
            if indexno == 5:
                video.release()
                break
```

```
print ("Releasing resources")
#video_writer.release()
video.release()

if __name__ == '__main__':
    loadLibraries()
    print("sample commands to run code with image or video")
    print("python yolo.py image input_image_path")
    print("python yolo.py video input_video_path")
    if len(sys.argv) == 3:
        if sys.argv[1] == 'image':
            detectFromImage(sys.argv[2])
        elif sys.argv[1] == 'video':
            detectFromVideo(sys.argv[2])
        else:
            print("invalid input")
    else:
        print("follow sample command to run code")

#video_path = None
#video_output_path = "out.avi"
```

### **HelmetDetection.py**

```
from tkinter import *
import tkinter
from tkinter import filedialog
import numpy as np
from tkinter.filedialog import askopenfilename
import pandas as pd
from tkinter import simpledialog
import numpy as np
import cv2 as cv
import subprocess
import time
import os
from yoloDetection import detectObject, displayImage
import sys
from time import sleep
from tkinter import messagebox
import pytesseract as tess
from keras.models import model_from_json
from keras.utils.np_utils import to_categorical

main = tkinter.Tk()
main.title("Helmet Detection") #designing main screen
main.geometry("800x700")

global filename
global loaded_model
```

```
global class_labels
global cnn_model
global cnn_layer_names

frame_count = 0
frame_count_out=0

confThreshold = 0.5
nmsThreshold = 0.4
inpWidth = 416
inpHeight = 416
global option
labels_value = []
with open("Models/labels.txt", "r") as file: #reading MRC dictionary
    for line in file:
        line = line.strip('\n')
        line = line.strip()
        labels_value.append(line)
    file.close()

with open('Models/model.json', "r") as json_file:
    loaded_model_json = json_file.read()
    plate_detector = model_from_json(loaded_model_json)

plate_detector.load_weights("Models/model_weights.h5")
plate_detector._make_predict_function()

classesFile = "Models/obj.names";
classes = None

with open(classesFile, 'rt') as f:
    classes = f.read().rstrip('\n').split('\n')

modelConfiguration = "Models/yolov3-obj.cfg";
modelWeights = "Models/yolov3-obj_2400.weights";

net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)

def getOutputsNames(net):
    layersNames = net.getLayerNames()
    return [layersNames[i[0] - 1] for i in
net.getUnconnectedOutLayers()]

def loadLibraries(): #function to load yolov3 model weight and class
labels
    global class_labels
    global cnn_model
    global cnn_layer_names
    class_labels = open('yolov3model/yolov3-
labels').read().strip().split('\n') #reading labels from yolov3 model
    print(str(class_labels)+" == "+str(len(class_labels)))
    cnn_model = cv.dnn.readNetFromDarknet('yolov3model/yolov3.cfg',
'yolov3model/yolov3.weights') #reading model
    cnn_layer_names = cnn_model.getLayerNames() #getting layers
from cnn model
```

```

        cnn_layer_names = [cnn_layer_names[i[0] - 1] for i in
cnn_model.getUnconnectedOutLayers()] #assigning all layers

def upload(): #function to upload tweeter profile
    global filename
    filename = filedialog.askopenfilename(initialdir="bikes")
    #messagebox.showinfo("File Information", "image file loaded")

def detectBike():
    global option
    option = 0
    indexno = 0
    label_colors = (0,255,0)
    try:
        image = cv.imread(filename)
        image_height, image_width = image.shape[:2]
    except:
        raise 'Invalid image path'
    finally:
        image, ops = detectObject(cnn_model, cnn_layer_names,
image_height, image_width, image, label_colors, class_labels,indexno)
        if ops == 1:
            displayImage(image,0)#display image with detected objects
label
            option = 1
        else:
            displayImage(image,0)

def drawPred(classId, conf, left, top, right, bottom,frame,option):
    global frame_count
    #cv.rectangle(frame, (left, top), (right, bottom), (255, 178, 50),
3)
    label = '%.2f' % conf
    if classes:
        assert(classId < len(classes))
        label = '%s:%s' % (classes[classId], label)
        labelSize, baseLine = cv.getTextSize(label,
cv.FONT_HERSHEY_SIMPLEX, 0.5, 1)
        top = max(top, labelSize[1])
        label_name,label_conf = label.split(':')
        print(label_name+" == "+str(conf)+" == "+str(option))
        if label_name == 'Helmet' and conf > 0.50:
            if option == 0 and conf > 0.90:
                cv.rectangle(frame, (left, top - round(1.5*labelSize[1])),
(left + round(1.5*labelSize[0]), top + baseLine), (255, 255, 255),
cv.FILLED)
                cv.putText(frame, label, (left, top),
cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,0), 1)
                frame_count+=1
            if option == 0 and conf < 0.90:
                cv.putText(frame, "Helmet Not detected", (10, top),
cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,0), 2)

```

