

# **CAMPUS PLACEMENT MANAGEMENT PORTAL**

**A Project Report Submitted in partial fulfilment of there quirements for the award of  
the degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

<b>SHEIK TAJUDDIN</b>	<b>22221A05A5</b>
<b>PATI SATYA SRI</b>	<b>22221A0589</b>
<b>SHEIK ASHMA</b>	<b>22221A05A4</b>
<b>MUSINI MEGHANA</b>	<b>22221A0576</b>
<b>PECCHETI SAI RAMA RAO</b>	<b>22221A0592</b>
<b>MANEPALLI BADRINATH</b>	<b>22221A0571</b>

**Under the Esteemed Guidance of**

**Mr.M.S.VENUGOPALA RAO**  
**M. Tech.(Ph.D)**

**Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE  
(AUTONOMOUS)**

**(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)**

**(Accredited by N.B.A & NAAC with 'A' Grade)**

**ODALAREVU – 533210**

**2021-2025**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE  
(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)  
(Accredited by N.B.A & NAAC with ‘A’ Grade)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project work entitled “**CAMPUS PLACEMENT MANAGEMENT PORTAL**” is being submitted for the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, at BVC Engineering College, Odalarevu, is a bonafide work done by **SHEIK TAJUDDIN (22221A05A5), PATI SATYA SRI (22221A0589), SHEIK ASHMA (22221A05A4), MUSHINI MEGHANA (22221A0576), PECHETTI SAI RAMA RAO (22221A0592), MANEPALLI BADRINATH (22221A0571)** under my guidance during the academic year 2021-2025 and it has been found suitable for acceptance according to the requirement of the University.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

**Project Guide**  
**Mr.M.S.VENUGOPALA RAO**  
Professor

M.Tech

**Head of the Department**  
**Dr. MEERASHARIF SHEIK**  
M. Tech,Ph. D  
Professor & HOD

**External Examiner**

## **ACKNOWLEDGEMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards our guide **Mr.M.S.VENUGOPALA RAO**, M.Tech.(Ph.D),Associate Professor of CSE Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for his excellent guidance right from selection of the project and his valuable suggestions throughout the project work. His constant encouragement and support has been the cause of our success, in completing this thesis in the college.

We also grateful to the Project Co-Ordinator **Mr.M.V.K.SUBASH**, M.Tech.,(Ph.D) Assistant Professor of CSE Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for her excellent guidance right from selection of the project and her valuable suggestions throughout the project work.

We really thankful to **Dr. MEERASHARIF SHEIK**, M.Tech,Ph.D, Professor and Head of the Department of Computer Science and Engineering for providing the laboratory facilities to the fullest extent as and when required and also for giving us the opportunity to carry out the project work in the college.

We also thankful to **Dr. MAHESWAR DUTTA**, M.E, Ph.D, Principal, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for his sup port during and till the completion of the project. And also thankful to our **MANAGEMENT**, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for their support during and till the completion of the project.

We are thankful to all the **TEACHING AND NON-TEACHING STAFF** of Computer Science and Engineering Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, and **FRIENDS** for their direct and indirect help provided to us in completing the project.

Last but not least, we own all our success to our **FAMILY MEMBERS, CLASSMATES AND TEACHERS** from our child hood, whose vision, love and inspiration made us to reach out for these glories.

## **PROJECT ASSOCIATES**

<b>SHEIK TAJUDDIN</b>	<b>22221A05A5</b>
<b>PATI SATYA SRI</b>	<b>22221A0589</b>
<b>SHEIK ASHMA</b>	<b>22221A05A4</b>
<b>MUSINI MEGHANA</b>	<b>22221A0576</b>
<b>PECCHETI SAI RAMA RAO</b>	<b>22221A0592</b>
<b>MANEPALLI BADRINATH</b>	<b>22221A0571</b>

## **ABSTRACT**

The Campus Placement Management Portal is a web-based application developed using Python with the Django framework to automate and simplify the campus recruitment process in educational institutions. Traditional placement management systems heavily rely on manual data handling, which is time-consuming, inefficient, and prone to errors. This project provides a centralized and reliable digital platform to manage placement-related activities effectively.

The system includes two main user roles: Administrator and Student. The administrator can manage student records, company details, placement drives, and eligibility criteria, while students can securely register, update their profiles, and view available and eligible job opportunities. Django's built-in features such as authentication, ORM, and security mechanisms ensure data integrity, scalability, and secure access.

By using Python with Django, the Campus Placement Management Portal enhances transparency, reduces paperwork, improves communication between placement coordinators and students, and enables efficient tracking of recruitment activities. This application offers a structured, scalable, and user-friendly solution for modern campus placement management.

## LIST OF CONTENTS

<b>CONTENTS</b>	<b>PAGE NO</b>
<b>CHAPTER - 1 INTRODUCTION .....</b>	<b>1</b>
<b>CHAPTER - 2 LITERATURE SURVEY .....</b>	<b>2</b>
<b>CHAPTER - 3 SYSTEM ANALYSIS .....</b>	<b>3</b>
3.1 Existing System .....	3
3.1.1 Limitations of Existing System .....	4
3.2 Proposed System .....	4
3.2.1 Functional Overview of the Proposed Systems .....	5
3.2.2 Advantages of the Proposed Systems .....	5
3.2.3 Disadvantages of the Proposed Systems .....	6
<b>CHAPTER – 4 REQUIREMENT ANALYSIS .....</b>	<b>7</b>
4.1 Hardware Requirements .....	7
4.2 Software Requirements .....	7
<b>CHAPTER - 5 SOFTWARE DESIGN .....</b>	<b>8</b>
5.1 System Architecture .....	8
5.2 UML Diagrams .....	8
5.2.1 Class Diagram .....	8
5.2.2 Use Case Diagram .....	9
5.2.3 Sequence Diagram .....	10
<b>CHAPTER - 6 SYSTEM ENVIRONMENT .....</b>	<b>11</b>
6.1 What is Python .....	11
6.2 Advantages of Python over Other Languages.....	13
6.3 Disadvantages of Python .....	13
6.4 History of Python .....	14
6.5 What is Machine Learning .....	15
6.6 Categories of Machine Learning .....	15
6.7 Challenges in Machine Learning .....	16
6.8 Applications of Machine Learning .....	17
6.9 Python Development Steps .....	21
<b>CHAPTER – 7 MODULES USED IN PROJECT .....</b>	<b>23</b>
7.1 Install Python Step-by-Step in Windows or MAC .....	25
7.2 How to Install Python on Windows and Mac .....	25
<b>CHAPTER – 8 TESTING .....</b>	<b>32</b>
8.1 System Test .....	32
8.2 Types Of Tests .....	32
8.2.1 Unit testing .....	32
8.2.2 Integrated Testing .....	32

8.2.3 Functional test .....	33
8.2.4 White Box Testing .....	33
8.2.5 Black Box Testing .....	33
8.2.6 Unit Testing .....	33
<b>CHAPTER - 9 IMPLEMENTATION .....</b>	<b>36</b>
9.1 Methodology .....	36
9.2 Source Code .....	38
<b>CHAPTER – 10 RESULTS .....</b>	<b>42</b>
<b>CHAPTER – 11 CONCLUSION.....</b>	<b>50</b>
<b>CHAPTER – 12 REFERENCES .....</b>	<b>51</b>

## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE NO</b>
Fig 5.1 System Architecture .....	8
Fig 5.2 Class Diagram.....	9
Fig 5.3 Use Case Diagram .....	9
Fig 5.4 Sequence Diagram .....	10
Fig 7.1 Open Python website to download the python latest version .....	26
Fig 7.2 Click on Download Python.....	26
Fig 7.3 Select the version what you want .....	27
Fig 7.4 Python Versions .....	27
Fig 7.5 Click on ok to install Python .....	28
Fig 7.6 Click on Install Now to install .....	28
Fig 7.7 After installation successful. Click on Close .....	29
Fig 7.8 Open Command Prompt .....	29
Fig 7.9 Type Python -V to check the python is correctly installed or not .....	30
Fig 7.10 Click on IDLE (Python3.7 64-bit) and launch the program .....	30
Fig 7.11 Click on File and Save the file .....	31
Fig 10.1 DataSet Contains following images .....	42
Fig 10.2 Run Server to start Python Server .....	43
Fig 10.3 After running server it shows user login .....	44
Fig 10.4 User login screen .....	44
Fig 10.5 Train Yolov8 alogorithm .....	45
Fig 10.6 Training Graph .....	46
Fig 10.7 Upload Test Image .....	46
Fig 10.8 select the image and sumbit .....	47
Fig 10.9 detected animal species is bear .....	47
Fig 10.10 Above species detected as ‘Bison’ .....	48
Fig 10.11 Above species detected as ‘Antelope’. ....	48
Fig 10.12 Above species detected as ‘Honey Bee’ .....	49
Fig 10.13 Above species detected as ‘Buffalo’ and similarly you can test any other image ...	49

## CHAPTER - 1

### INTRODUCTION

Campus placement is a vital activity in higher educational institutions, acting as a bridge between students and the corporate world. It provides students with career opportunities while helping institutions maintain strong industry connections and placement records. As the number of students and recruiting companies increases each year, managing placement activities manually becomes inefficient, time-consuming, and prone to errors. Hence, there is a strong need for a reliable, automated system to manage campus recruitment processes effectively.

The Campus Placement Management Portal is a web-based application developed using Python with the Django framework to automate and streamline the placement management process. The system is designed to replace traditional manual methods such as paperwork, spreadsheets, and offline communication with a centralized digital platform. This portal enables effective management of student information, company details, job postings, eligibility criteria, and placement drives in a structured and secure manner.

The application supports role-based access control with two primary users: Administrator and Student. The administrator is responsible for managing student records, updating company information, scheduling placement drives, and defining eligibility conditions. Students can register into the system, update their academic and personal details, and view placement opportunities for which they are eligible. This ensures transparency and equal access to placement information for all students.

