NUMBER PLATE DETECTION WITHOUT HELMET

A Project Report submitted in partial fulfillment of the degree of the   
Bachelor of Technology in Computer Science and Engineering

By

T.Sai varsh kumar 19C41A05F3

M.Bashwanth 19C41A05H2

D. Sandeep naik 19C41A05G3

Under the Guidance of

Mr B.Naresh



**Department of Computer Science and Engineering**

**Jayamukhi institute of technological Sciences**

**Narsampet, warangal-506 332**

**(Affiliated to JNTUH, Accredited by NAAC ‘A’ Grade)**

**2023**

**Jayamukhi institute of technological Sciences**

**Narsampet, warangal – 506 332**

**(Affiliated to JNTUH, Accredited by NAAC ‘A’ Grade)**

**CERTIFICATE**



This is to certify that the Project Report entitled “**Number plate Detection without helmet**” is a bona fide work of the students  **T.sai varsh kumar(19C41A05F3), M.Bashwanth(19C41A05H2), D.Sandeeep naik(19C41A05G3)** submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering** during the academic year **2022-23.**

**Guide Head of the Department**

**Principal**

**ACKNOWLEDGEMENT**

It gives us immense pleasure in expressing sincere and deep sense of gratitude to **Dr.V.Janaki Principal.** Jayamukhi Institute of Technological sciences for the facility made available for the progress and completion of my main project.

We are extremely thankful to **Dr.P.Srinivas Rao, Head of the Department** Computer Science and Engineering for the permission and encouragement accorded to carry out this major Project work.

We are extremely thankful to **Mr. B. Naresh** Guide Computer Science and Engineering for his enthusiastic and innovative guidance and the permission, encouragement accorded to carry out this project work.

**T.Sai varsh kumar(19C41A05F3)**

**M. Bashwanth(19C41A05H2)**

**D. Sandeep Naik(19C41A05G3)**

**ABSTRACT**

In India, six two-wheeler riders die every hour in road accidents. Also, we have seen that during this pandemic people wear masks, and to avoid congestion they do not wear helmets which attracted our concern and we decided to work on a project where these helmetless people can be penalized for violating traffic rules. To achieve an efficient helmet detection model, we have used the yolov5 object detection model using transfer learning. Further to check whether the biker is wearing a helmet or not we are using two methods, one being checking for overlapping between bounding boxes and the second method is, checking if a helmet exists in the specified range of coordinates above the motorcycle. Our model gives a map of 0.995 and to the best of our knowledge, we used overlapping methods for interlinking objects for finding the person not wearing a helmet. For number plate recognition we are using easyOCR.

**CONTENT**

* **INTRODUCTION**
* **LITERATURE SURVEY**
* **SYSTEM DESIGN**
* SYSTEM ARCHITECTURE
* UML DIAGRAMS
* USE - CASE DIAGRAMS
* CLASS DIAGRAMS
* SEQUENCE DIAGRAMS
* ACTIVITY DIAGRAMS
* DEPLOYMENT DIAGRAMS
* **MODULES USED IN PROJECT**
* **SYSTEM REQUIREMENTS**
* HARDWARE REQUIREMENTS
* SOFTWARE REQUIREMENTS
* **SYSTEM STUDY**
* TECHNICAL FEASIBILITY
* OPERATIONAL FEASIBILITY
* ECONOMIC FEASIBILITY
* **SOFTWARE ENVIRONMENT**
* **SYSTEM TEST**
* **RESULTS**
* **CONCLUSION**
* **REFERENCES**

**INTRODUCTION**

We come across numerous incidents where motorcyclists get severely injured simply because they didn’t wear safety helmets. This negligence has caused a threat to many innocent lives. In India, six two-wheeler riders die every hour in road accidents. Despite having strict traffic regulations, people still neglect the importance of wearing a helmet. Also, the existing surveillance system requires significant human assistance and humans are prone to make mistakes. So automating this system is highly desirable. For doing so, we used object detection deep learning algorithms like YOLOv5.

We are dealing with a variety of motorcyclists with distinct colors of clothes, helmets, and angles of motorcyclists. To achieve this, we require a deep neural network that will help to determine the motorcyclist very accurately. We mainly aim to collect a database of all the motorcyclists who have violated the rules. One of the key problems we faced was determining whether the person is wearing a helmet or not and to differentiate between biker and pedestrian. To solve this problem, we have come up with two methods for finding the same. In the first method, we check the overlap between the classes and create a link between the bike rider and the number plate. In the second method, depending upon the height of the motorcycle bounding box, it will check if a helmet class exists at a particular distance above the motorcycle. In conclusion, the system will determine whether the motorcyclist is wearing a helmet or not. And depending upon the answer, it would extract the number plates of those motorcyclists who are not wearing helmets.

**LITERATURE SURVEY**

**Title:** “Helmet presence classification with motorcycle detection and tracking,”

## Abstract: Helmets are essential for the safety of a motorcycle rider, however, the enforcement of helmet wearing is a time-consuming labour intensive task. A system for the automatic classification and tracking of motorcycle riders with and without helmets is therefore described and tested. The system uses support vector machines trained on histograms derived from head region image data of motorcycle riders using both static photographs and individual image frames from video data. The trained classifier is incorporated into a tracking system where motorcycle riders are automatically segmented from video data using background subtraction. The heads of the riders are isolated and then classified using the trained classifier. Each motorcycle rider results in a sequence of regions in adjacent time frames called tracks. These tracks are then classified as a whole using a mean of the individual classifier results. Tests show that the classifier is able to accurately classify whether riders are wearing helmets or not on static photographs. Tests on the tracking system also demonstrate the validity and usefulness of the classification approach.

**Title:** “Vehicle detection, tracking and classification in urban traffic,”

**Abstract:** This paper presents a system for vehicle detection, tracking and classification from roadside CCTV. The system counts vehicles and separates them into four categories: car, van, bus and motorcycle (including bicycles). A new background Gaussian Mixture Model (GMM) and shadow removal method have been used to deal with sudden illumination changes and camera vibration. A Kalman filter tracks a vehicle to enable classification by majority voting over several consecutive frames, and a level set method has been used to refine the foreground blob. Extensive experiments with real world data have been undertaken to evaluate system performance. The best performance results from training a SVM (Support Vector Machine) using a combination of a vehicle silhouette and intensity-based pyramid HOG features extracted following background subtraction, classifying foreground blobs with majority voting. The evaluation results from the videos are encouraging: for a detection rate of 96.39%, the false positive rate is only 1.36% and false negative rate 4.97%. Even including challenging weather conditions, classification accuracy is 94.69%.

**Title:**  “Automatic Number Plate Recognition System”,

**Abstract:** Automatic Number Plate Recognition (ANPR) is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or area around top government offices e.g. Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle’s owner, place of registration, address, etc. The system is implemented and simulated in Matlab, and it performance is tested on real image. It is observed from the experiment that the developed system successfully detects and recognize the vehicle number plate on real images.

**Title:** “Faster R-CNN”

**Abstract:** This article gives a review of the Faster R-CNN model developed by a group of researchers at Microsoft. Faster R-CNN is a deep convolutional network used for object detection, that appears to the user as a single, end-to-end, unified network. The network can accurately and quickly predict the locations of different objects. In order to truly understand Faster R-CNN, we must also do a quick overview of the networks that it evolved from, namely R-CNN and Fast R-CNN.

The article starts by quickly reviewing the region-based CNN (**R-CNN**), which is the first trial towards building an object detection model that extracts features using a pre-trained CNN. Next, **Fast R-CNN** is quickly reviewed, which is faster than the **R-CNN** but unfortunately neglects how the region proposals are generated. This is later solved by the **Faster R-CNN**, which builds a region-proposal network that can generate region proposals that are fed to the detection model (Fast R-CNN) to inspect for objects.

**Title:** “Automatic detection of bike-riders without helmet using surveillance videos in real-time,”

**Abstract:** In this paper, we propose an approach for automatic detection of bike-riders without helmet using surveillance videos in real time. The proposed approach first detects bike riders from surveillance video using background subtraction and object segmentation. Then it determines whether bike-rider is using a helmet or not using visual features and binary classifier. Also, we present a consolidation approach for violation reporting which helps in improving reliability of the proposed approach. In order to evaluate our approach, we have provided a performance comparison of three widely used feature representations namely histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and local binary patterns (LBP) for classification. The experimental results show detection accuracy of 93.80% on the real world surveillance data. It has also been shown that proposed approach is computationally less expensive and performs in real-time with a processing time of 11.58 ms per frame.

**Title:**  “Helmet Detection Based On Improved YOLO V3 Deep Model”

**Abstract:** Helmet wearing is very important to the safety of workers at construction sites and factories. How to warn/identify/certify workers “whether or not the helmet is worn” is often a difficult point for enterprises to monitor. Based on the YOLO V3 full-regression deep neural network architecture, this paper utilizes the advantage of Densenet in model parameters and technical cost to replace the backbone of the YOLO V3 network for feature extraction, thus forming the so-called YOLO-Densebackbone convolutional neural network. The test results show that the improved model can effectively deal with situations that the helmet is stained, partially occluded, or there are many targets with a low image resolution. In the test set, compared with the traditional YOLO V3, the improved algorithm detection accuracy increased by 2.44% with the same detection rate. The establishment of this model has important practical significance for improving helmet detection and ensuring safe construction.