```

        frame_count+=1
        img = cv.imread(filename)
        img = cv.resize(img, (64,64))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,64,64,3)
        X = np.asarray(im2arr)
        X = X.astype('float32')
        X = X/255
        preds = plate_detector.predict(X)
        predict = np.argmax(preds)
        #img = cv.imread(filename)
        #img = cv.resize(img, (500,500))
        #text = tess.image_to_string(img, lang='eng')
        #text = text.replace("\n", " ")
        #messagebox.showinfo("Number Plate Detection Result",
"Number plate detected as "+text)
        textarea.insert(END,filename+"\n\n")
        textarea.insert(END,"Number plate detected as
"+str(labels_value[predict]))
        if option == 1:
            cv.rectangle(frame, (left, top - round(1.5*labelSize[1])),
(left + round(1.5*labelSize[0]), top + baseLine), (255, 255, 255),
cv.FILLED)
            cv.putText(frame, label, (left, top),
cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,0), 1)
            frame_count+=1

    if(frame_count> 0):
        return frame_count

def postprocess(frame, outs, option):
    frameHeight = frame.shape[0]
    frameWidth = frame.shape[1]
    global frame_count_out
    frame_count_out=0
    classIds = []
    confidences = []
    boxes = []
    classIds = []
    confidences = []
    boxes = []
    cc = 0
    for out in outs:
        for detection in out:
            scores = detection[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                center_x = int(detection[0] * frameWidth)
                center_y = int(detection[1] * frameHeight)
                width = int(detection[2] * frameWidth)
                height = int(detection[3] * frameHeight)
                left = int(center_x - width / 2)
                top = int(center_y - height / 2)
                classIds.append(classId)
                #print(classIds)
                confidences.append(float(confidence))

```

```

        boxes.append([left, top, width, height])

    indices = cv.dnn.NMSBoxes(boxes, confidences, confThreshold,
nmsThreshold)
    count_person=0 # for counting the classes in this loop.
    for i in indices:
        i = i[0]
        box = boxes[i]
        left = box[0]
        top = box[1]
        width = box[2]
        height = box[3]
        frame_count_out = drawPred(classIds[i], confidences[i], left,
top, left + width, top + height,frame,option)
        my_class='Helmet'
        unknown_class = classes[classId]
        print("===="+str(unknown_class))
        if my_class == unknown_class:
            count_person += 1

    print(str(frame_count_out))
    if count_person == 0 and option == 1:
        cv.putText(frame, "Helmet Not detected", (10, 50),
cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,0), 2)
    if count_person >= 1 and option == 0:
        #path = 'test_out/'
        #cv.imwrite(str(path)+str(cc)+".jpg", frame)          # writing to
folder.
        #cc = cc + 1
        frame = cv.resize(frame, (500,500))
        cv.imshow('img',frame)
        cv.waitKey(50)

def detectHelmet():
    textarea.delete('1.0', END)
    if option == 1:
        frame = cv.imread(filename)
        frame_count =0
        blob = cv.dnn.blobFromImage(frame, 1/255, (inpWidth,
inpHeight), [0,0,0], 1, crop=False)
        net.setInput(blob)
        outs = net.forward(getOutputsNames(net))
        postprocess(frame, outs,0)
        t, _ = net.getPerfProfile()
        label = 'Inference time: %.2f ms' % (t * 1000.0 /
cv.getTickFrequency())
        print(label)
        cv.putText(frame, label, (0, 15), cv.FONT_HERSHEY_SIMPLEX, 0.5,
(0, 0, 255))
        print(label)
    else:
        messagebox.showinfo("Person & Motor bike not detected in
uploaded image", "Person & Motor bike not detected in uploaded image")

def videoHelmetDetect():
    global filename

```



```

videofile = askopenfilename(initialdir = "videos")
video = cv.VideoCapture(videofile)
while(True):
    ret, frame = video.read()
    if ret == True:
        frame_count = 0
        filename = "temp.png"
        cv.imwrite("temp.png", frame)
        blob = cv.dnn.blobFromImage(frame, 1/255, (inpWidth,
inpHeight), [0,0,0], 1, crop=False)
        net.setInput(blob)
        outs = net.forward(getOutputsNames(net))
        postprocess(frame, outs, 1)
        t, _ = net.getPerfProfile()
        #label=''
        #cv.putText(frame, label, (0, 15), cv.FONT_HERSHEY_SIMPLEX,
0.5, (0, 0, 255))
        cv.imshow("Predicted Result", frame)
        if cv.waitKey(5) & 0xFF == ord('q'):
            break
    else:
        break
video.release()
cv.destroyAllWindows()