Python with Django is chosen as the development framework due to its simplicity, scalability, and strong security features. Django's built-in authentication system, Object Relational Mapping (ORM), and MVC-based architecture ensure secure data handling, efficient database management, and rapid application development. The system also supports easy maintenance and future enhancements such as adding new user roles or advanced analytics.

Overall, the Campus Placement Management Portal provides an efficient, user-friendly, and scalable solution for managing campus recruitment activities. By reducing manual effort, improving accuracy, and enabling real-time access to placement data, the system enhances coordination between students and placement administrators and contributes to a more organized and effective placement process.

## CHAPTER - 2

### LITERATURE SURVEY

Campus placement management is a critical administrative function in higher educational institutions, connecting students with potential employers and ensuring smooth recruitment processes. Earlier placement systems were largely manual, relying on paper-based records, spreadsheets, notice boards, and verbal communication. Studies highlight that such traditional methods are inefficient, time-consuming, and prone to data inconsistency, making it difficult for placement officers to manage large volumes of student and company information effectively.

Several researchers have proposed web-based placement management systems to overcome these challenges. These systems focus on digitizing student records, company profiles, and placement drive information into a centralized database. Literature indicates that web portals significantly reduce paperwork, improve accessibility of information, and enable faster communication between students and placement coordinators. Most of the surveyed systems emphasize features such as student registration, resume management, company postings, and eligibility checking.

Role-based access control is another key aspect discussed in many studies. Existing literature suggests separating functionalities for administrators and students to ensure data security and operational efficiency. Administrators manage placement drives, eligibility criteria, and student data, while students access placement information and apply for eligible opportunities. This role-based structure minimizes unauthorized access and improves system reliability.

Recent research also highlights the adoption of modern web development frameworks to build scalable placement portals. Python-based frameworks, especially Django, are widely recommended due to their simplicity, rapid development capabilities, and strong security features. Django's built-in authentication system, Object Relational Mapping (ORM), and modular architecture support secure user management and efficient database operations. Literature emphasizes that using Django reduces development complexity while ensuring maintainability and scalability.

From the literature survey, it is evident that while many placement systems address digitization and automation, gaps still exist in terms of real-time updates, scalability, and user-friendly design. These limitations motivate the development of a robust Campus Placement Management Portal using Python with Django, which aims to provide a secure, centralized, and efficient solution for managing campus recruitment activities.

## CHAPTER - 3

### SYSTEM ANALYSIS

System analysis is a fundamental phase in the Software Development Life Cycle (SDLC) that focuses on understanding the problem domain, evaluating the existing system, and defining the functional requirements of the proposed solution. In educational institutions, campus placement plays a critical role in shaping students' careers and strengthening institutional reputation. As the volume of student data and recruiting companies increases, managing placement activities using traditional methods becomes inefficient and unreliable.

The Campus Placement Management Portal aims to analyze the shortcomings of the current placement process and propose an automated, centralized, and secure web-based solution using Python with the Django framework.

#### 3.1 Existing System

In the existing campus placement process, most educational institutions follow a manual or semi-automated approach to manage placement activities. Student information, academic records, and eligibility details are collected through paper forms or basic digital formats such as spreadsheets. Company information and placement schedules are often communicated through notice boards, emails, or messaging groups, which lack proper organization and traceability.

Placement officers are required to manually verify student eligibility for each recruitment drive based on criteria such as academic performance, branch, and year of study. This verification process is time-consuming and prone to human errors, especially when handling a large number of students. Any change in eligibility criteria requires rechecking student records manually, increasing administrative workload.

The existing system also lacks a centralized database, resulting in data being stored across multiple files and systems. This fragmentation leads to data redundancy, inconsistencies, and difficulty in retrieving accurate information. Maintaining historical placement data for future reference or institutional reporting becomes challenging due to unstructured data storage.

Communication between placement coordinators and students is another major concern. Students depend on manual announcements, emails, or messaging platforms, which may not reach all eligible candidates on time. As a result, students may miss important placement opportunities due to delayed or unclear communication. There is no mechanism to track whether a student has received or acknowledged a placement notification.

Security is a significant limitation of the existing system. Sensitive student information such as academic records, personal details, and placement status is often stored without proper access control. Unauthorized access or accidental data loss can occur due to improper data handling practices. Additionally, there is no reliable backup mechanism to protect placement data from system failures or accidental deletion.

Scalability is another major issue with the existing system. As the number of students, companies, and placement drives increases, manual systems fail to handle the growing workload effectively. The lack of automation makes it difficult for placement cells to manage large-scale recruitment processes efficiently.

Overall, the existing campus placement system lacks automation, integration, security, and scalability. These drawbacks significantly affect the efficiency and reliability of placement activities, highlighting the urgent need for a centralized, automated, and secure placement management solution.

### **3.1.1 Limitations of Existing Systems**

The existing placement management system suffers from several limitations:

- Manual Processing: Most placement operations require human intervention, increasing time and effort.
- Data Redundancy: Repeated data entry causes inconsistencies and errors.
- Limited Accessibility: Students depend on physical notices or messages for updates.
- Poor Data Security: Sensitive student data is not adequately protected.
- Lack of Automation: Eligibility checking and reporting are done manually.
- Scalability Issues: The system cannot efficiently handle large numbers of students and companies.

## **3.2 Proposed System**

The proposed Campus Placement Management Portal is a web-based application developed using Python with Django to automate and streamline campus placement activities. The system provides a centralized platform where all placement-related information is stored in a structured and secure database.

The proposed system supports role-based access, allowing administrators and students to interact with the portal according to their privileges. Administrators can manage student profiles, company details, placement drives, and eligibility criteria, while students can register, update their information, and view placement opportunities for which they are eligible.

Django's built-in authentication, ORM, and security features ensure secure access, data integrity, and efficient database operations. Automated eligibility checks and real-time updates reduce manual effort and improve accuracy.

### **3.2.1 Functional Overview of the Proposed Systems**

#### **Administrator Functions**

- Manage student information
- Add and update company profiles
- Create and schedule placement drives
- Define eligibility criteria
- Monitor placement activities and reports

#### **Student Functions**

- Register and login securely
- Update personal and academic profiles
- View eligible placement drives
- Receive placement-related notifications

### **3.2.2 Advantages of the Proposed Systems**

The proposed system offers several advantages over the existing system:

- Automation: Reduces manual work and human errors
- Centralized Data Management: All placement data stored in a single database
- Improved Accessibility: Students can access placement information anytime
- Enhanced Security: Role-based access and secure authentication
- Time Efficiency: Faster eligibility checking and data retrieval
- Scalability: Can handle growing numbers of users and companies
- Transparency: Equal access to placement information for all students

### **3.2.3 Disadvantages of the proposed Systems**

Despite its benefits, the proposed system has some limitations:

- Initial Setup Cost: Requires time and resources for development and deployment
- Internet Dependency: Continuous internet access is required
- Technical Knowledge: Administrators may require basic training
- Maintenance: Regular updates and maintenance are needed to ensure security

## CHAPTER – 4

### REQUIREMENT ANALYSIS

#### **4.1 Hardware Requirements**

- **Processor :** Intel i5 or Above
- **Ram :** Minimum 512 Mb
- **Secondary Storage :** Minimum 40 GB
- **Supported Devices :** Desktop Computers, Laptops, Tablets, Smartphones

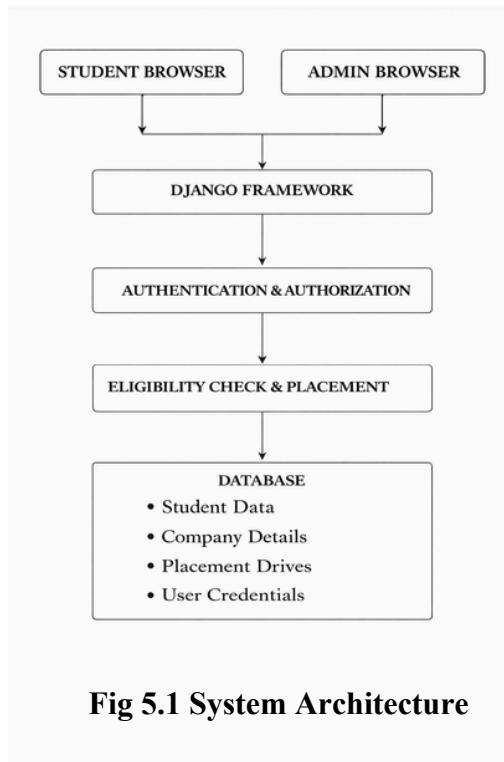
#### **4.2 Software Requirements**

- **Operating System :**
  - **Server-Side :** Windows 10 / 11, Linux
  - **Client-side :** Windows 10 / 11, Linux, macOS, Android / iOS
- **Development Environment :**
  - **Programming Language :** Python 3.14.2
  - **Web Framework :** Django 6.0
  - **IDE / Code Editor :** Visual Studio Code
  - **Package Manager :** pip (Python Package Installer)
- **Frontend Technologies :** HTML5, CSS3
- **Backend Technologies :** Python, Django Framework
- **Database Requirements :**
  - **Database Type :** Relational Database Management System (RDMS)
  - **Supported Databases :** MySQL 8.0.40
- **Browser Requirements :** Google Chrome, Microsoft Edge
- **Security Requirements :**
  - Secure login and authentication
  - Role-based access control (Admin, Student)
  - Regular Database Backups