**Title:**  "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks",

**Abstract:** Visual examination of the workplace and in-time reminder to the failure of wearing a safety helmet is of particular importance to avoid injuries of workers at the construction site. Video monitoring systems provide a large amount of unstructured image data on-site for this purpose, however, requiring a computer vision-based automatic solution for real-time detection. Although a growing body of literature has developed many deep learning-based models to detect helmet for the traffic surveillance aspect, an appropriate solution for the industry application is less discussed in view of the complex scene on the construction site. In this regard, we develop a deep learning-based method for the real-time detection of a safety helmet at the construction site. The presented method uses the SSD-MobileNet algorithm that is based on convolutional neural networks. A dataset containing 3261 images of safety helmets collected from two sources, i.e., manual capture from the video monitoring system at the workplace and open images obtained using web crawler technology, is established and released to the public. The image set is divided into a training set, validation set, and test set, with a sampling ratio of nearly 8 : 1 : 1. The experiment results demonstrate that the presented deep learning-based model using the SSD-MobileNet algorithm is capable of detecting the unsafe operation of failure of wearing a helmet at the construction site, with satisfactory accuracy and efficiency.

**Title:** “Real-time number plate detection for non-helmeted motorcyclist using YOLO”,

**Abstract:** Nowadays, detection of license plate (LP) for non-helmeted motorcyclist has become mandatory to ensure the safety of the motorcyclists. This paper presents the real-time detection of LP for non-helmeted motorcyclist using the real-time object detector YOLO (You Only Look Once). In this proposed approach, a single convolutional neural network was deployed to automatically detect the LP of a non-helmeted motorcyclist from the video stream. The centroid tracking method with a horizontal reference line was used to eliminate the false positive generated by the helmeted motorcyclist as they leave the video frames. The overall LP detection rate was 98.52%.

**Title:** “Automatic Helmet Violation Detection of Motorcyclists from Surveillance Videos using Deep Learning Approaches of Computer Vision”

**Abstract:** Automatic detection of helmet for motorcyclists from real-time surveillance videos is a rising application in computer science. Object detection and classification using deep learning was recently well-known over the years. Researchers used these techniques for solving several surveillance-related problems. Several deep learning models adopt for automatic detection of helmet for motorcyclists but they cannot achieve state-of-the-art results due to different difficulties such as low resolution, whether conditions, occlusion and illumination etc. In this paper, we proposed a methodology for surveillance videos that automatically detect the helmet wear by motorcyclist or not. For this purpose, we used the Faster R-CNN model. First, we apply Region Proposal Network (RPN) starting with the input image that has been delivered into the backbone. Then RPN weights are settled and proposals from RPN are utilized to train the Faster RCNN model. For training, we used the self-generated dataset of three different locations in Lahore, Pakistan. The experimental results detect 97.26% accuracy on real-time surveillance videos for the detection of helmet for motorcyclists.

**Title:** “Single Line License Plate Detection Using OpenCV And Tesseract”

**Abstract:** – License plate detection is an image processing technology that uses a license (number) plate for vehicle identification. The objective is to design and implement an efficient vehicle identification system that identifies the vehicle using the vehicle’s license plate. The system can be implemented on the entrance of parking lots, toll booths, or any private premises like college, etc. to keep the records of ongoing and outgoing vehicles. It can be used to allow access to only permitted vehicles inside the premises. The developed system first captures the image of the vehicle’s front, then detects the license plate and then reads the license plate. The vehicle license plate is extracted using the image processing of the image. Optical character recognition (OCR) is used for character recognition. The system is implemented using OpenCV and its performance is tested on various images. It is observed that the developed system successfully detects and recognizes the vehicle license plate

**PROPOSED SYSTEM**

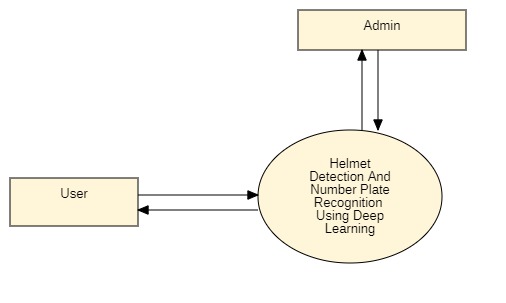
We are dealing with a variety of motorcyclists with distinct colors of clothes, helmets, and angles of motorcyclists. To achieve this, we require a deep neural network that will help to determine the motorcyclist very accurately. We mainly aim to collect a database of all the motorcyclists who have violated the rules. One of the key problems we faced was determining whether the person is wearing a helmet or not and to differentiate between biker and pedestrian. To solve this problem, we have come up with two methods for finding the same. In the first method, we check the overlap between the classes and create a link between the bike rider and the number plate. In the second method, depending upon the height of the motorcycle bounding box, it will check if a helmet class exists at a particular distance above the motorcycle.

In conclusion, the system will determine whether the motorcyclist is wearing a helmet or not. And depending upon the answer, it would extract the number plates of those motorcyclists who are not wearing helmets.

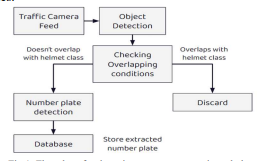
**SYSTEM DESIGN**

**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



**System Architechure**



**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

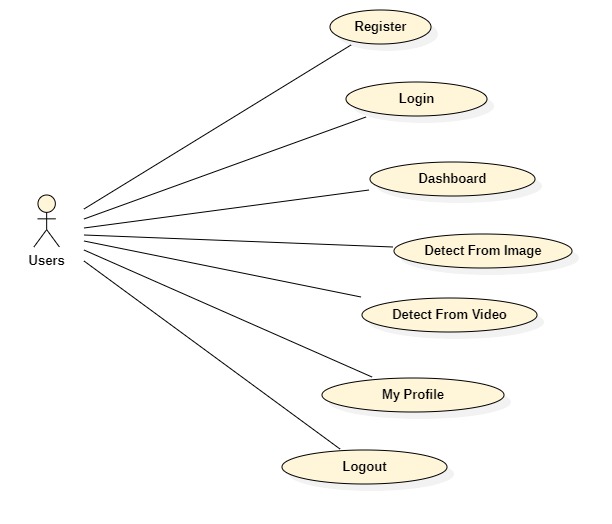
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

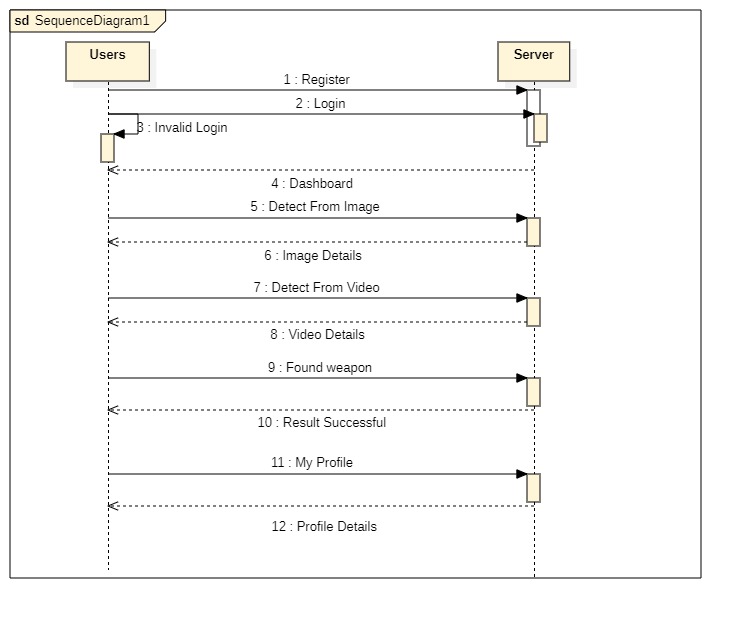


**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

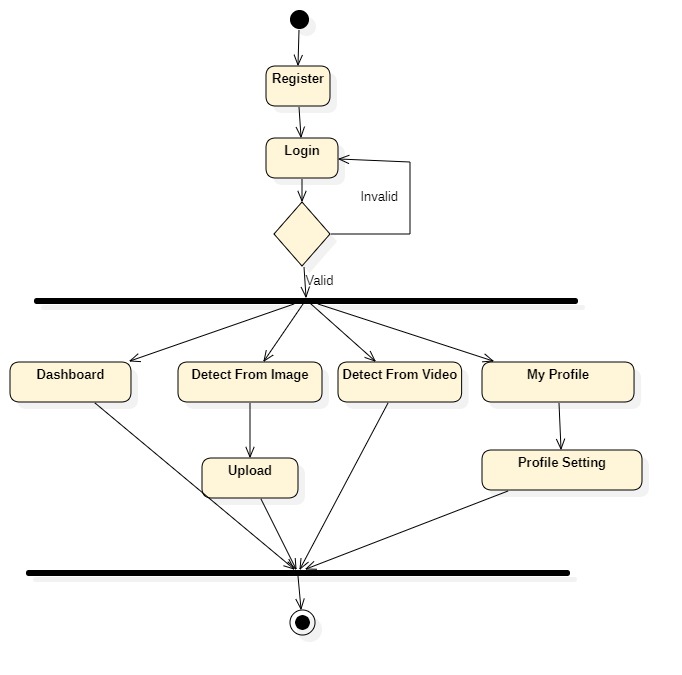
**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) 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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.0



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**DEPLOYMENT DIAGRAM:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

The deployment diagram maps the software architecture created in design to the physical system architecture that executes it. In distributed systems, it models the distribution of the software across the physical nodes.