def exit():
    global main
    main.destroy()

font = ('times', 16, 'bold')
title = Label(main, text='Number Plate Detection without Helmet',
justify=LEFT)
title.config(bg='lavender blush', fg='DarkOrchid1')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=100,y=5)
title.pack()

font1 = ('times', 14, 'bold')
model = Button(main, text="Upload Image", command=upload)
model.place(x=200,y=100)
model.config(font=font1)

uploadimage = Button(main, text="Detect Motor Bike & Person",
command=detectBike)
uploadimage.place(x=200,y=150)
uploadimage.config(font=font1)

classifyimage = Button(main, text="Detect Helmet",
command=detectHelmet)
classifyimage.place(x=200,y=200)
classifyimage.config(font=font1)

exitapp = Button(main, text="Exit", command=exit)
exitapp.place(x=200,y=250)
exitapp.config(font=font1)

```

```
font1 = ('times', 12, 'bold')
textarea=Text(main,height=15,width=60)
scroll=Scrollbar(textarea)
textarea.configure(yscrollcommand=scroll.set)
textarea.place(x=10,y=300)
textarea.config(font=font1)
```

```
loadLibraries()
```

```
main.config(bg='light coral')
main.mainloop()
```

### voloDetection.py

```
import numpy as np
import argparse
import cv2 as cv
import subprocess
import time
import os

def detectObject(CNNnet, total_layer_names, image_height, image_width,
image, name_colors, class_labels,indexno,
                 Boundingboxes=None, confidence_value=None, class_ids=None,
ids=None, detect=True):

    if detect:
        blob_object = cv.dnn.blobFromImage(image,1/255.0,(416,
416),swapRB=True,crop=False)
        CNNnet.setInput(blob_object)
        cnn_outs_layer = CNNnet.forward(total_layer_names)
        Boundingboxes, confidence_value, class_ids =
listBoundingBoxes(cnn_outs_layer, image_height, image_width, 0.5)
        ids = cv.dnn.NMSBoxes(Boundingboxes, confidence_value, 0.5,
0.3)
        if Boundingboxes is None or confidence_value is None or ids is
None or class_ids is None:
            raise '[ERROR] unable to draw boxes.'
        image,option = labelsBoundingBoxes(image, Boundingboxes,
confidence_value, class_ids, ids, name_colors, class_labels,indexno)

    return image,option

def labelsBoundingBoxes(image, Boundingbox, conf_thr, classID, ids,
color_names, predicted_labels,indexno):
    option = 0
    if len(ids) > 0:
        for i in ids.flatten():
            # draw boxes
            xx, yy = Boundingbox[i][0], Boundingbox[i][1]
            width, height = Boundingbox[i][2], Boundingbox[i][3]

            class_color = (0,255,0)#[int(color) for color in
color_names[classID[i]]]

            cv.rectangle(image, (xx, yy), (xx+width, yy+height),
class_color, 2)
```

```

        print(classID[i])
        if classID[i] <= 1:
            text_label = "{}:
{:4f}".format(predicted_labels[classID[i]], conf_thr[i])
            #displayImage(image,indexno)
            cv.putText(image, text_label, (xx, yy-5),
cv.FONT_HERSHEY_SIMPLEX, 0.5, class_color, 2)
            option = 1

    return image,option

def listBoundingBoxes(image, image_height, image_width,
threshold_conf):
    box_array = []
    confidence_array = []
    class_ids_array = []

    for img in image:
        for obj_detection in img:
            detection_scores = obj_detection[5:]
            class_id = np.argmax(detection_scores)
            confidence_value = detection_scores[class_id]
            if confidence_value > threshold_conf and class_id <= 1:
                BoundingBox = obj_detection[0:4] * np.array([image_width,
image_height, image_width, image_height])
                center_X, center_Y, box_width, box_height =
BoundingBox.astype('int')

                xx = int(center_X - (box_width / 2))
                yy = int(center_Y - (box_height / 2))

                box_array.append([xx, yy, int(box_width),
int(box_height)])
                confidence_array.append(float(confidence_value))
                class_ids_array.append(class_id)