## CHAPTER - 5

# SOFTWARE DESIGN

### 5.1 System Architecture

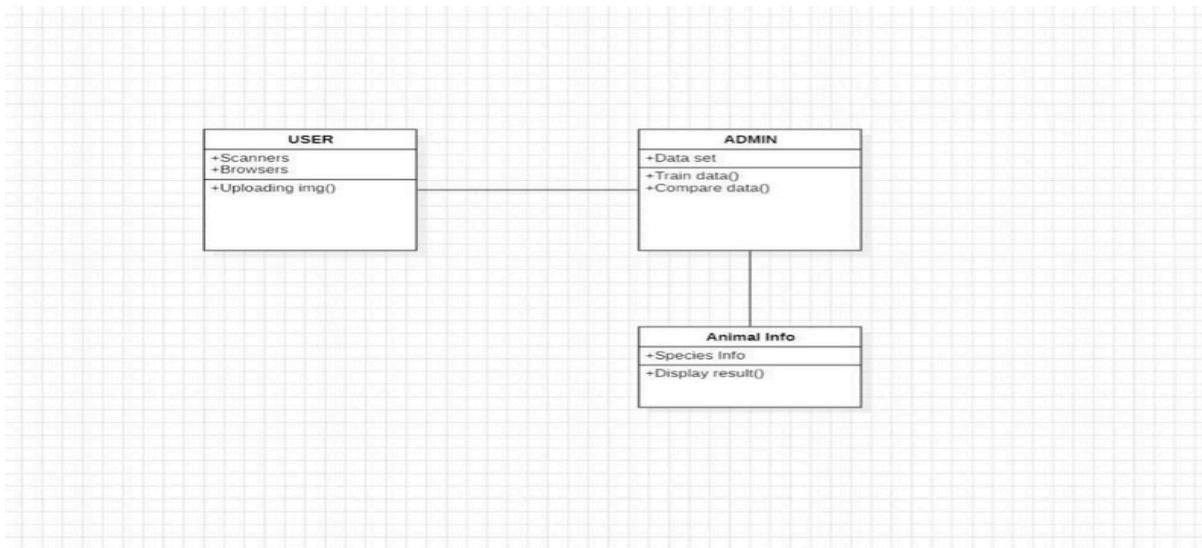


**Fig 5.1 System Architecture**

### 5.2 UML Diagrams

#### 5.2.1 Class Diagram

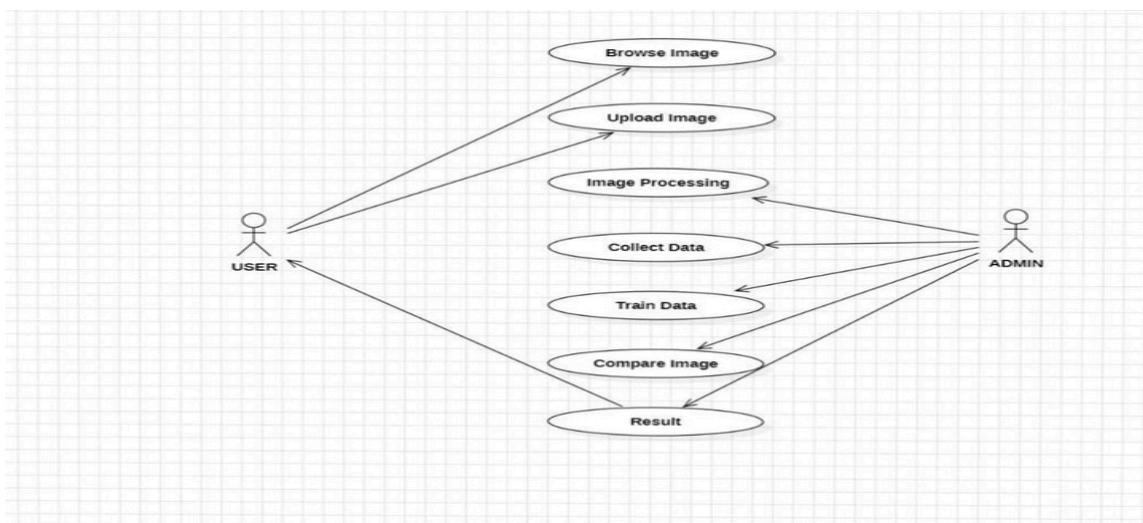
The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely.



**Fig 5.2 Class Diagram**

### 5.2.2 Use Case Diagram

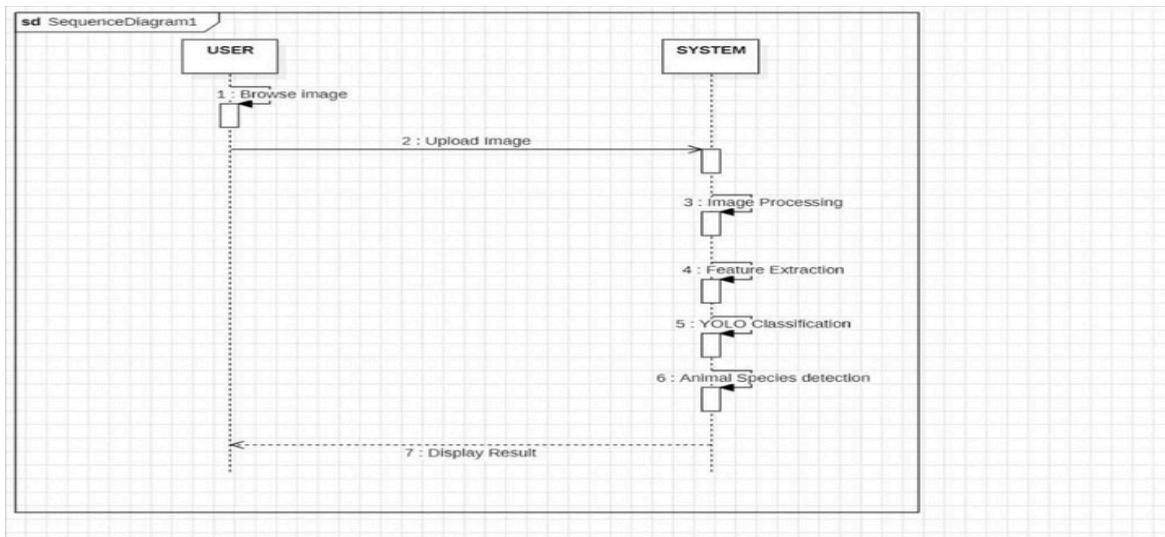
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



**Fig 5.3 Use Case Diagram**

### 5.2.3 Sequence Diagram

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



**Fig 5.4 Sequence Diagram**

## CHAPTER - 6

# SYSTEM ENVIRONMENT

### 6.1 What is Python

Below are some facts about Python.

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

The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc. )
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Advantages of Python :-
- Let's see how Python dominates over other languages.

#### 1. Extensive Libraries

Python comes with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

### **3. Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

### **4. Improved Productivity**

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

### **5. IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

### **6. Simple and Easy**

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

### **7. Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

### **8. Object-Oriented**

This language supports both the procedural and object- oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

### **9. Free and Open-Source**

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

### **10. Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere

(WORA). However, you need to be careful enough not to include any system-dependent features.

## **11. Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

## **6.2 Advantages of Python over Other Languages**

### **1. Less Coding**

- Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

### **2. Affordable**

- Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support. The 2019 Github annual survey showed us that
- Python has overtaken Java in the most popular programming language category.

### **3. Python is for Everyone**

- Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## **6.3 Disadvantages of Python**

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

### **1. Speed Limitations**

- We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

## 2. Weak in Mobile Computing and Browsers

- While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.
- The reason it is not so famous despite the existence of Brython is that it isn't that secure.

## 3. Design Restrictions

- As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

## 4. Underdeveloped Database Access Layers

- Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

## 5. Simple

- No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary. This was all about the Advantages and
- Disadvantages of Python Programming Language.

## 6.4 History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners<sup>1</sup>, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI).

I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all

my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

## 6.5 What is Machine Learning

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data.

Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

## 6.6 Categories of Machine Learning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification

tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction.

Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## 6.7 Challenges in Machine Learning

- While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are-
- Quality of data – Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.
- Time-Consuming task – Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

- Lack of specialist persons – As ML technology is still in its infancy stage, availability of expert resources is a tough job.
- No clear objective for formulating business problems – Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
- Issue of overfitting & underfitting – If the model is overfitting or underfitting, it cannot be represented well for the problem.
- Curse of dimensionality – Another challenge ML model faces is too many features of data points. This can be a real hindrance.
- Difficulty in deployment – Complexity of the ML model makes it quite difficult to be deployed in real life.

## 6.8 Applications of Machine Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML –

- Emotion analysis
- Sentiment analysis
  - Error detection and prevention
  - Weather forecasting and prediction
- Stock market analysis and forecasting
  - Speech synthesis
  - Speech recognition
- Customer segmentation
  - Object recognition
  - Fraud detection
  - Fraud prevention
- Recommendation of products to customer in online shopping

## How to Start Learning Machine Learning?

- Arthur Samuel coined the term “Machine Learning” in 1959 and defined it as a “Field of study that gives computers the capability to learn without being explicitly programmed”.
- And that was the beginning of Machine Learning! In modern times, Machine Learning is

one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of \$146,085 per year.

- But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

## **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### **Step 1 – Understand the Prerequisites**

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### **a) Learn Linear Algebra and Multivariate Calculus**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### **b) Learn Statistics**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

### c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### a) Terminologies of Machine Learning

- Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis. Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- Training – The idea is to give a set of inputs(features) and its expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide

#### b) Types of Machine Learning

- Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

- Unsupervised Learning – This involves using unlabeled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- Semi-supervised Learning – This involves using unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- Reinforcement Learning – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### **Advantages of Machine learning :-**

#### **1. Easily identifies trends and patterns-**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### **2. No human intervention needed (automation)**

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### **3. Continuous Improvement**

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### **4. Handling multi-dimensional and multi-variety data**

Machine Learning algorithms are good at handling data that are multi- dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### **5. Wide Applications**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

## Disadvantages of Machine Learning :-

### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

### 4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

## 6.9 Python Development Steps

- Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x.
- The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python:

"There should be one -- and preferably only one-- obvious way to do it." Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "://" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

### Purpose:-

- We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

### Python

- Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.
- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.
- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background- without breaking.