The software systems are manifested using various artifacts, and then they are mapped to the execution environment that is going to execute the software such as nodes. Many nodes are involved in the deployment diagram; hence, the relation between them is represented using communication paths.

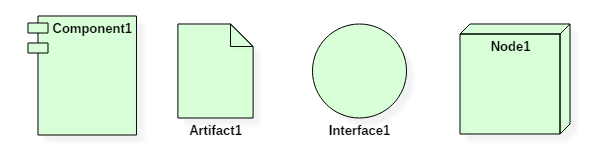
**There are two forms of a deployment diagram.**

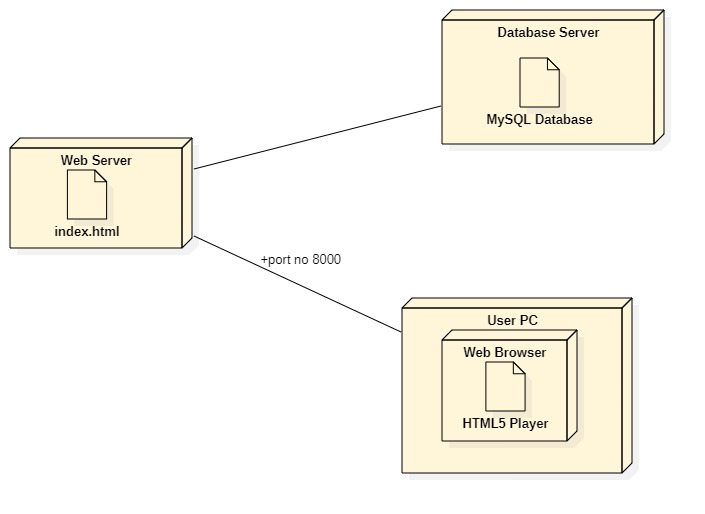
* Descriptor form
* It contains nodes, the relationship between nodes and artifacts.
* Instance form
* It contains node instance, the relationship between node instances and artifact instance.
* An underlined name represents node instances.

**Purpose of a deployment diagram**

Deployment diagrams are used with the sole purpose of describing how software is deployed into the hardware system. It visualizes how software interacts with the hardware to execute the complete functionality. It is used to describe software to hardware interaction and vice versa.

**Deployment Diagram Symbol and notations**

Deployment Diagram Notations



**Modules Used in Project**

1. **Helmet Detection using Rekognition**

Amazon Rekognition, which is a computer vision service provided by AWS, uses a combination of deep learning and machine learning algorithms to analyze images and videos. Specifically, it uses convolutional neural networks (CNNs) to recognize objects, faces, and scenes in images and videos, and to perform various types of analysis such as facial analysis, object and scene detection, and image moderation.

Rekognition also employs other machine learning techniques, such as unsupervised learning, to automatically learn patterns and identify correlations in the data it processes. Additionally, Rekognition provides a range of pre-built models and APIs for common use cases, such as face recognition and celebrity recognition, that are trained using these algorithms.

Amazon Rekognition, the cloud-based computer vision service provided by Amazon Web Services (AWS), can detect weapons by analyzing images and videos to identify objects and scenes within them. Rekognition uses deep learning algorithms and machine learning models to recognize objects and scenes in images, including helmets. It can identify a wide range of helmets in near real-time. However, the accuracy of helmet detection depends on the quality and resolution of the image or video, as well as the context and position of the helmet within the frame

1. **Number plate Recognition using EasyOCR**

EasyOCR is an open-source Python package that is used to perform Optical Character Recognition- OCR (extract text from images).

The package is very easy to use and at the time of this writing, it supports 80+ languages including Chinese, Arabic, French, English, Cyrillic, etc.

In license plate recognition, not only we want to find the bounding boxes of the license plate, but we also want to extract the text from the detected license plate.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| MINIMUM (Required for Execution) | | MY SYSTEM (Development) |
| System | Pentium IV 2.2 GHz | i3 Processor 5th Gen |
| Hard Disk | 20 Gb | 500 Gb |
| Ram | 1 Gb | 4 Gb |

**SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Operating System | Windows 10/11 |
| Development Software | Python 3.10 |
| Programming Language | Python |
| Domain | Machine Learning |
| Integrated Development Environment (IDE) | Visual Studio Code |
| Front End Technologies | HTML5, CSS3, Java Script |
| Back End Technologies or Framework | Django |
| Database Language | SQL |
| Database (RDBMS) | MySQL |
| Database Software | WAMP or XAMPP Server |
| Web Server or Deployment Server | Django Application Development Server |

**SYSTEM STUDY**

**INTRODUCTION**

A feasibility study assesses the operational, technical and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management-oriented activity. The objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

Projects are initiated for two broad reasons:

1. Problems that lend themselves to systems solutions

2. Opportunities for improving through:

(a) upgrading systems

(b) altering systems

(c) installing new systems

A feasibility study should provide management with enough information to decide:

* Whether the project can be done
* Whether the final product will benefit its intended users and organization
* What are the alternatives among which a solution will be chosen
* Is there a preferred alternative?

**TECHNICAL FEASIBILITY**

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

The essential questions that help in testing the operational feasibility of a system include the following:

* Is the project feasible within the limits of current technology?
* Does the technology exist at all?
* Is it available within given resource constraints?
* Is it a practical proposition?
* Manpower- programmers, testers & debuggers
* Software and hardware
* Are the current technical resources sufficient for the new system?
* Can they be upgraded to provide to provide the level of technology necessary for the new system?
* Do we possess the necessary technical expertise, and is the schedule reasonable?
* Can the technology be easily applied to current problems?
* Does the technology have the capacity to handle the solution?
* Do we currently possess the necessary technology?

**OPERATIONAL FEASIBILITY**

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a 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.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The essential questions that help in testing the operational feasibility of a system include the following:

* Does current mode of operation provide adequate throughput and response time?
* Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
* Does current mode of operation provide cost-effective information services to the business?
* Could there be a reduction in cost and or an increase in benefits?
* Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?
* Does current mode of operation make maximum use of available resources, including people, time, and flow of forms?
* Does current mode of operation provide reliable services
* Are the services flexible and expandable?
* Are the current work practices and procedures adequate to support the new system?
* If the system is developed, will it be used?
* Manpower problems
* Labour objections
* Manager resistance
* Organizational conflicts and policies
* Social acceptability
* Government regulations
* Does management support the project?
* Are the users not happy with current business practices?
* Will it reduce the time (operation) considerably?
* Have the users been involved in the planning and development of the project?
* Will the proposed system really benefit the organization?
* Does the overall response increase?
* Will accessibility of information be lost?
* Will the system affect the customers in considerable way?
* Legal aspects
* How do the end-users feel about their role in the new system?
* What end-users or managers may resist or not use the system?
* How will the working environment of the end-user change?
* Can or will end-users and management adapt to the change?

**ECONOMIC FEASIBILITY**

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

* Is the system cost effective?
* Do benefits outweigh costs?
* The cost of doing full system study
* The cost of business employee time
* Estimated cost of hardware
* Estimated cost of software/software development
* Is the project possible, given the resource constraints?
* What are the savings that will result from the system?
* Cost of employees' time for study
* Cost of packaged software/software development
* Selection among alternative financing arrangements (rent/lease/purchase)

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system.

The economic feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization.

**SOFTWARE ENVIRONMENT**

**What is Python programming language?**

Python is a **high-level, general-purpose, interpreted**programming language.

**1) High-level**

Python is a high-level programming language that makes it easy to learn. Python doesn’t require you to understand the details of the computer in order to develop programs efficiently.

**2) General-purpose**

domains including: Python is a general-purpose language. It means that you can use Python in various

* Web applications
* Big data applications
* Testing
* Automation
* Data science, machine learning, and AI
* Desktop software
* Mobile apps

The targeted language like SQL which can be used for querying data from relational databases.

**3) Interpreted**

Python is an interpreted language. To develop a Python program, you write Python code into a file called source code.

To execute the source code, you need to convert it to the machine language that the computer can understand. And the Python **interpreter**turns the source code, line by line, once at a time, into the machine code when the Python program executes.

Compiled languages like Java and C# use a **compiler**that compiles the whole source code before the program executes.

**Why Python**

Python increases your productivity. Python allows you to solve complex problems in less time and fewer lines of code. It’s quick to make a prototype in Python.

Python becomes a solution in many areas across industries, from web applications to data science and machine learning.

Python is quite easy to learn in comparison with other programming languages. Python syntax is clear and beautiful.

Python has a large ecosystem that includes lots of libraries and frameworks.