    return box_array, confidence_array, class_ids_array

def displayImage(image,index):
    #cv.imwrite('bikes/'+str(index)+'.jpg',image)
    #index = index + 1
    cv.imshow("Final Image", image)
    cv.waitKey(0)

```

## **6. TESTING**

### **6.1 Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautious because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link

together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## **6.2 Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modified as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## **6.3 Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system properly function as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

## **6.4 System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to withstand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**6.5 Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**6.6 Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**6.7 Acceptance Testing**

When that user find no major problems with its accuracy the system passes through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which eliminates wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

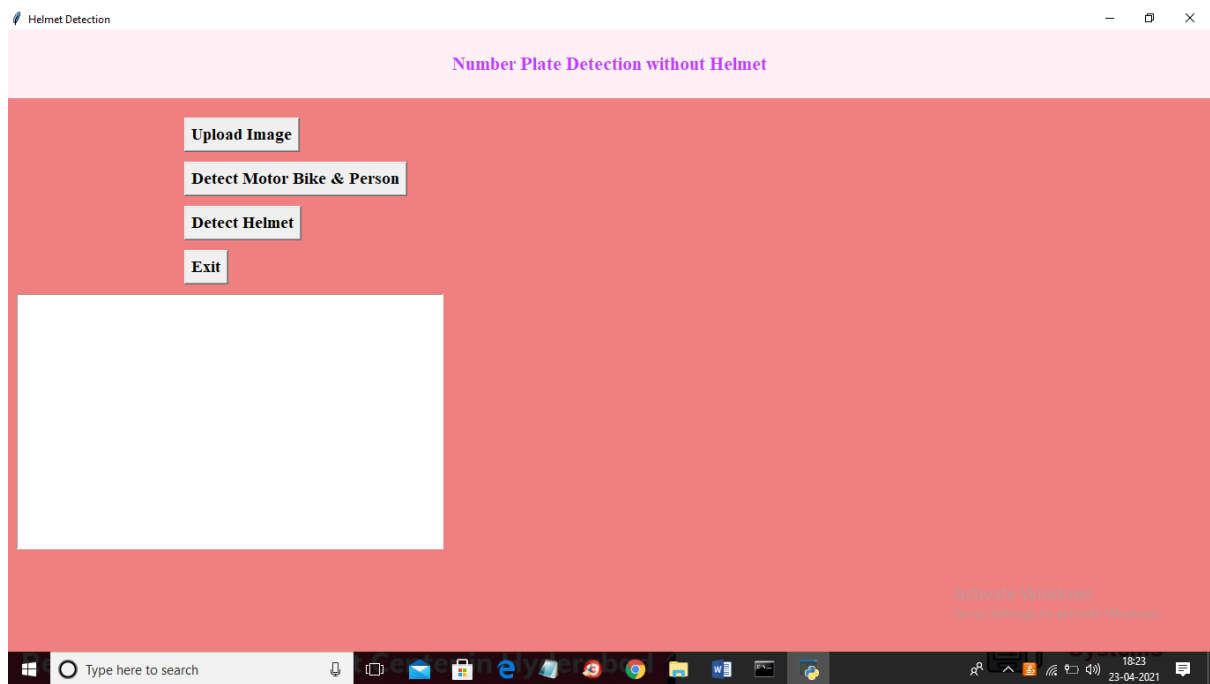