## CHAPTER – 7

### MODULES USED IN PROJECT

#### **Tensorflow**

- TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.
- TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

#### **Numpy**

- Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:
- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities
- Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

#### **Pandas**

- Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

## Matplotlib

- Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.
- For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

## Scikit – learn

- Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.
- Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.
- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.
- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other

languages can pick up basic Python skills and the huge standard library is key to another area where Python excels.

- All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background- without breaking.

## 7.1 Install Python Step-by-Step in Windows or MAC

- Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high- level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct
- provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

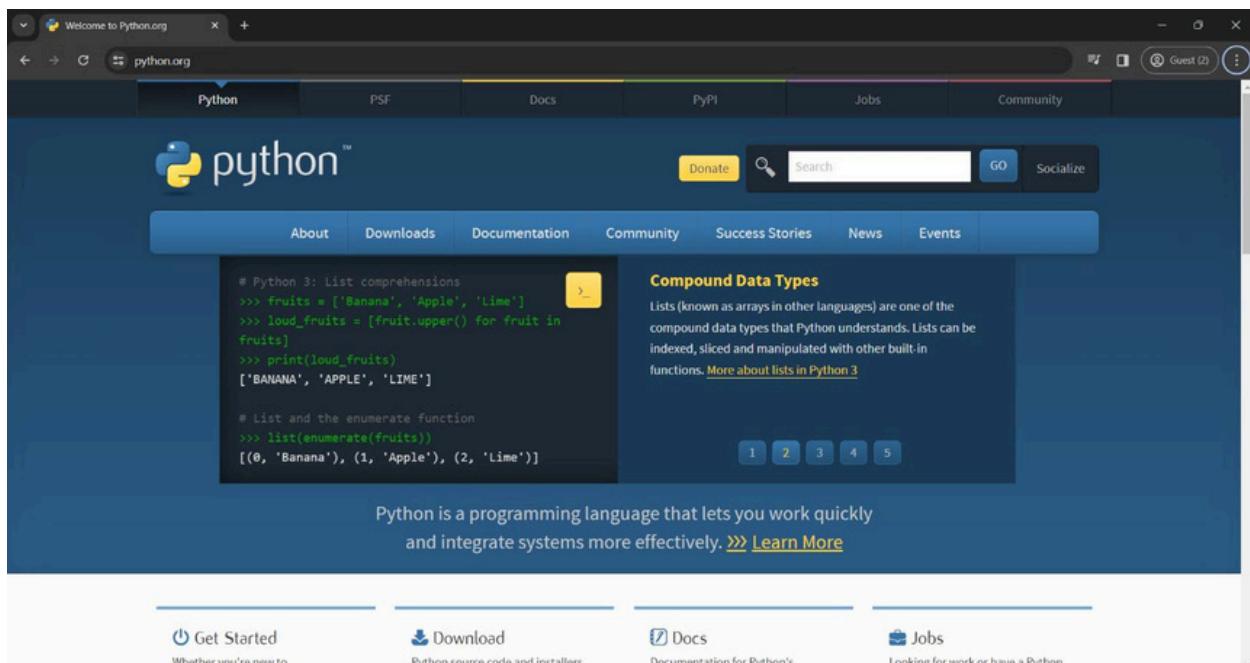
## 7.2 How to Install Python on Windows and Mac

- There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.
- Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.
- Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e.
- operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

**Step-1:** Go to the official site and use Google Chrome or any other web browser to download and install Python. Or click on the following link:

<https://www.python.org/>



**Fig 7.1 Open Python website to download the python latest version**

Now check the latest and correct version of your operating system.

**Step-2:** Click on the Download Tab.



**Fig 7.2 Click on Download Python**

**Step-3:** You can select the yellow Download Python for Windows3.7.4 button, or you can scroll download click the download of the corresponding version. Here, we are downloading the latest version of Python for Window 3.7.4.

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
<a href="#">Python 3.7.4</a>	July 8, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 3.6.9</a>	July 2, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 3.7.3</a>	March 25, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 3.4.10</a>	March 18, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 3.5.7</a>	March 18, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 2.7.16</a>	March 4, 2019	Download	<a href="#">Release Notes</a>
<a href="#">Python 3.7.2</a>	Dec. 24, 2018	Download	<a href="#">Release Notes</a>

**Fig 7.3 Select the version what you want**

**Step-4:** Scroll down the page until you find the "File" option.

**Step5:** Here you will see different versions of Python and operating systems.

Files						
Version	Operating System	Description	MD5 Sum	File Size	GPG	
<a href="#">Gzipped source tarball</a>	Source release		68111671e5b2db4aef7b9ab01bf0f9be	22.0 MB	<a href="#">SIG</a>	
<a href="#">XZ compressed source tarball</a>	Source release		d33e4aae66097051c2eca45ee3604803	16.3 MB	<a href="#">SIG</a>	
<a href="#">macOS 64-bit installer</a>	macOS	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	26.8 MB	<a href="#">SIG</a>	
<a href="#">macOS 64-bit/32-bit installer</a>	macOS	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	33.3 MB	<a href="#">SIG</a>	
<a href="#">Windows help file</a>	Windows		d63999573a2c06b2ac56cade6b4f7cd2	7.8 MB	<a href="#">SIG</a>	
<a href="#">Windows x86 embeddable zip file</a>	Windows		9fab3b81f8841879fd94133574139d8	6.4 MB	<a href="#">SIG</a>	
<a href="#">Windows x86 executable installer</a>	Windows		33cc602942a54446a3d6451476394789	24.5 MB	<a href="#">SIG</a>	
<a href="#">Windows x86 web-based installer</a>	Windows		1b670cfa5d317df82c30983ea371d87c	1.3 MB	<a href="#">SIG</a>	
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7.2 MB	<a href="#">SIG</a>	
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	25.4 MB	<a href="#">SIG</a>	
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1.3 MB	<a href="#">SIG</a>	

**Fig 7.4 Python Versions**

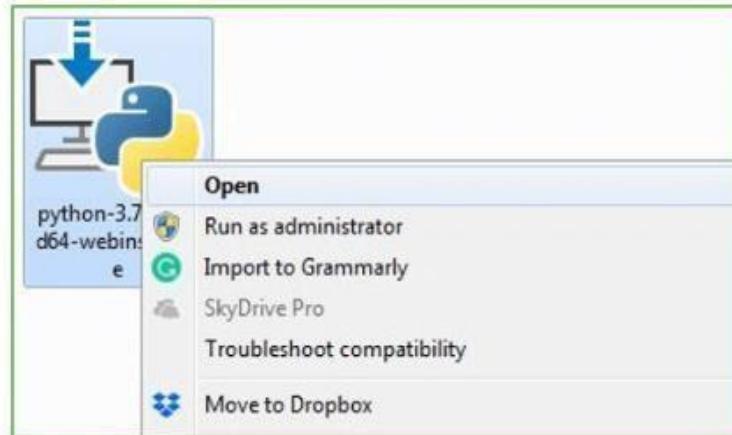
- To download Windows 32bit Python, you can select the built-in Windows X86 Zip file, WindowsX86 executable installer or WindowsX86 installer on the Web.
- To download Windows 64 bit Python, you can select any option of the three options. Zip File Embeddable Windows X866, Windows X8664 executable installer or Windows X8664 installer based on the web.

Here you will install the installer based on the Windows X8664 website. Here, the first part of the Python version was completed. Now we will go in advance the second part when installing PythonI. Note: You can click on the option of the release of the version to know the changes or updates made in the version.

## Installation of Python

**Step-1:** Goto Download and Open the downloaded python version to carry out the installation process.

**Step-2:** Before you click on Install Now, make sure to put a tick on Add Python 3.7 to Path



**Fig 7.5 Click on ok to install Python**



**Fig 7.6 Click on Install Now to install**

**Step-3:** Click on Install Now After the installation is successful. Click on Close.



**Fig 7.7 After installation successful. Click on Close**

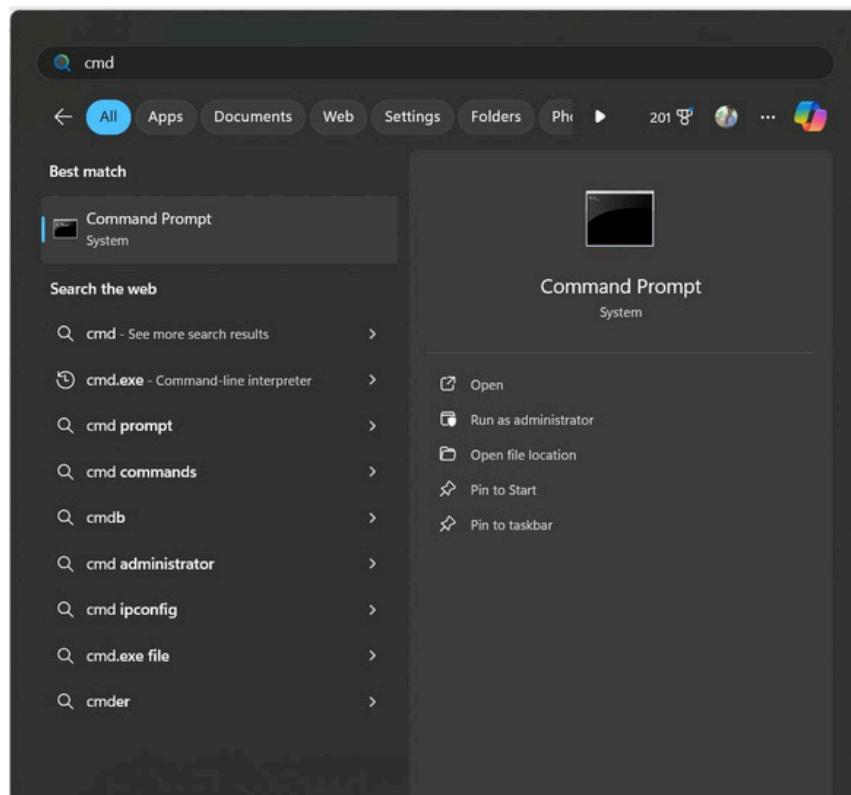
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