Python is cross-platform. Python programs can run on Windows, Linux, and macOS.

Python has a huge community. Whenever you get stuck, you can get help from an active community.

Python developers are in high demand.

**History of Python**

* Python was created by Guido Van Rossum.
* The design began in the late 1980s and was first released in February 1991.

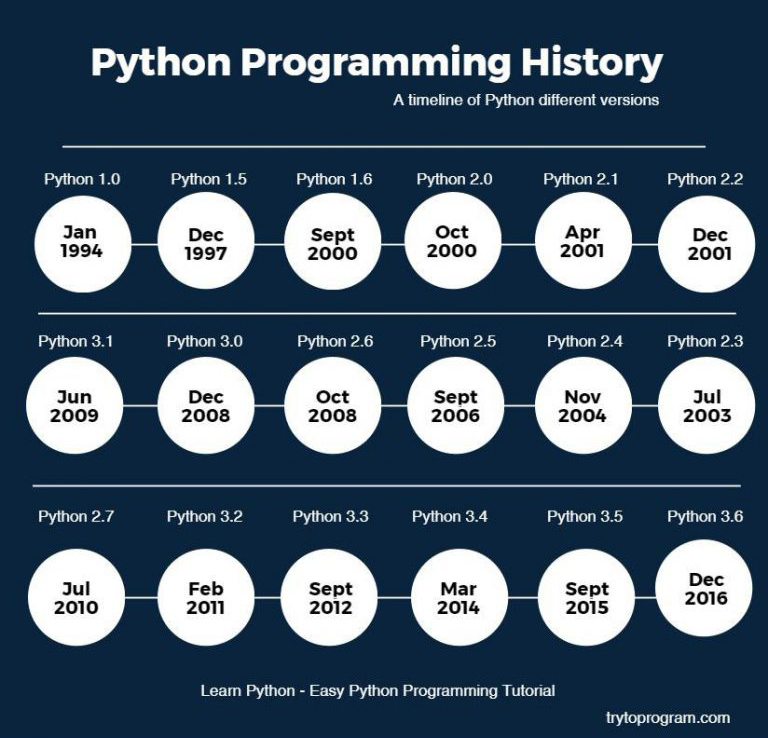
**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late 70s. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Python Version History**

Internal releases – 1990 Implementation

started - December 1989

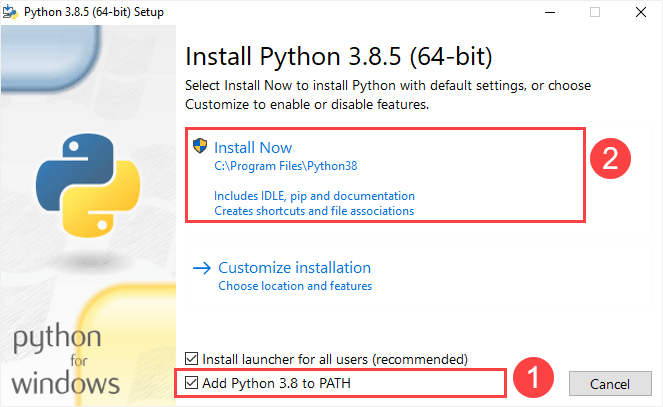


**Install Python on Windows**

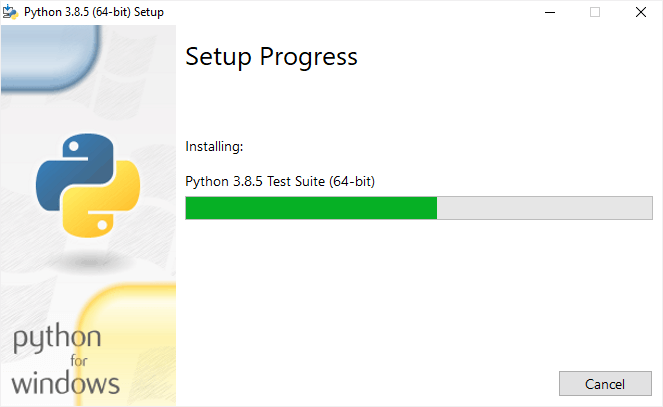
First, [download the latest version of Python](https://www.python.org/downloads/) from the download page.

Second, double-click the installer file to launch the setup wizard.

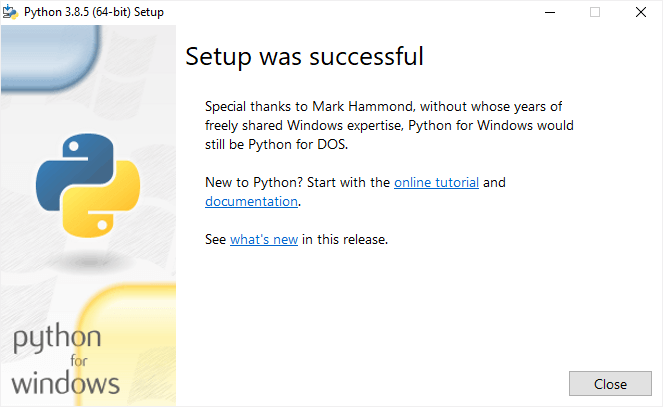
In the setup window, you need to check the **Add Python 3.8 to PATH**and click Install Now to begin the installation.



It’ll take a few minutes to complete the setup.

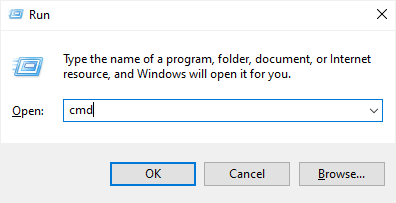


Once the setup completes, you’ll see the following window:

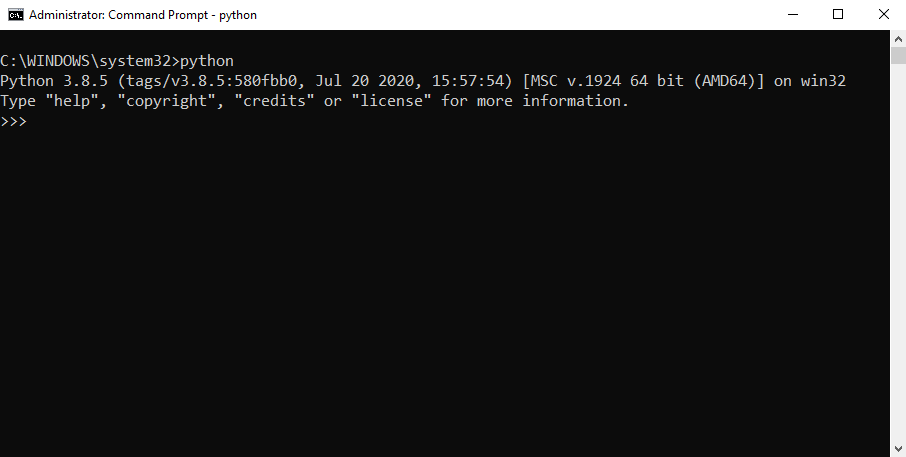


**Verify the installation**

To verify the installation, you open the Run window and type cmd and press Enter:



In the Command Prompt, type python command as follows:



If you see the output like the above screenshot, you’ve successfully installed Python on your computer.

To exit the program, you type Ctrl-Z and press Enter.

If you see the following output from the Command Prompt after typing the python command:

'python' is not recognized as an internal or external command,

operable program or batch file.

Likely, you didn’t check the **Add Python 3.8 to PATH** checkbox when you install Python.

**Install Python on macOS**

an official installer. Here are the steps: It’s recommended to install Python on macOS using

* First, [download a Python release for macOS](https://www.python.org/downloads/macos/).
* Second, run the installer by double-clicking the installer file.
* Third, follow the instruction on the screen and click the Next button until the installer completes.

**Install Python on Linux**

Before installing Python 3 on your Linux distribution, you check whether Python 3 was already installed by running the following command from the terminal:

python3 --version

If you see a response with the version of Python, then your computer already has Python 3 installed. Otherwise, you can install Python 3 using a package management system.

For example, you can install Python 3.10 on Ubuntu using apt:

sudo apt install python3.10

To install the newer version, you replace 3.10 with that version.

A quick introduction to the Visual Studio Code

Visual Studio Code is a lightweight source code editor. The Visual Studio Code is often called VS Code. The VS Code runs on your desktop. It’s available for Windows, macOS, and Linux.

VS Code comes with many features such as IntelliSense, code editing, and extensions that allow you to edit Python source code effectively. The best part is that the VS Code is open-source and free.

Besides the desktop version, [VS Code also has a browser version](https://vscode.dev/) that you can use directly in your web browser without installing it.

This tutorial teaches you how to set up Visual Studio Code for a Python environment so that you can edit, run, and debug Python code.