Test	Test case	Test Case Desc.				Test Case	Test Case
------	-----------	-----------------	--	--	--	-----------	-----------

Cas e ID	Name		Step	Expected	Actual	Statu s	Priorit y
01	Upload Image	Verify the Either Image is uploaded or not	If Image May Not be uploaded	We cannot do the further operations	Test Image is Uploaded	High	High
02	Detect Motorbike & Person	Verify The Motorbike & Person Detect Or not	If The Motorbike & Person May Not be Detected	We cannot do the further operations	Motorbike & person can be detected	High	High
03	Detect Helmet	Verify The Helmet Detect Or not	If The Helmet May Not be Detected	We cannot do the further operations	Helmet can be detected	High	High

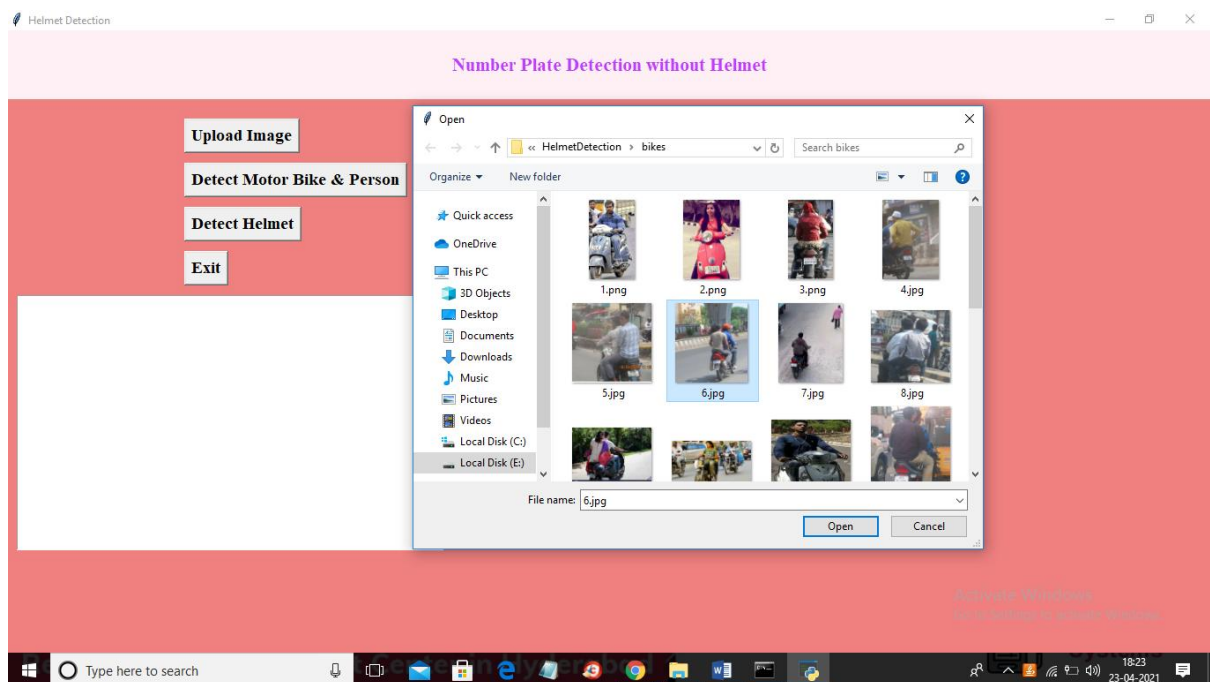
**7.SCREEN SHOTS:**

In this project we have built CNN model to detect HELMETS and number plates from 25 different images and we can detect more images but we don't have sufficient dataset to train CNN model so our application can detect presence of helmet from 25 different images and if helmet not present then it will identify number plate and if helmet detected then it will not identify number plate.

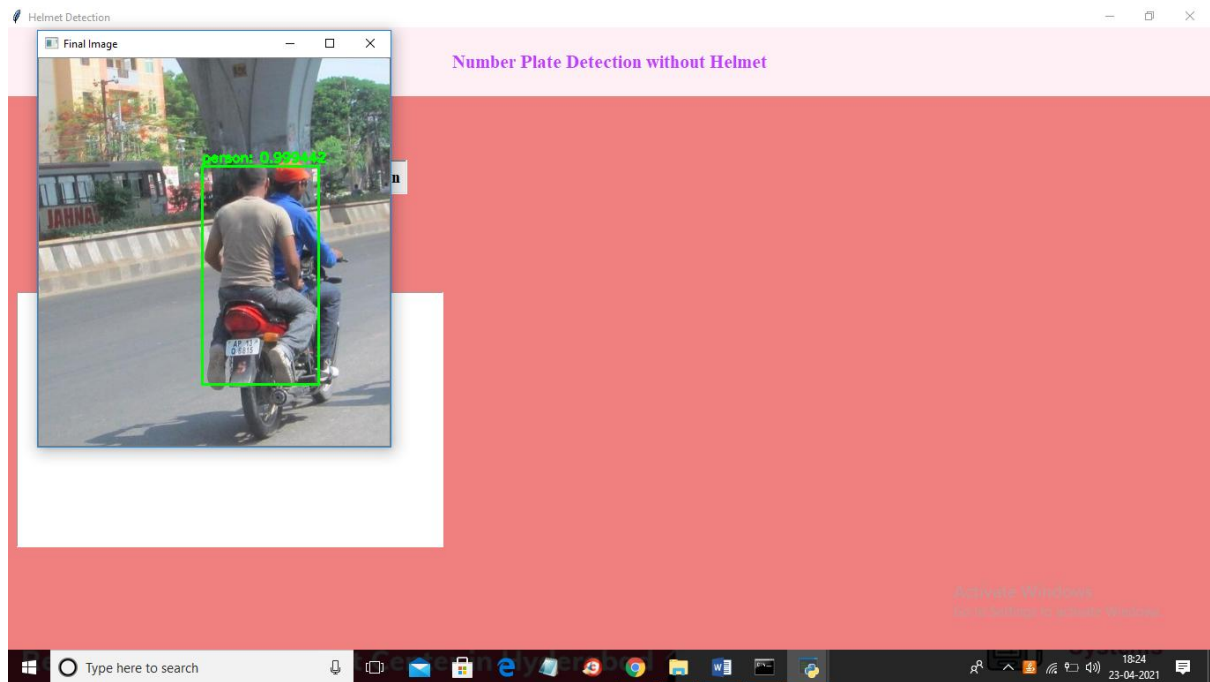
To run project double click on 'run.bat' file to get below screen