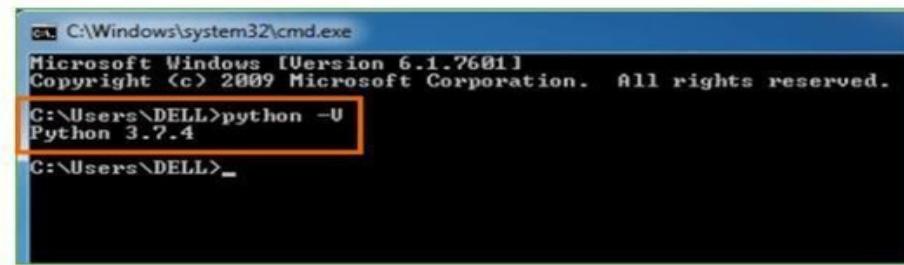
**Step-1:** Click on Start

**Step-2:** In the Windows Run Command, type “cmd”.



**Fig 7.8 Open Command Prompt**

**Step-3:** Open the Command prompt option.



**Fig 7.9 Type Python -V to check the python is correctly installed or not**

**Step-4:** Let us test whether the python is correctly installed. Type **python -V** and press Enter.

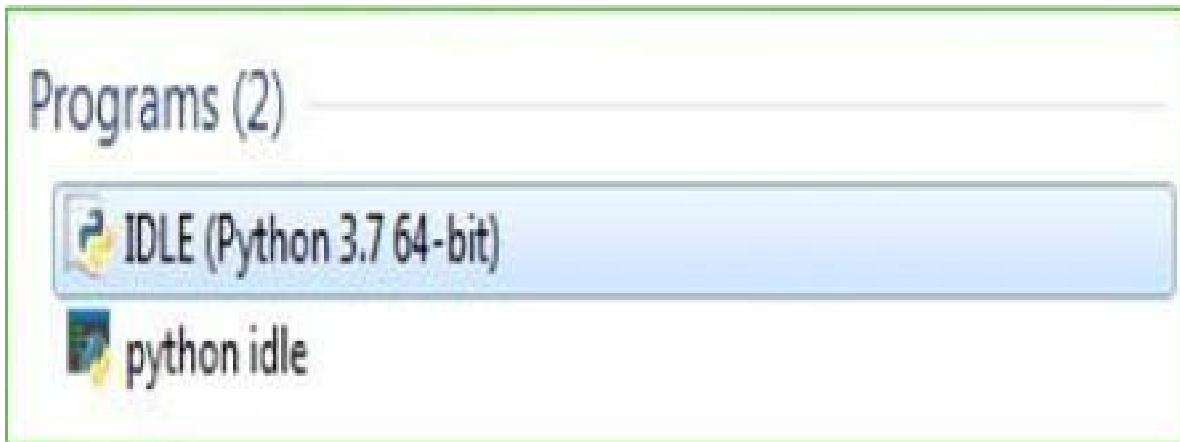
**Step-5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step-1:** Click on Start

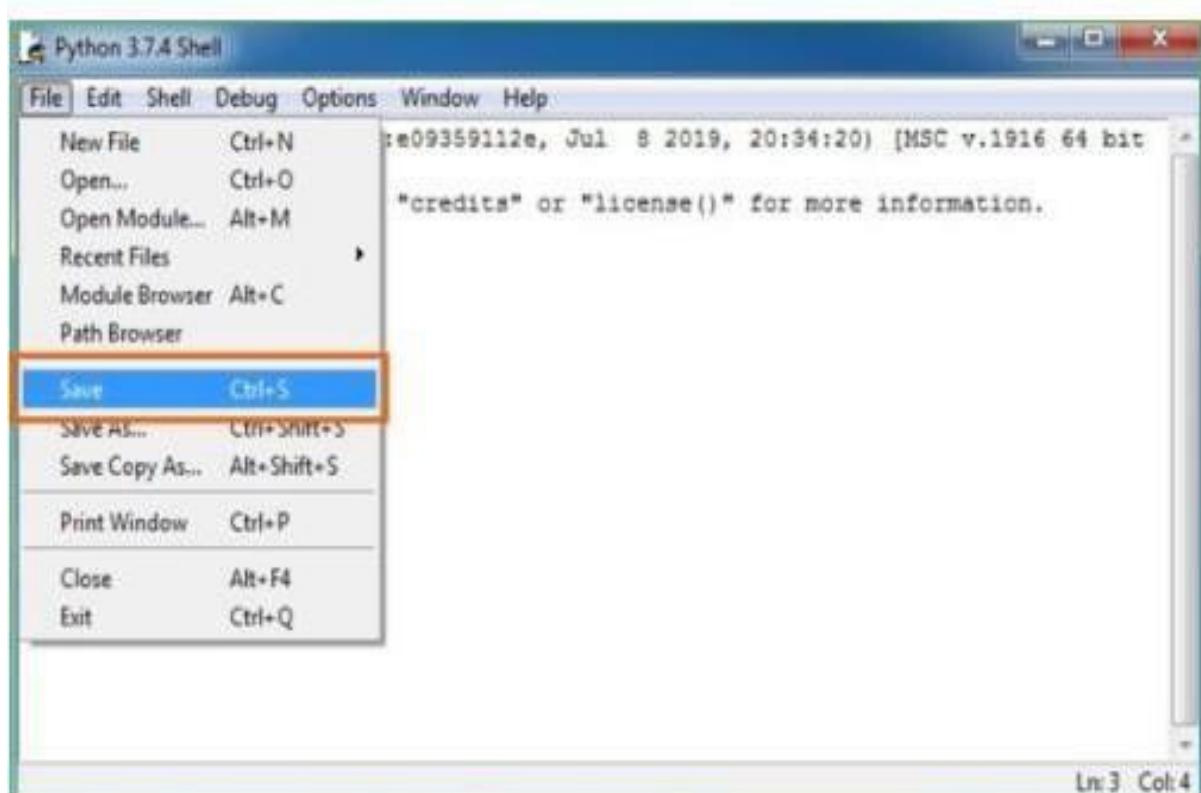
**Step-2:** In the Windows Run command, type “python idle”.



**Fig 7.10 Click on IDLE (Python3.7 64-bit) and launch the program**

**Step-3:** Click on IDLE (Python3.7 64-bit) and launch the program

**Step-4:** To go ahead with working in IDLE you must first save the file. Click on File Click on Save



**Fig 7.11 Click on File and Save the file**

**Step-5:** Name the file and save as type should be Python files.

Click on SAVE. Here I have named the files as Hey World.

**Step-6:** Now for e.g. enter print

## CHAPTER – 8

# TESTING

### 8.1 System Test

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 8.2 Types Of Tests

#### 8.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### 8.2.2 Integrated Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 8.2.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

**Functional testing is centered on the following items:**

1. **Valid Input:** Identified classes of valid input must be accepted.
2. **InvalidInput:** Identified classes of invalid input must be rejected
3. **Functions:** Identified functions must be exercised.
4. **Output:** Identified classes of application outputs must be exercised.
5. **Systems/Procedures:** Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### 8.2.4 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

### 8.2.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### 8.2.6 Unit Testing

- Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

- Test strategy and approach
- Field testing will be performed manually and functional tests will be written in detail.
- Test objectives
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.
- Integration Testing
- Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
- The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.
- Test Results: All the test cases mentioned above passed successfully. No defects encountered.

## Acceptance Testing

- User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Test Case – 1:

Test case for Login form:

FUNCTION:	LOGIN
EXPECTED RESULTS:	Should Validate the user and check his existence in database
ACTUAL RESULTS:	Validate the user and checking the user against the database
LOW PRIORITY	No

HIGH PRIORITY	Yes
---------------	-----

### Test Case – 2:

Test case for User Registration form:

FUNCTION:	USER REGISTRATION
EXPECTED RESULTS:	Should check if all the fields are filled by the user and saving the user to database.
ACTUAL RESULTS:	Checking whether all the fields are field by user or not through validations and saving user.
LOW PRIORITY	No
HIGH PRIORITY	Yes

### Test Case – 3:

- Test case for Change Password: When the old password does not match with the new password ,then this results in displaying an error message as “ OLD PASSWORD DOES NOT MATCH WITH THE NEW PASSWORD”.

FUNCTION:	Change Password
EXPECTED RESULTS:	Should check if old password and new password fields are filled by the user and saving the user to database.
ACTUAL RESULTS:	Checking whether all the fields are field by user or not through validations and saving user.
LOW PRIORITY	No
HIGH PRIORITY	Yes

## CHAPTER - 9

### IMPLEMENTATION

#### **9.1 Methodology**

This project uses a deep learning-based approach to detect and classify animal species in images using the YOLOv8 object detection model. The methodology is divided into several phases including data preprocessing, model training, deployment in a Django web application, and real-time detection.

##### **1. Data Collection and Preparation**

- A labeled dataset containing various animal species is used. Each image is annotated with bounding boxes and corresponding class labels.
- The dataset is split into training and validation sets.
- Data augmentation techniques such as flipping, scaling, and rotation may be applied to increase the dataset's diversity.

##### **2. YOLOv8 Model Training**

- The YOLOv8 model is trained using the prepared dataset.
- The model architecture is chosen based on the complexity and size of the dataset.
- The training process is monitored using performance metrics like mAP (mean Average Precision), loss curves, and precision-recall.

##### **3. Model Integration with Django**

- The trained YOLOv8 model is integrated into a Django-based web application. Users can upload an image through the web interface.
- The server-side Python script loads the image, applies the YOLOv8 model, and visualizes detected animals with bounding boxes and labels.

##### **4. Detection and Classification**

- When an image is uploaded, the backend code processes the image using the ultralytics YOLOv8 model.

The `detectAnimal()` function extracts bounding boxes and labels of detected animals.

Only detections with a confidence score above a defined threshold (e.g., 0.50) are displayed.