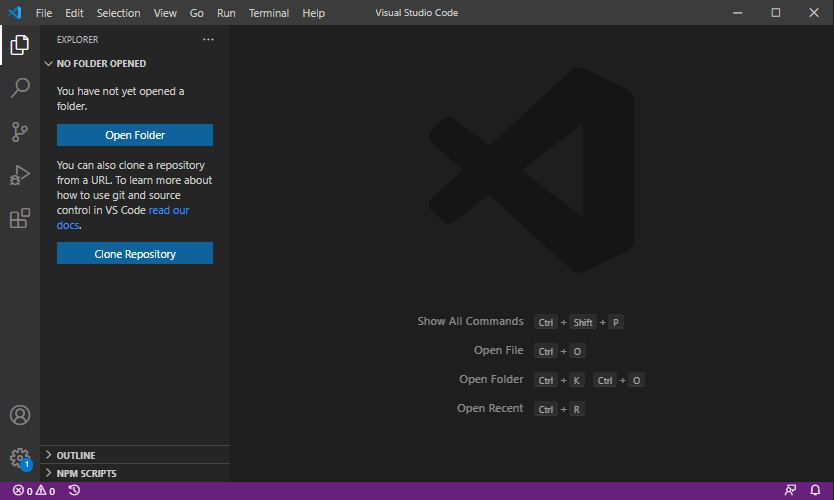
Setting up Visual Studio Code

To set up the VS Code, you follow these steps:

First, navigate to the [VS Code official](https://code.visualstudio.com/) website and download the VS code based on your platform (Windows, macOS, or Linux).

Second, launch the setup wizard and follow the steps.

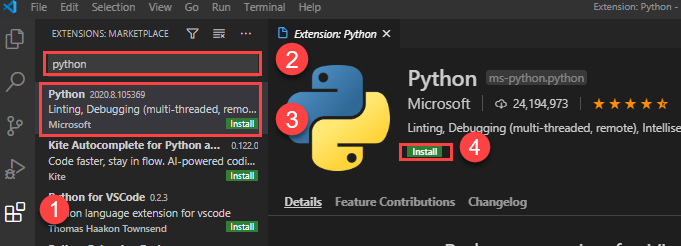
Once the installation completes, you can launch the VS code application:



**Install Python Extension**

To make the VS Code works with Python, you need to install the Python extension from the Visual Studio Marketplace.

The following picture illustrates the steps:



* First, click the **Extensions** tab.
* Second, type the python extension pack keyword on the search input.
* Third, click the Python extension pack. It’ll show detailed information on the right pane.
* Finally, click the **Install** button to install the Python extension.

Now, you’re ready to develop the first program in Python.

Creating a new Python project

First, create a new folder called helloworld.

Second, launch the VS code and open the helloworld folder.

Third, create a new app.py file and enter the following code and save the file:

print('Hello, World!')

Code language: Python (python)

The print() is a built-in function that displays a message on the screen. In this example, it’ll show the message 'Hello, Word!'.

What is a function

When you sum two numbers, that’s a function. And when you multiply two numbers, that’s also a function.

Each function takes your inputs, applies some rules, and returns a result.

In the above example, the print() is a function. It accepts a string and shows it on the screen.

Python has many built-in functions like the print() function to use them out of the box in your program.

In addition, Python allows you to define your functions, which you’ll learn how to do it later.

Executing the Python Hello World program

To execute the app.py file, you first launch the Command Prompt on Windows or Terminal on macOS or Linux.

Then, navigate to the helloworld folder.

After that, type the following command to execute the app.py file:

python app.py

Code language: Python (python)

If you use macOS or Linux, you use python3 command instead:

python3 app.py

Code language: CSS (css)

If everything is fine, you’ll see the following message on the screen:

Hello, World!

Code language: Python (python)

If you use VS Code, you can also launch the Terminal within the VS code by:

* Accessing the menu **Terminal > New Terminal**
* Or using the keyboard shortcut Ctrl+Shift+`.

Typically, the backtick key (`) locates under the Esc key on the keyboard.

Python IDLE

Python IDLE is the Python Integration Development Environment (IDE) that comes with the Python distribution by default.

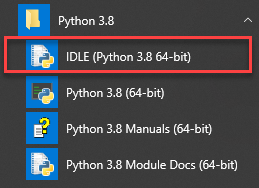
The Python IDLE is also known as an interactive interpreter. It has many features such as:

* Code editing with syntax highlighting
* Smart indenting
* And auto-completion

In short, the Python IDLE helps you experiment with Python quickly in a trial-and-error manner.

The following shows you step by step how to launch the Python IDLE and use it to execute the Python code:

First, launch the Python IDLE program:



A new Python Shell window will display as follows:



Now, you can enter the Python code after the cursor >>> and press Enter to execute it.

For example, you can type the code print('Hello, World!') and press Enter, you’ll see the message Hello, World! immediately on the screen:



**Python Syntax**

## Whitespace and indentation

If you’ve been working in other programming languages such as Java, C#, or C/C++, you know that these languages use semicolons (;) to separate the statements.

However, Python uses whitespace and indentation to construct the code structure.

The following shows a snippet of Python code:

*# define main function to print out something*

def main():

i = 1

max = 10

while (i < max):

print(i)

i = i + 1

*# call function main*

main()

The meaning of the code isn’t important to you now. Please pay attention to the code structure instead.

At the end of each line, you don’t see any semicolon to terminate the statement. And the code uses indentation to format the code.

By using indentation and whitespace to organize the code, Python code gains the following advantages:

* First, you’ll never miss the beginning or ending code of a block like in other programming languages such as Java or C#.
* Second, the coding style is essentially uniform. If you have to maintain another developer’s code, that code looks the same as yours.
* Third, the code is more readable and clearer in comparison with other programming languages.

## Comments

The comments are as important as the code because they describe why a piece of code was written.

When the Python interpreter executes the code, it ignores the comments.

In Python, a single-line comment begins with a hash (#) symbol followed by the comment. For example:

*# This is a single line comment in Python*

## Continuation of statements

Python uses a newline character to separate statements. It places each statement on one line.

However, a long statement can span multiple lines by using the backslash (\) character.

The following example illustrates how to use the backslash (\) character to continue a statement in the second line:

if (a == True) and (b == False) and \

(c == True):

print("Continuation of statements")

## Identifiers

Identifiers are names that identify variables, functions, modules, classes, and other objects in Python.

The name of an identifier needs to begin with a letter or underscore (\_). The following characters can be alphanumeric or underscore.

Python identifiers are case-sensitive. For example, the counter and Counter are different identifiers.

In addition, you cannot use Python keywords for naming identifiers.

## Keywords

Some words have special meanings in Python. They are called keywords.

The following shows the list of keywords in Python:

False class finally is return

None continue for lambda try

True def from nonlocal while

and del global not with

as elif if or yield

assert else import pass

break except in raise

Python is a growing and evolving language. So, its keywords will keep increasing and changing.

Python provides a special module for listing its keywords called keyword.

To find the current keyword list, you use the following code:

import keyword

print(keyword.kwlist)

## String literals

Python uses single quotes ('), double quotes ("), triple single quotes (''') and triple-double quotes (""") to denote a string literal.

The string literal need to be surrounded with the same type of quotes. For example, if you use a single quote to start a string literal, you need to use the same single quote to end it.

The following shows some examples of string literals:

s = 'This is a string'

print(s)

s = "Another string using double quotes"

print(s)

s = ''' string can span

multiple line '''

print(s)

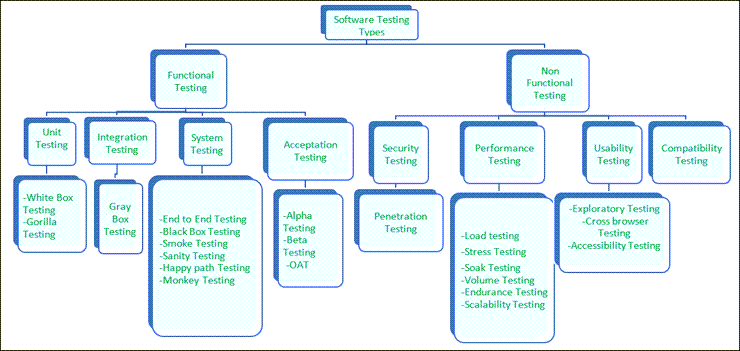
**SYSTEM TESTING**

**Types of Software Testing: Different Testing Types with Details**

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub-types, etc.

Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

**Different Types of Software Testing**



**Functional Testing**

There are four main types of functional testing.

**1) Unit Testing**

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution.

Unit testing is important because we can find more defects at the unit test level.

**For example,** there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

**a) White Box Testing**

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

**b) Gorilla Testing**

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

**For example,** the tester is testing the pet insurance company’s website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let’s say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

**2) Integration Testing**

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems. **For example,** a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

**a) Gray box testing**

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

**3) System Testing**

System testing is types of testing where tester evaluates the whole system against the specified requirements.

**a) End to End Testing**

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate

**For example** ,a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users’ accounts, updating user address information, receiving order confirmation emails and policy documents.

**b) Black Box Testing**

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

**c) Smoke Testing**

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists. The testing team will ensure that the build is stable, and a detailed level of testing will be carried out further.

**For example,** tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

**d) Sanity Testing**

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

**For example,** a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

**e) Happy path Testing**

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

**f) Monkey Testing**

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

The objective of Monkey Testing is to check if an application or system gets crashed by providing random input values/data. Monkey Testing is performed randomly, no test cases are scripted, and it is not necessary to be aware  
of the full functionality of the system.