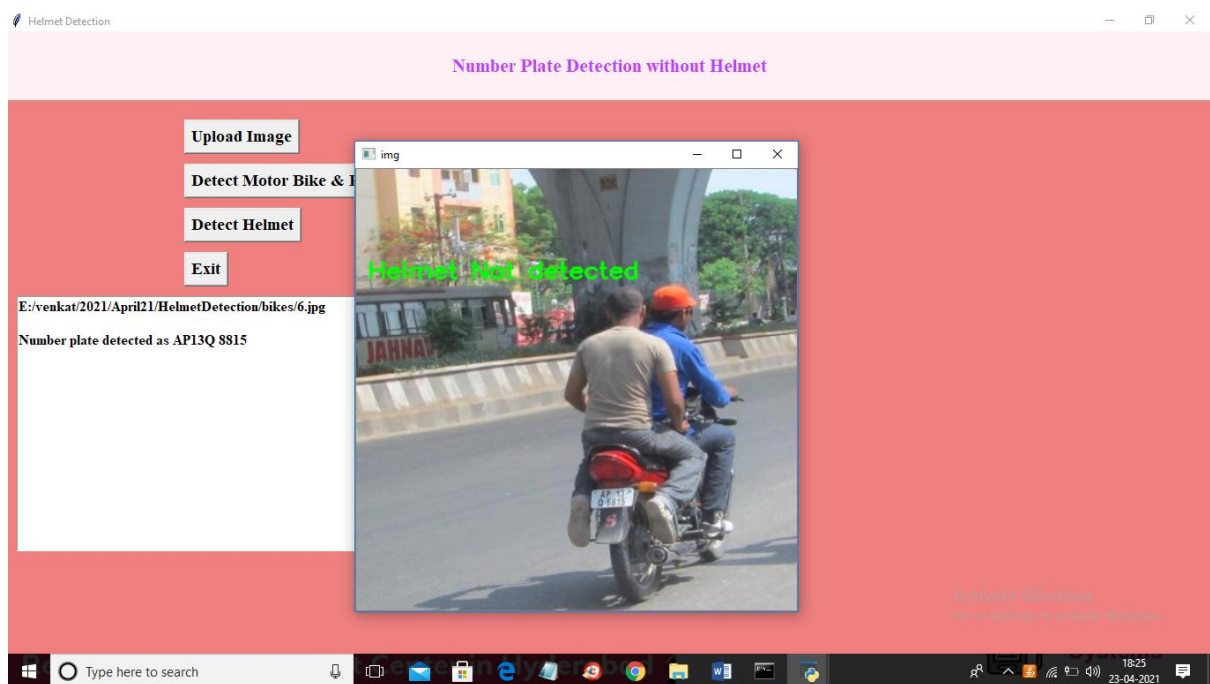
In above screen click on 'Upload Image' button to upload image



In above screen selecting and uploading '6.jpg' file and then click on 'Open' button to load image and then click on 'Detect Motor Bike & Person' button to detect whether image contains person with bike or not

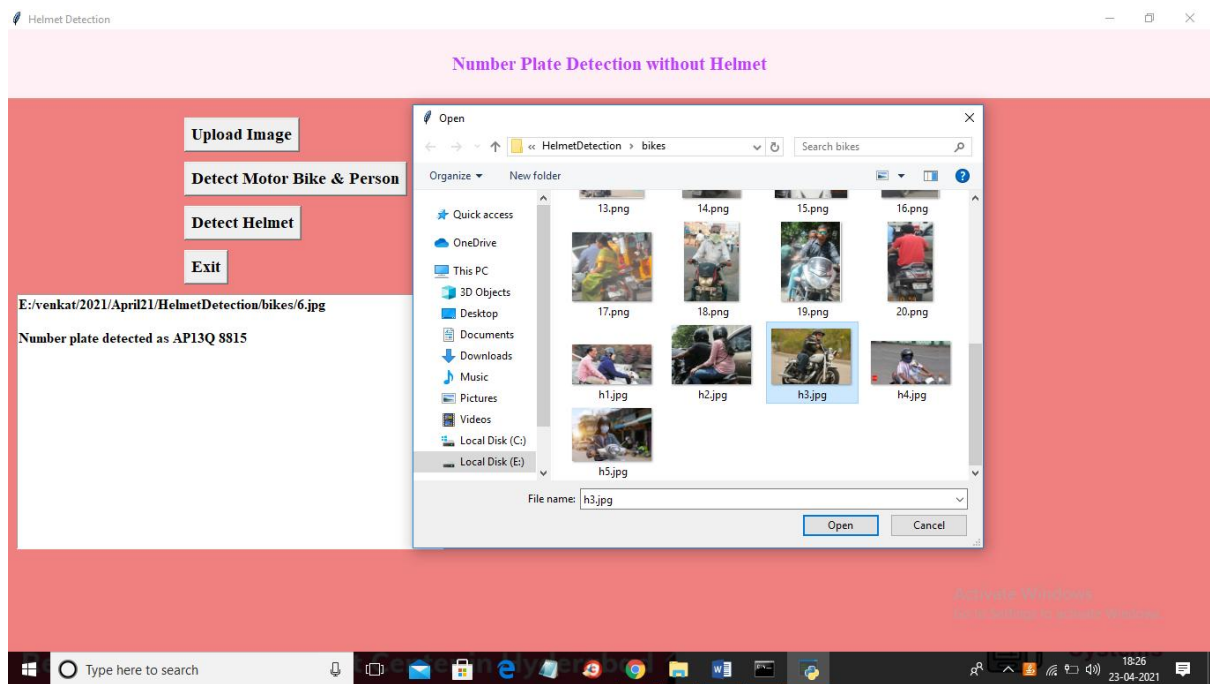


In above screen if person with bike detected then it put bounding box and then click on 'Detect Helmet' button to get below output

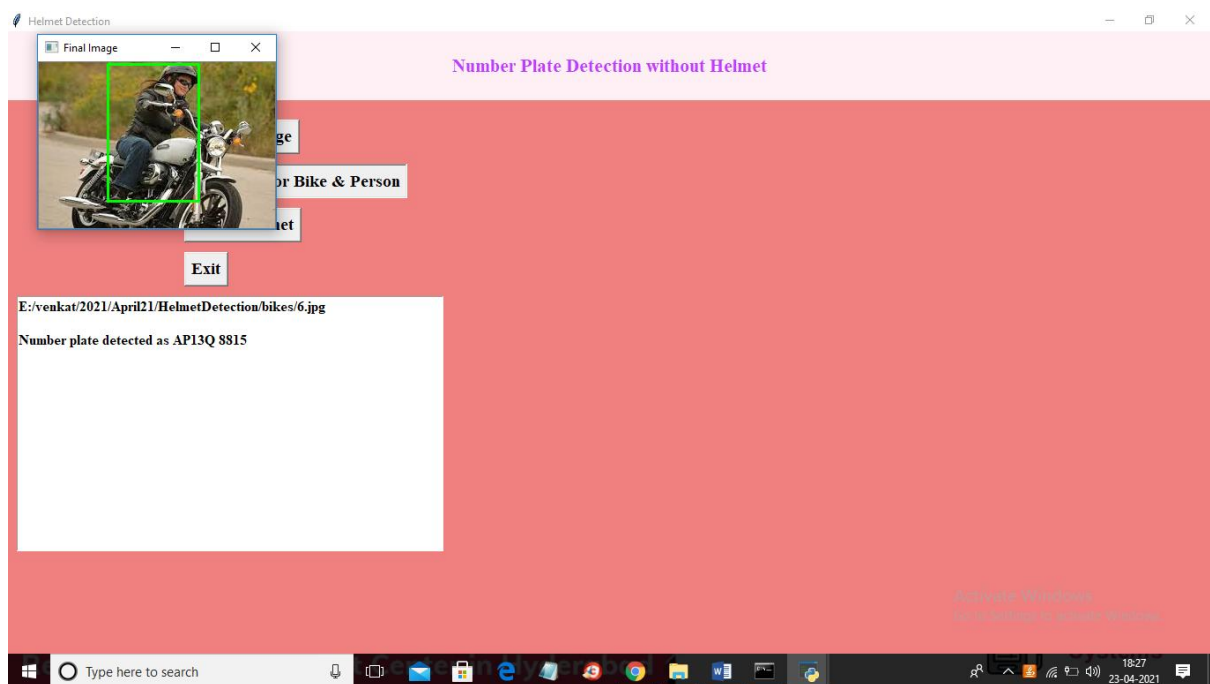


In above screen we can see helmet not detected and then application identify number plate and display on the text area as 'AP13Q 8815'. Now try with other image by uploading it

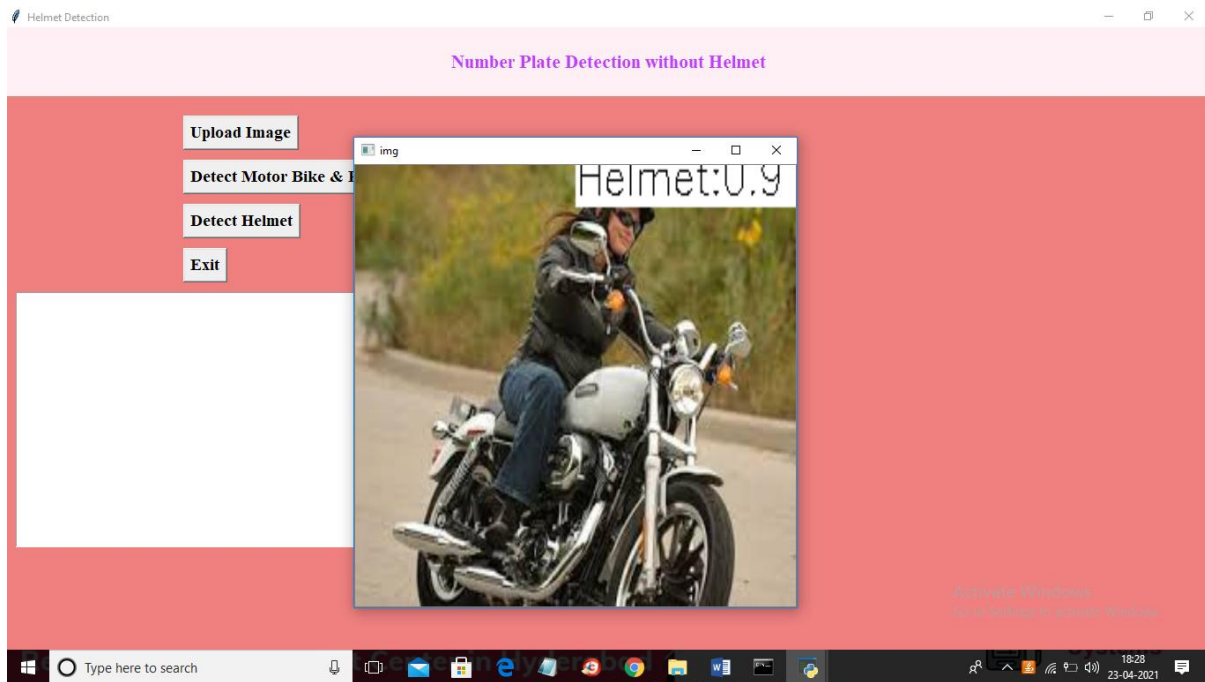




In above screen selecting and uploading 'h3.jpg' file and then click on 'Open' button then click on 'Detect Motor Bike & Person' button to get below result



In above screen person with motor bike detected and now close above image and then click on 'Detect Helmet' button to get below result



In above screen application detected helmet with helmet matching score as 0.90%. Similarly you can upload other images and test.