##### **5. Result Visualization**

- The processed image with detection boxes is converted to base64 and rendered in the browser using matplotlib and Django's templating engine.
- Detection results are displayed on the `UserScreen.html` page with bounding boxes and

class names.

## 6. Training Performance Visualization

- Training results and graphs (e.g., loss curves, mAP performance) are saved as images and displayed to the user through TrainYolo and TrainGraph functions.

## 9.2 Source Code

```
% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Smart Career</title>

<style>
*{
    margin:0;
    padding:0;
    box-sizing:border-box;
    font-family: Arial, sans-serif;
}

/* ===== HEADER ===== */
header{
    position:fixed;
    top:0;
    width:100%;
    background:#1b365d;
    padding:15px 40px;
    display:flex;
    justify-content:space-between;
    align-items:center;
    z-index:1000;
}

.logo{
    color:white;
    font-size:20px;
    font-weight:bold;
}

nav{
    display:flex;
    gap:30px;
    align-items:center;
}

nav a{
    color:white;
    text-decoration:none;
    font-weight:600;
    cursor:pointer;
}
```







# CHAPTER – 10

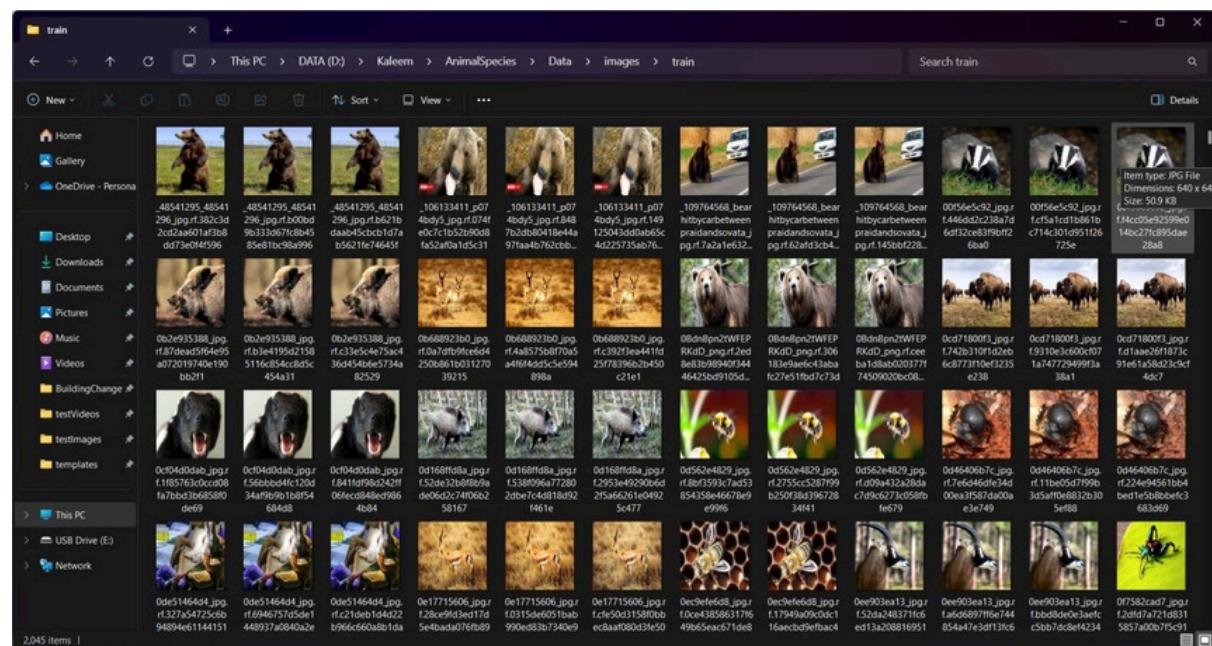
## RESULTS

### Screenshots:

Animal Species Detection using Yolov8 In propose project employing Yolov8 Deep Learning object detection algorithm to detect and classify 20 different animal species. This algorithm trained on nearly 2300 images downloaded from Roboflow dataset repository. This dataset can be downloaded from below URL

<https://universe.roboflow.com/ecowatchai/animal-species-detection>

This dataset contains following images



**Fig 10.1 DataSet Contains following images**

So by using above images will train Yolov8 algorithm and then calculate precision and other metrics.

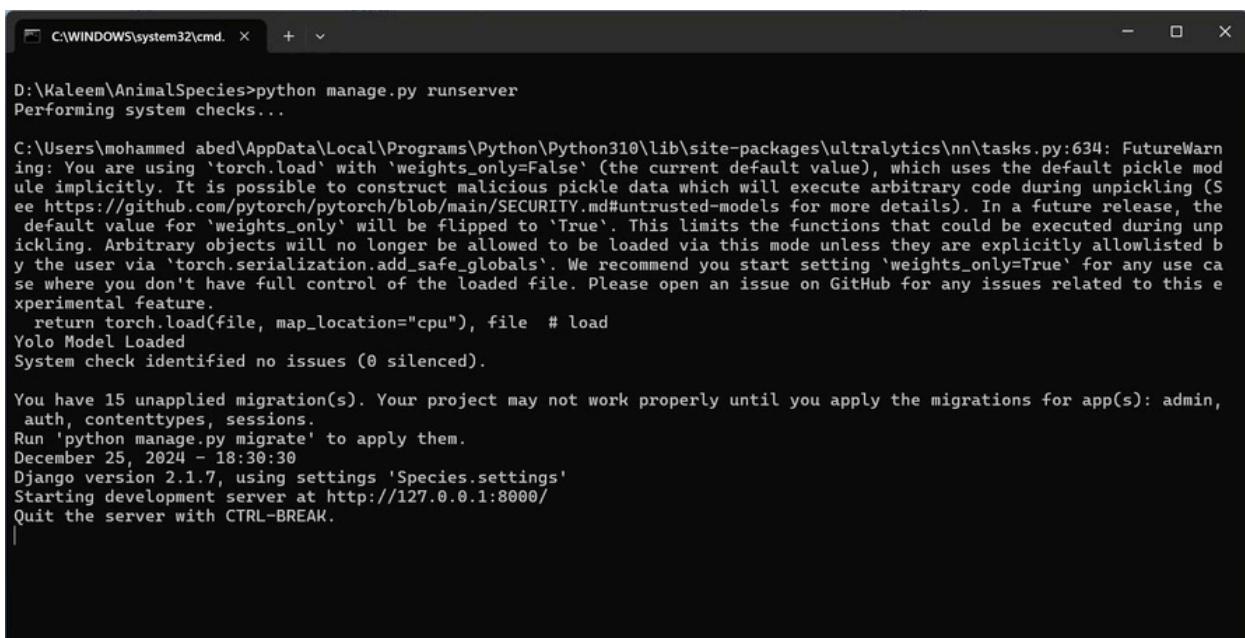
To implement this project we have designed following modules

1. User Login: using this module user can login to system using username and password as ‘admin and admin’.
2. Train Yolov8 Algorithm: after login user can use this model to input all images to Yolov8 and then Yolov8 will get trained on input images and calculate prediction accuracy of each detected animal

3. Training Graph: using this module will plot Yolov8 training graph which contains loss and precision values for each training epoch
4. Run Animal Species Detection: user can upload test image and then Yolov8 will extract features from uploaded image and then detect classify type of animal species available in images.