**4) Acceptance Testing**

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

**a) Alpha Testing**

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

**For example,** the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

**b) Beta Testing**

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment**before releasing the product to the market for the actual end-users.

Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

**c) Operational acceptance testing (OAT)**

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

**The focus of the OAT is on the following points:**

* Testing of backup and restore.
* Installing, uninstalling, upgrading software.
* The recovery process in case of natural disaster.
* User management.
* Maintenance of the software.

**Non-Functional Testing**

There are four main types of functional testing.

**1) Security Testing**

penetrate the system. It is a type of testing performed by a special team. Any hacking method can

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.

It also checks how software behaves for any hacker’s attack & malicious programs and how software is maintained for data security after such a hacker attack.

**a) Penetration Testing**

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.

Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

**Notes:** Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

**2) Performance Testing**

Performance testing is testing of an application’s stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner, etc. are good tools available in the market.

**a) Load testing**

Load testing is testing of an application’s stability and response time by applying load, which is equal to or less than the designed number of users for an application.

**For example,** your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

**b) Stress Testing**

Stress testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

**For example,** your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

**c) Scalability Testing**

Scalability testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

**For example,** your application handles 1000 users at a time with a response time of 2 seconds, then scalability testing can be done by applying a load of more than 1000 users and gradually increasing the number of users to find out where exactly my application is crashing.

Let’s say my application is giving response time as follows:

* 1000 users -2 sec
* 1400 users -2 sec
* 4000 users -3 sec
* 5000 users -45 sec
* 5150 users- crash – This is the point that needs to identify in scalability testing

**d) Volume testing (flood testing)**

Volume testing is testing an application’s stability and response time by transferring a large volume of data to the database. Basically, it tests the capacity of the database to handle the data.

**e) Endurance Testing (Soak Testing)**

Endurance testing is testing an application’s stability and response time by applying load continuously for a longer period to verify that the application is working fine.

**For example,** car companies soak testing to verify that users can drive cars continuously for hours without any problem.

**3) Usability Testing**

Usability testing is testing an application from the user’s perspective to check the look and feel and user-friendliness.

**For example,** there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background colour of the app should be black and price of and stock is displayed in red or green colour.

The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

**a) Exploratory testing**

Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and look for defects that exist in the application. Testers use the knowledge of the business domain to test the application. Test charters are used to guide the exploratory testing.

**b) Cross browser testing**

Cross browser testing is testing an application on different browsers, operating systems, mobile devices to see look and feel and performance.

Why do we need cross-browser testing? The answer is different users use different operating systems, different browsers, and different mobile devices. The goal of the company is to get a good user experience regardless of those devices.

Browser stack provides all the versions of all the browsers and all mobile devices to test the application. For learning purposes, it is good to take the free trial given by browser stack for a few days.

**c) Accessibility Testing**

The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.

Here, disability means deafness, colour blindness, mentally disabled, blind, old age, and other disabled groups. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

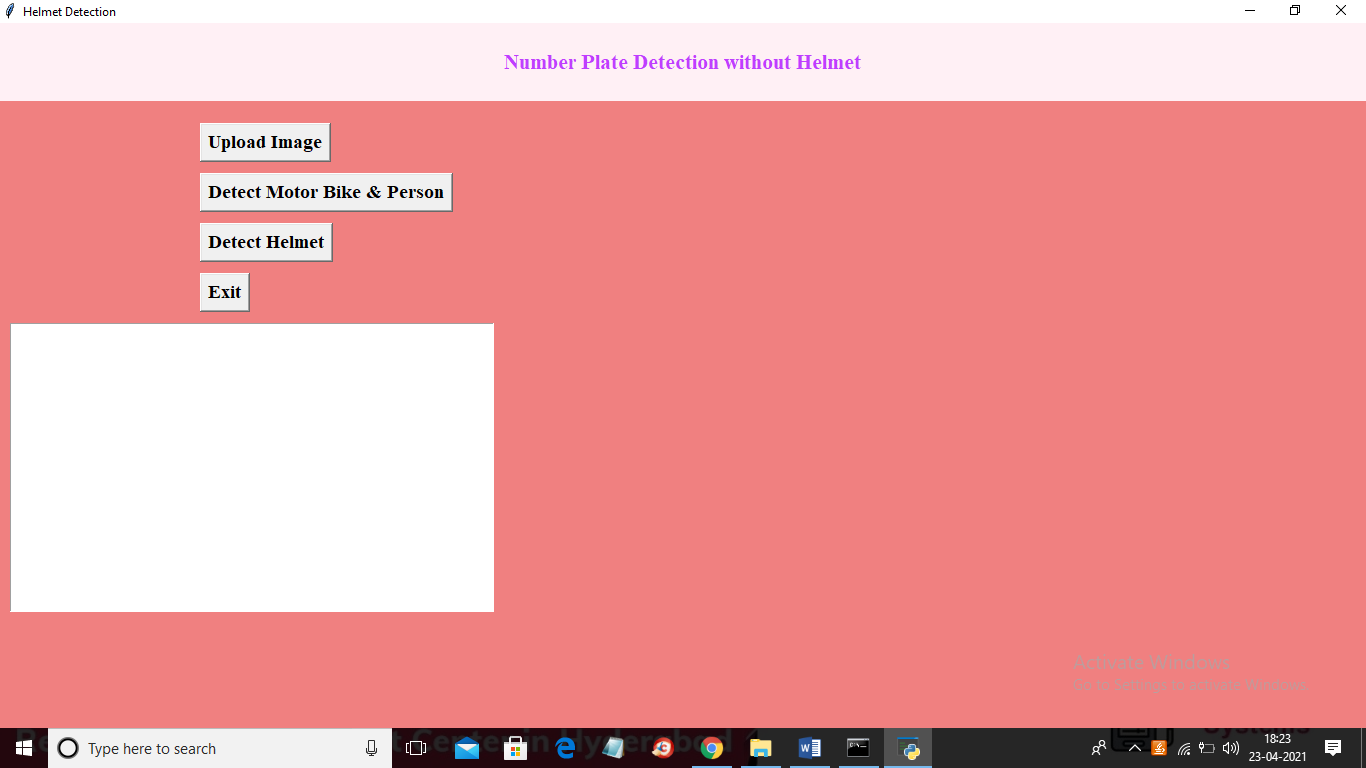
**4) Compatibility testing**

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.

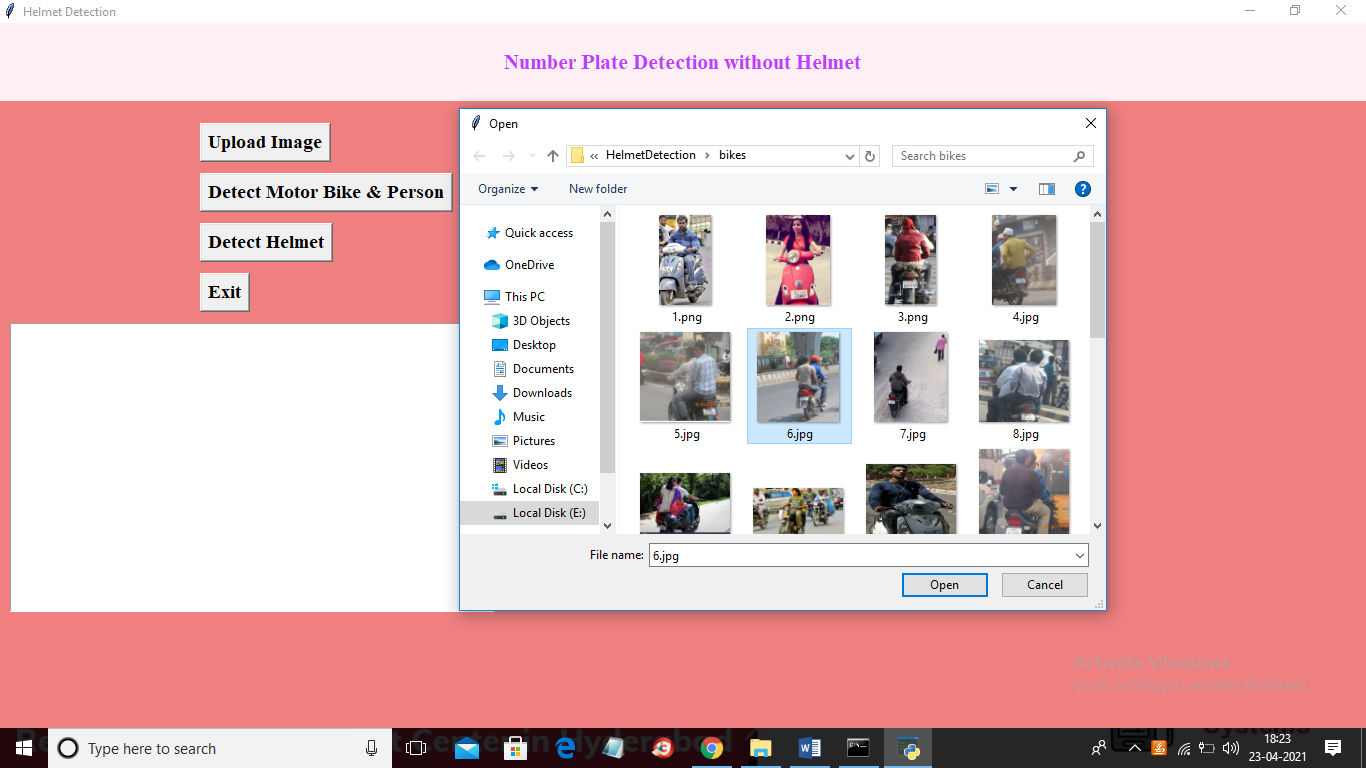
**RESULT**

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.