## 8. CONCLUSIONS:

From the results shown above it is evident that the YOLO object detection is well suited for real-time processing and was able to accurately classify and localize all the object classes. The proposed end-to-end model was developed successfully and has all the capabilities to be automated and deployed for monitoring. For extracting the number plates some techniques are employed by considering different cases such as multiple riders without helmets and designed to handle most of the cases. All the libraries and software used in our project are open source and hence is very flexible and cost efficient. The project was mainly built to solve the problem of non-efficient traffic management. Hence at the end of it we can say that if deployed by any traffic management departments, it would make their job easier and more efficient.

**9.REFERENCES:**

- [1] Viola and Jones, “Robust Real-time Object Detection”, IJCV 2001.
- [2] Navneet Dalal and Bill Triggs, “Histogram of oriented gradients for human detection”.
- [3] Ross, Jeff, Trevor and Jitendra “Rich feature Hierarchy for Accurate object Detection”.
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Fast R-CNN” (Submitted on 4 Jun 2015 (v1), last revised 6 Jan 2016 (this version, v3)).
- [5] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, University of Washington, Allen Institute Of AI.
- [6] Joseph Redmon, Ali Farhadi, “YOLOv3: An Incremental Improvement”, University of Washington, Allen Institute of AI. [6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng – Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector”.
- [7] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust real-time unusual event detection using multiple fixedlocation monitors,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555–560, March 2008.
- [8] AlexeyAB, <https://github.com/AlexeyAB/darknet#requirements>.
- [9] C.-Y. Wen, S.-H. Chiu, J.-J. Liaw, and C.-P. Lu, “The safety helmet detection for atm’s surveillance system via the modified hough transform,” in IEEE 37th Annual International Carnahan Conference on Security Technology., 2003, pp. 364–369.
- [10] C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, “Motorcycle detection and tracking system with occlusion segmentation,” in WIAMIS ’07, USA, 2007.
- [11] A. Hirota, N. H. Tiep, L. Van Khanh, and N. Oka, Classifying Helmeted and Non-helmeted Motorcyclists. Cham: Springer International Publishing, 2017, pp. 81–86.