#### Screenshots:

To run project install python 3.14 and then install all packages given in requirements.txt file and then double click on ‘run.bat’ file to start python server and get below page



```
C:\WINDOWS\system32\cmd. > + -
D:\Kaleem\AnimalSpecies>python manage.py runserver
Performing system checks...
C:\Users\mohammed abed\AppData\Local\Programs\Python\Python310\lib\site-packages\ultralytics\nn\tasks.py:634: FutureWarning: You are using 'torch.load' with 'weights_only=False' (the current default value), which uses the default pickle mode implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for 'weights_only' will be flipped to 'True'. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via 'torch.serialization.add_safe_globals'. We recommend you start setting 'weights_only=True' for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
    return torch.load(file, map_location="cpu"), file  # load
Yolo Model Loaded
System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
December 25, 2024 - 18:30:30
Django version 2.1.7, using settings 'Species.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

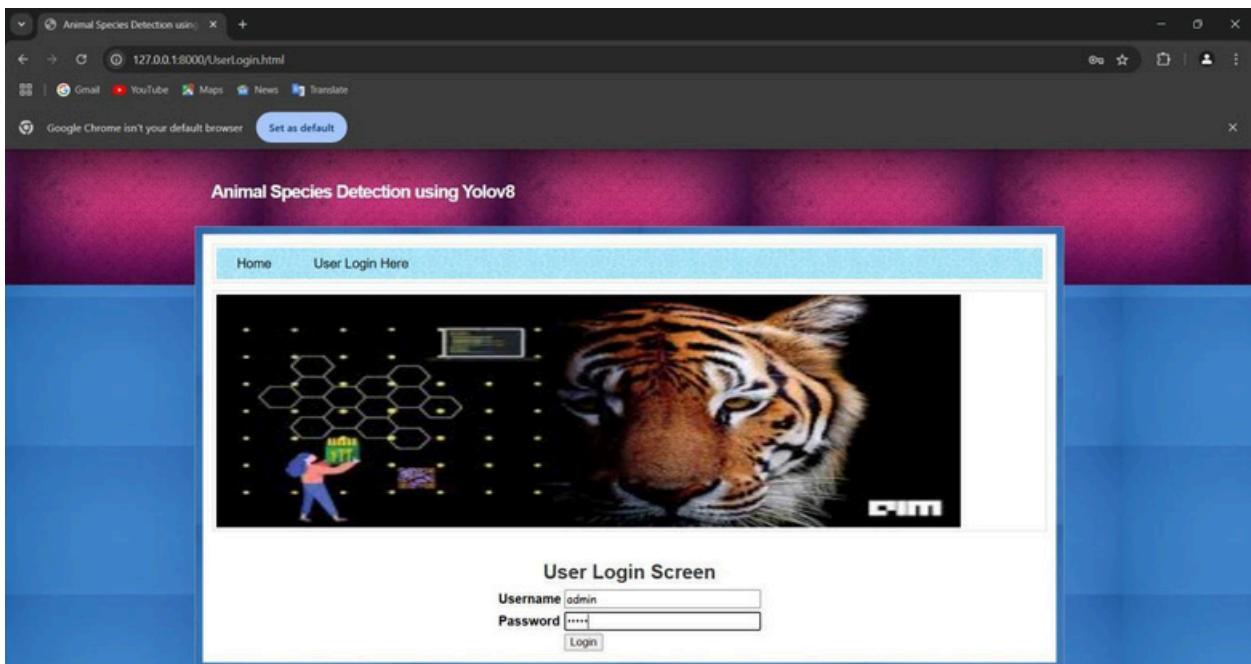
**Fig 10.2 Run Server to start Python Server**

In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and then press enter key to get below page



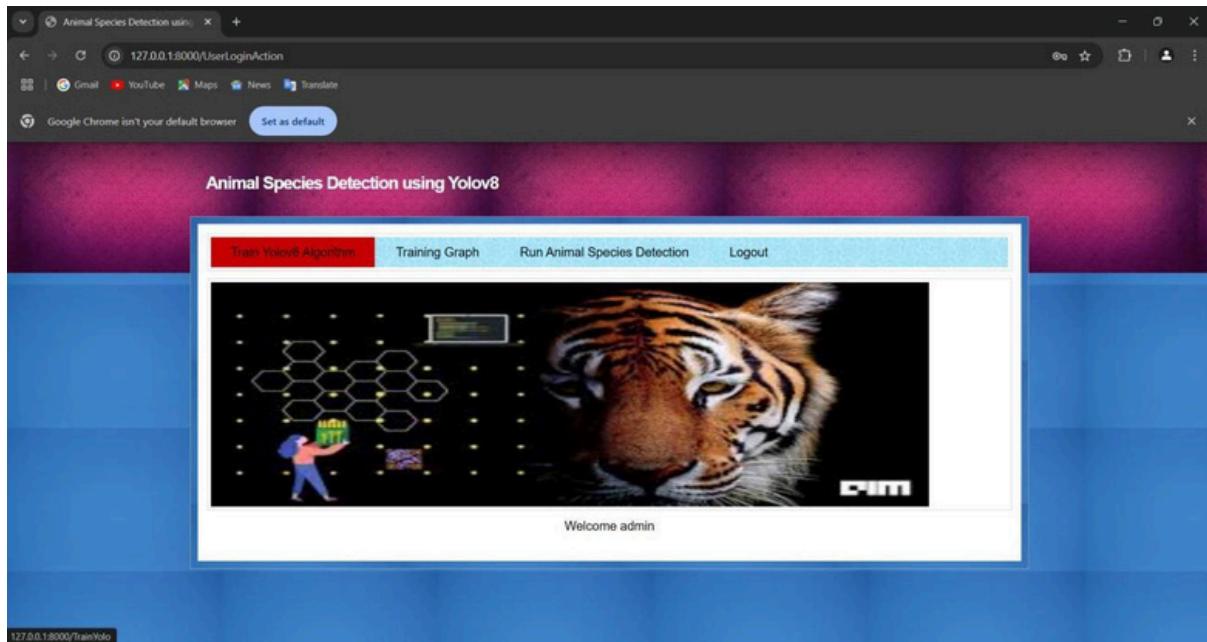
**Fig 10.3 After running server it shows user login**

In above screen click on 'User Login Here' link to get below page