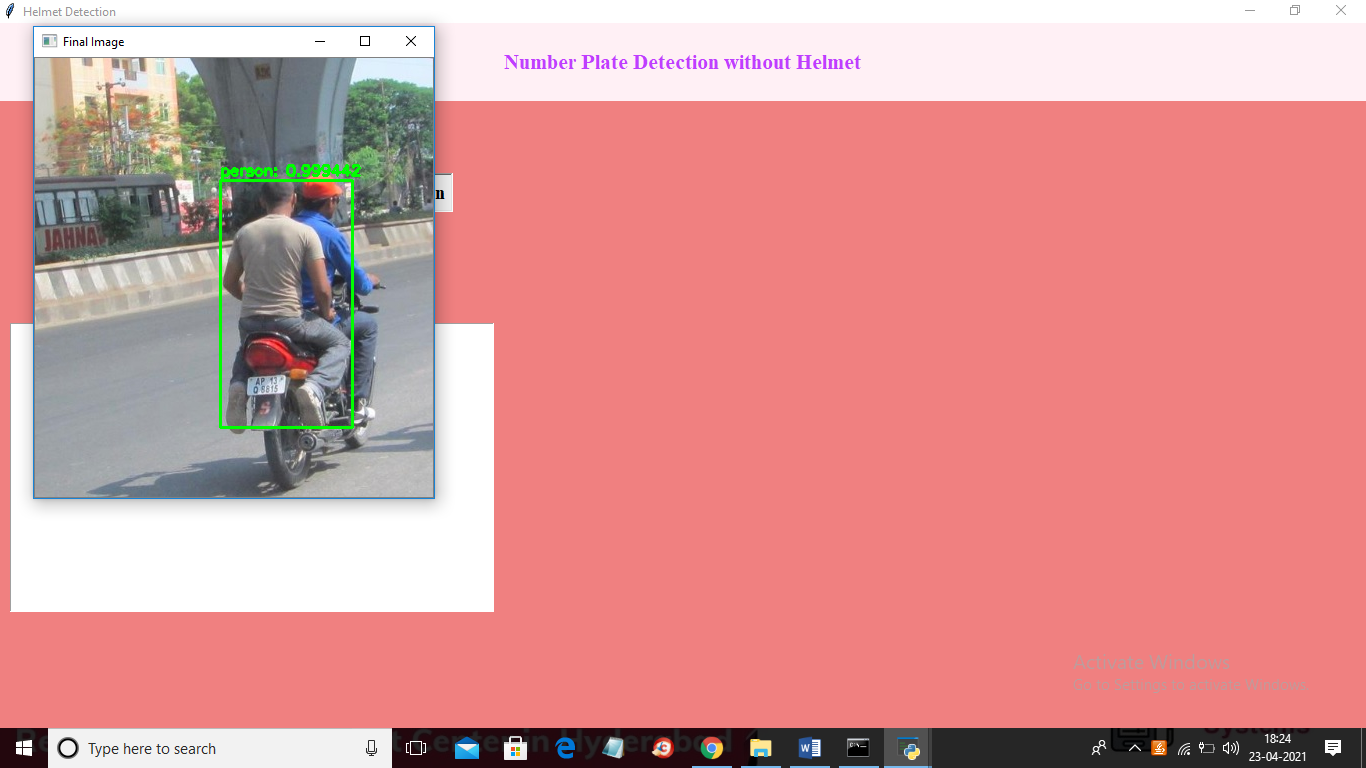
To run project double click on ‘run.bat’ file to get below screen



In above screen click on ‘Upload Image’ button to upload image



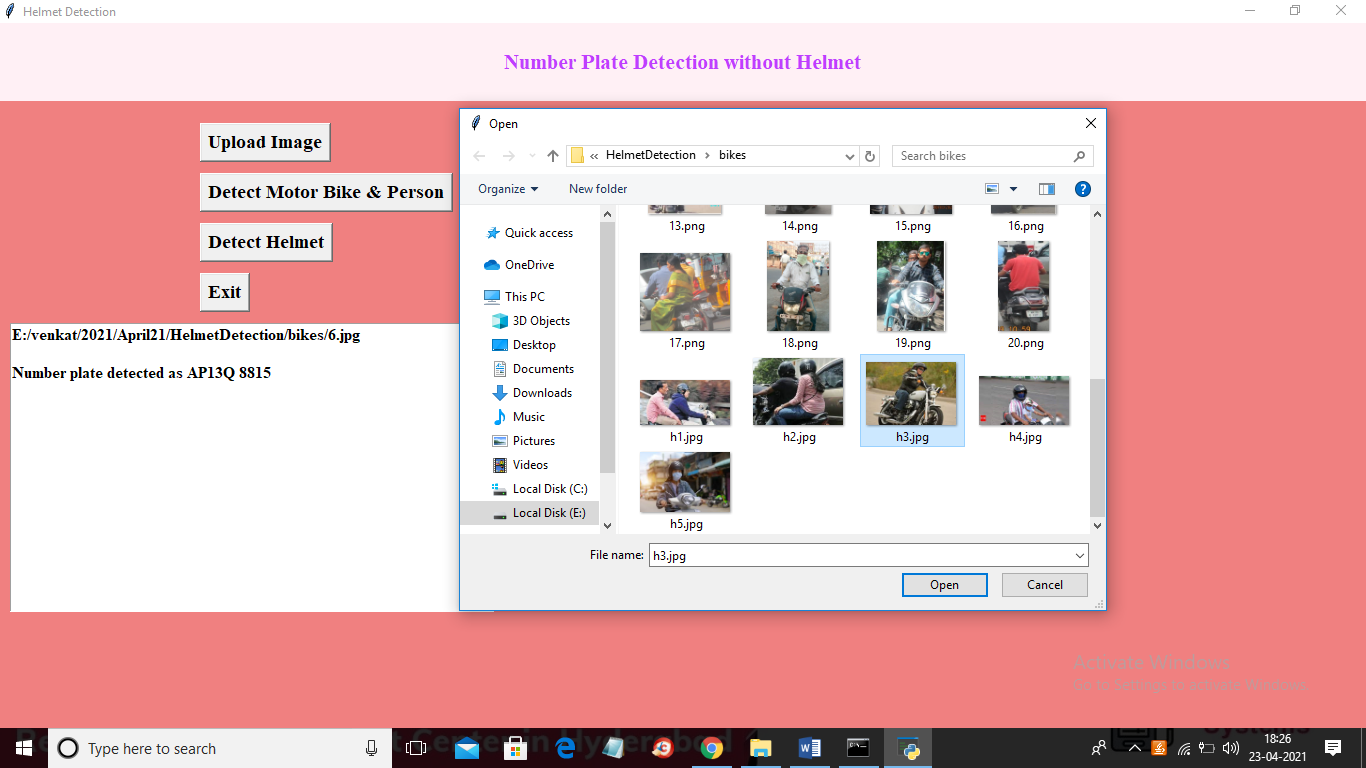
In above screen selecting and uploading ‘6.jpg’ file and then click on ‘Open’ button to load image and then click on ‘Detect Motor Bike & Person’ button to detect whether image contains person with bike or not



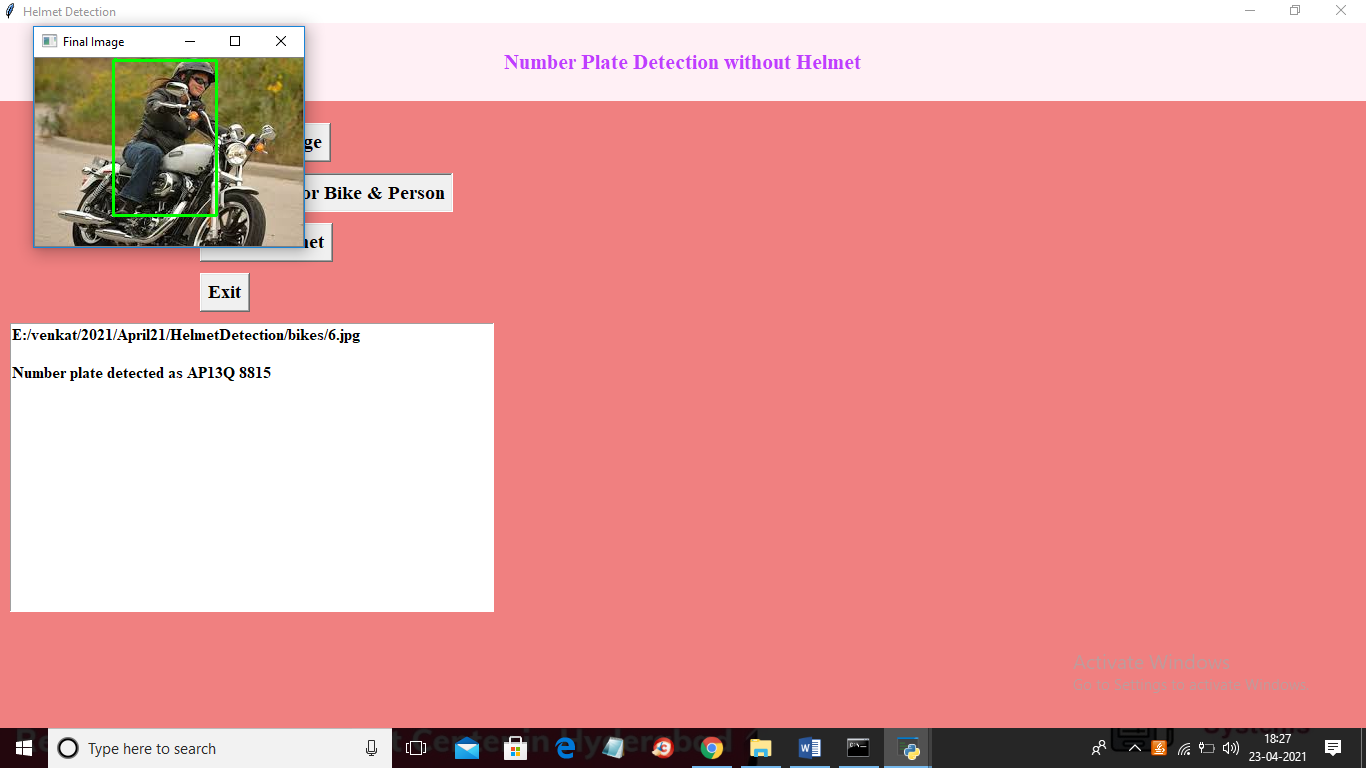
In above screen if person with bike detected then it put bounding box and then click on ‘Detect Helmet’ button to get below output



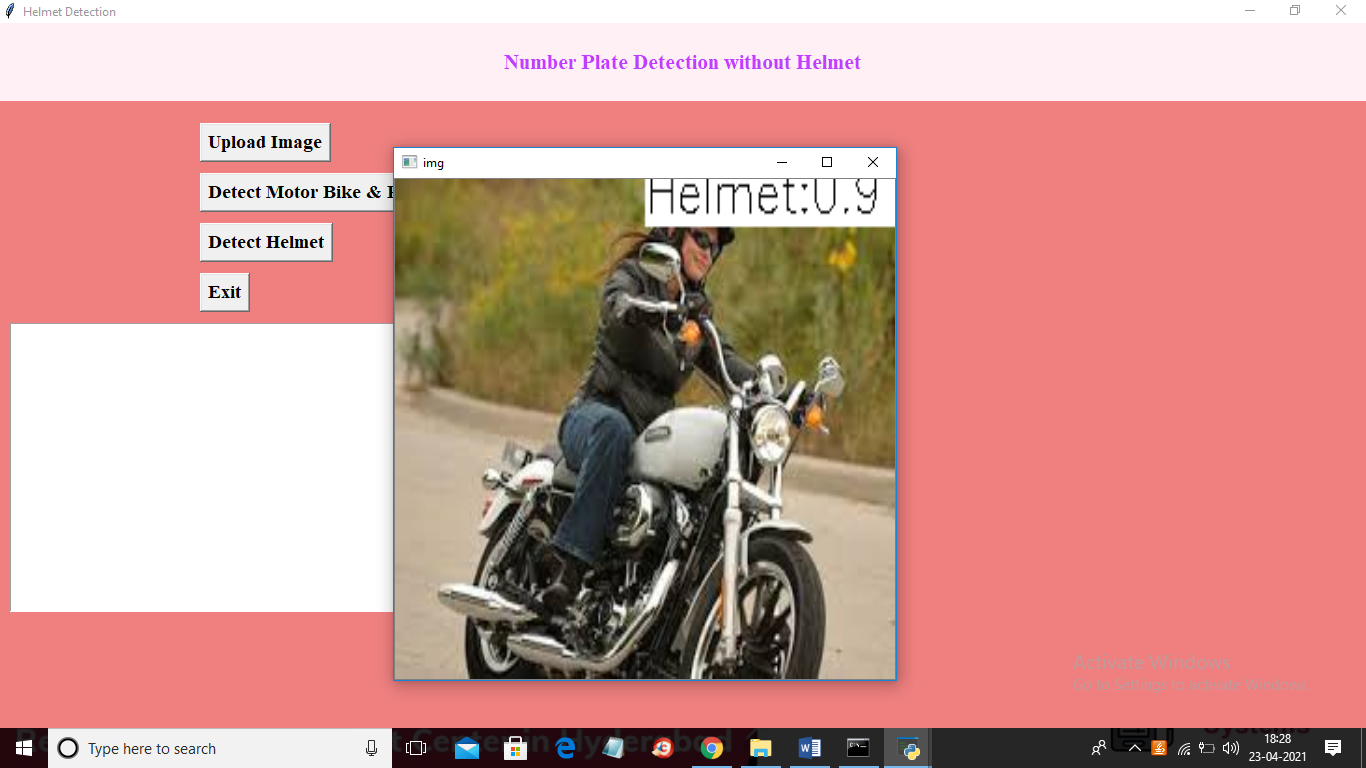
In above screen we can see helmet not detected and then application identify number plate and display on the text area as ‘AP13Q 8815’. Now try with other image by uploading it



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

**CONCLUSION**

We use the YOLOv5 Algorithm which is considered to be the best in real-time object detection algorithm. We have used two methods for detecting whether the motorcyclist is wearing a helmet or not and proceeded with the extraction of the number plate. This number plate character was displayed in text format. This text format is stored in a sheet for violating the law of not wearing a helmet. We achieved a mAP of 0.995 for detecting objects and overlapping conditions make our work different from any others. Our project makes this system autonomous and makes it easier for the authority to penalize people.

**REFERENCES**

[1] J. Chiverton, “Helmet presence classification with motorcycle detection and tracking,” Intelligent Transport Systems (IET), vol. 6, no. 3, September 2012.

[2] Z. Chen, T. Ellis, and S. Velastin, “Vehicle detection, tracking and classification in urban traffic,” in Procs. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITS), Anchorage, AK, Sept 2012.

[3] C. Patel, D. Shah, A. Patel, “Automatic Number Plate Recognition System”, Article in International Journal of Computer Applications, May 2013.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN” Submitted on 4 Jun 2015 (v1), last revised 6 Jan 2016.

[5] K. Dahiya, D. Singh, and C. K. Mohan, “Automatic detection of bike-riders without helmet using surveillance videos in real-time,” in 2016 International Joint Conference on Neural Networks (IJCNN), July 2016.

[6] Fan Wu, Guoquing Jin, “Helmet Detection Based On Improved YOLO V3 Deep Model”, in IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), May 2019.

[7] Felix Wilhelm Sieberta, Hanhe Linb "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks", Department of Psychology and Ergonomics, Berlin, Germany, 2019.

[8] Yonten J., Panomkhawn R., Rattapoom W., “Real-time number plate detection for non-helmeted motorcyclist using YOLO”, Received 29 November 2019; received in revised form 3 June 2020; accepted 31 July 2020, Available online 22 August 2020.

[9] A. Adil, H. S. Umer, M. Z. Khan, “Automatic Helmet Violation Detection of Motorcyclists from Surveillance Videos using Deep Learning Approaches of Computer Vision” in International Conference on Artificial Intelligence (ICAI), Pakistan, April 2021.

[10] Tushar Goel, Dr. K.C. Tripathi, Dr. M.L. Sharma, “Single Line License Plate Detection Using OpenCV And Tesseract”, International Research Journal of Engineering and Technology (IRJET), May 2020.

[11] Ritik Singh, Shubham Shetty, Gaurav Patil, Pramod J. Bide, “Helmet Detection Using Detectron2 and EfficientDet, 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), July 2021.

[12] Abhinu C G, Aswin P, Kiran Krishnan, Bonymol Baby, Dr. K S Angel Viji, “Multiple Object Tracking using Deep Learning with YOLO V5”, in International Journal of Engineering Research & Technology (IJERT), August 2021.

[13] Open Images Dataset (OIDv6): https://storage.googleapis.com/openimages/web/visualizer/inde x.html?set=train&type=segmentation&r=false&c=%2Fm%2F0 1d380

[14] Makesense (Image Annotation Tool): <https://www.makesense.ai/>

[15] LabelImg (Image Annotation Tool): <https://github.com/tzutalin/labelImg>

[16] Roboflow (Data Augmentation Tool): <https://roboflow.com/>

[17] Ultralytics YOLOv5: <https://ultralytics.com/yolov5>

[18] “Tesseract” vs “Keras-OCR” vs “EasyOCR”, Thanga Sami, 2021 URL: https://medium.com/mlearning-ai/tesseractvs-keras-ocr-vs-easyocr-ec8500b9455b

[19] Weights and Biases (Wandb): https://wandb.ai