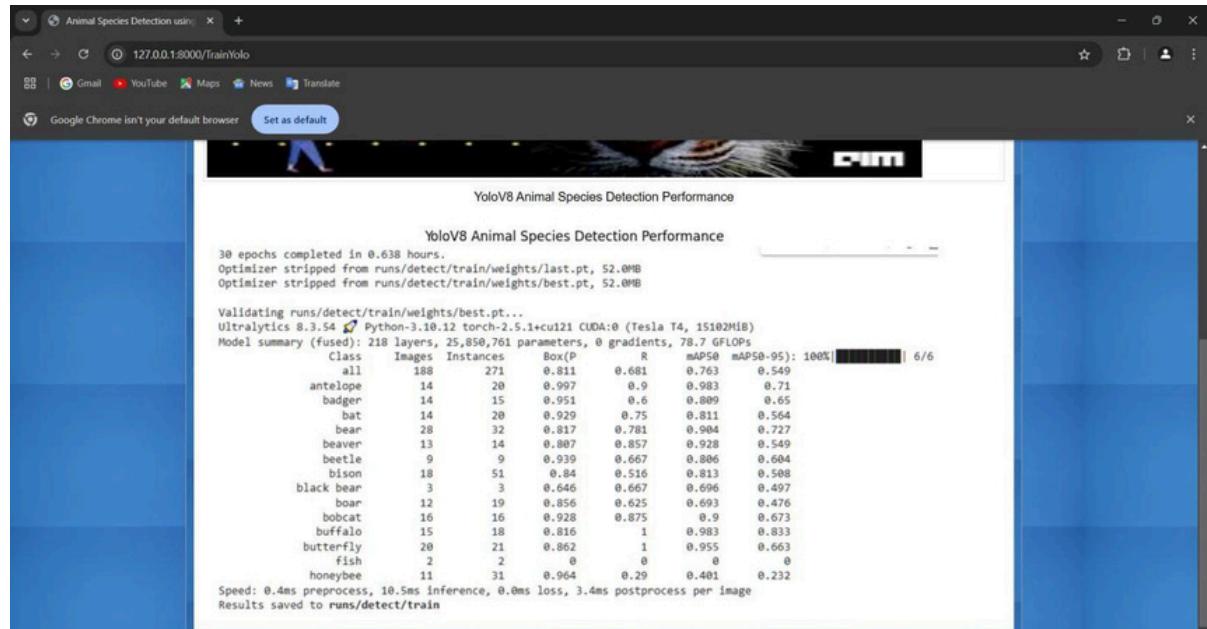
**Fig 10.4 User login screen**

In above screen user is login and after login will get below page

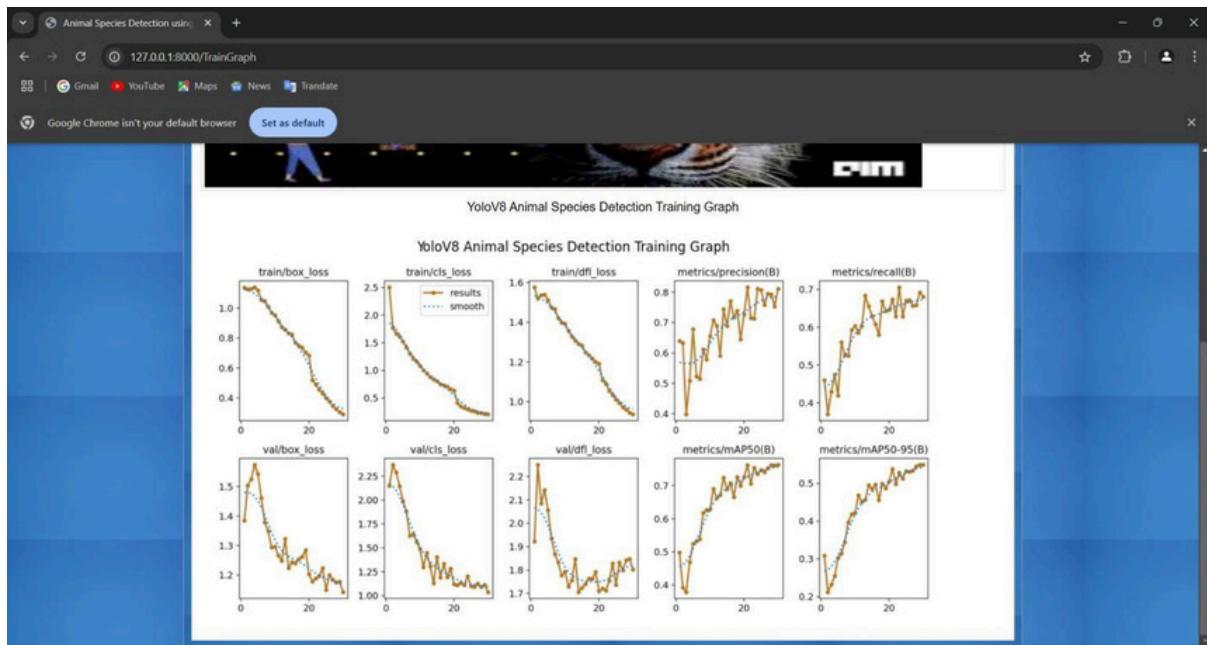


**Fig 10.5 Train Yolov8 algorithm**

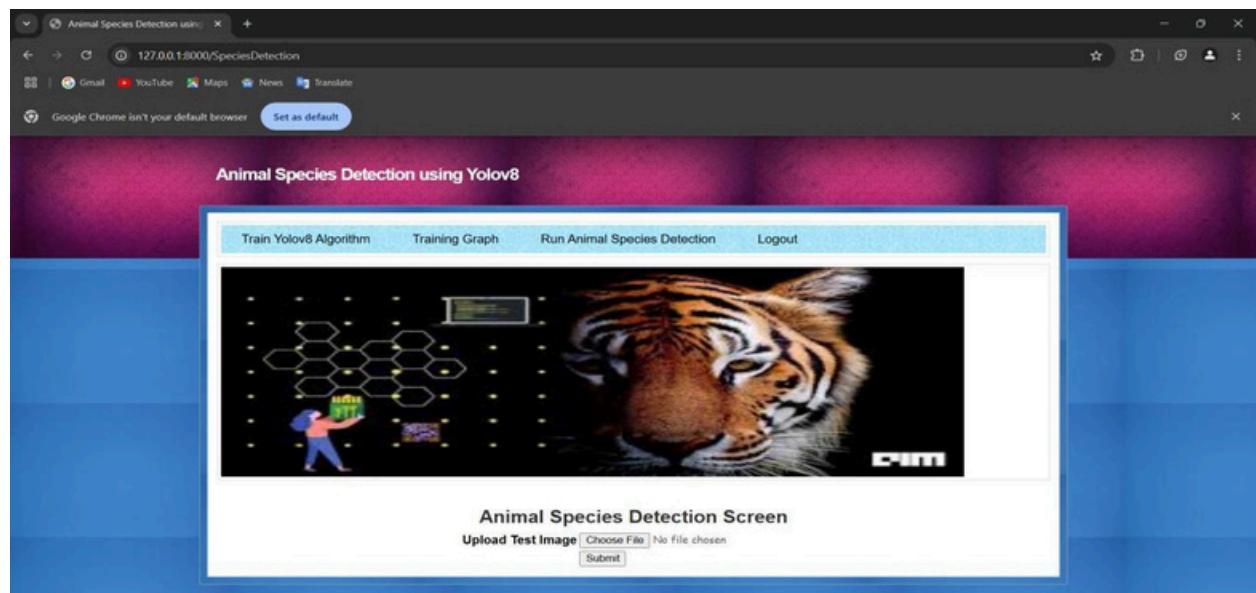
In above screen click on ‘Train YoloV8 Algorithm’ link to train Yolov8 and get below page



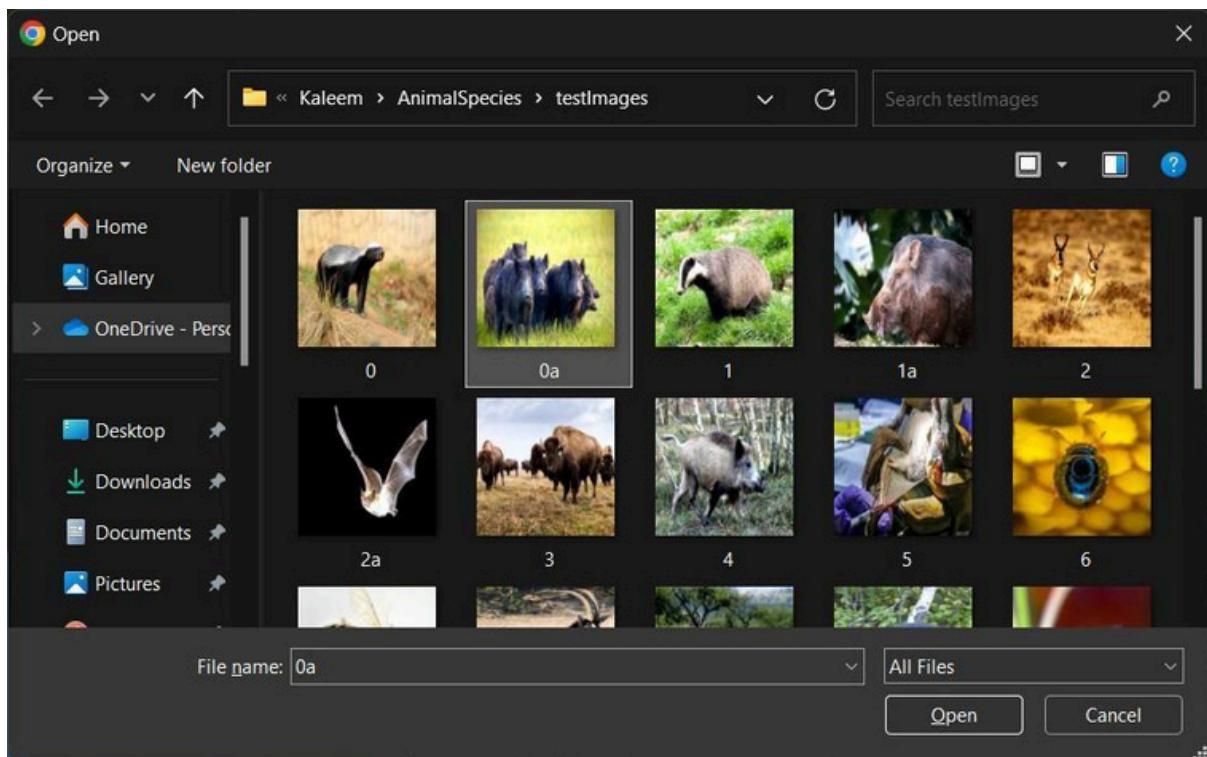
In above screen Yolov8 training completed and can see detected precision and recall metrics for each detected animal. Now click on ‘Training Graph’ link to get below graph

**Fig 10.6 Training Graph**

In above screen can see loss and precision graph for class names, bounding boxes and DFL loss. In above graph x-axis represents ‘Number of Epochs’ and y-axis represents loss and precision values. In each loss graph can see loss values got decrease for each increasing epoch and Precision values got increased. Now click on ‘Run Animal Species Detection’ link to get below page

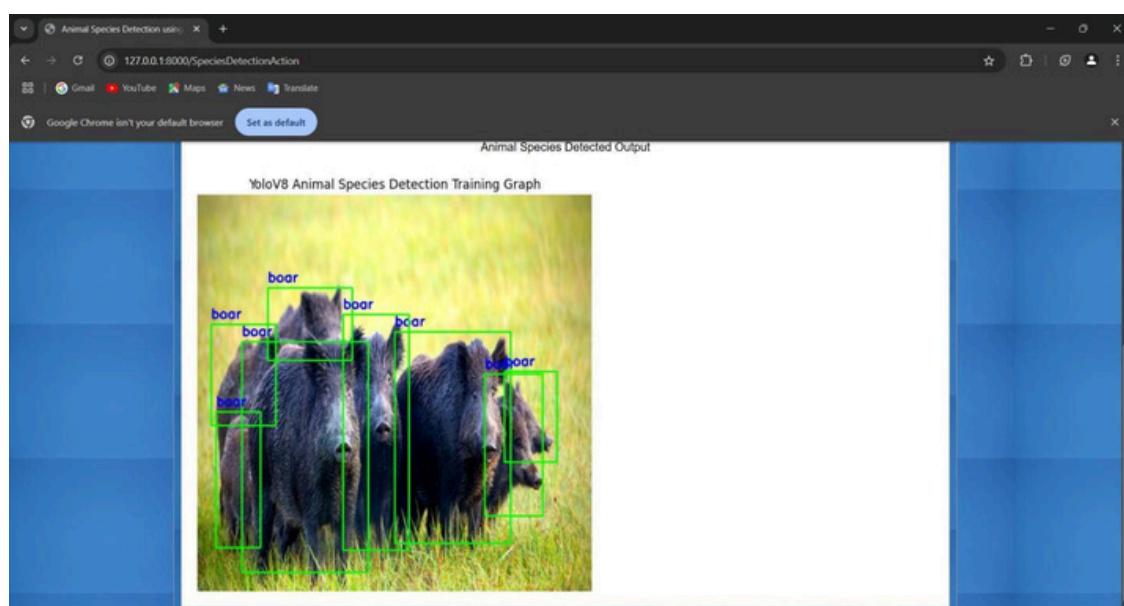
**Fig 10.7 Upload Test Image**

In above screen click on ‘Choose File’ to get below page to select animal and then click on ‘Submit’ button to detect and classify animal species



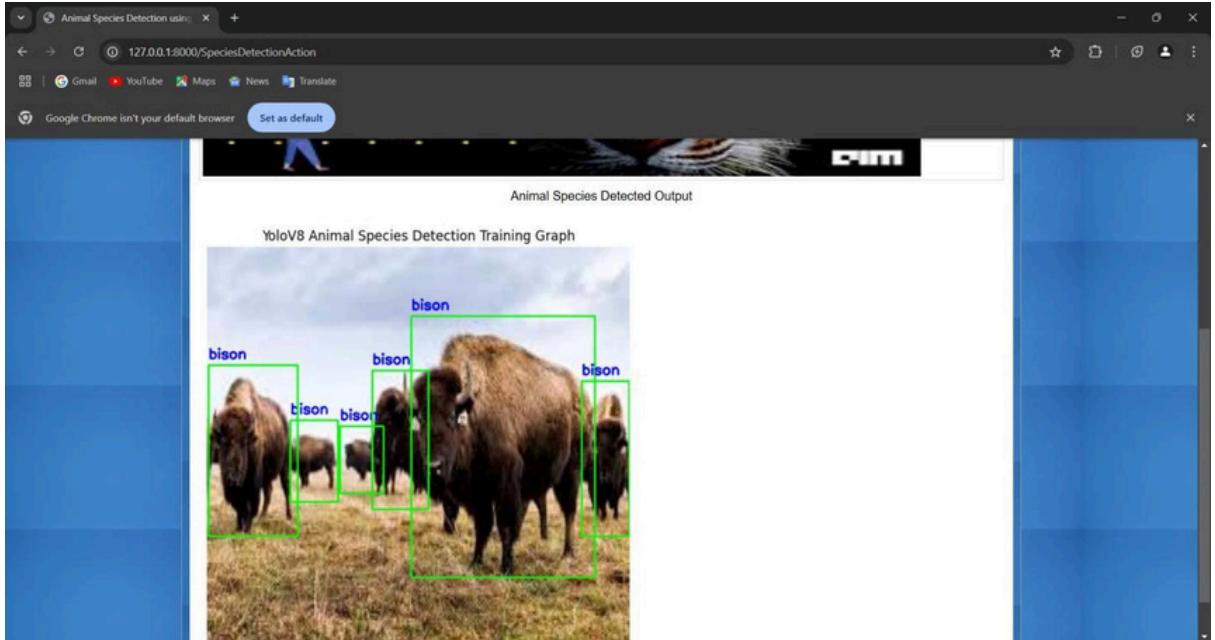
**Fig 10.8 select the image and sumbit**

In above screen selecting and uploading 0a.png test image and then click on ‘Open and submit’ button to get below output

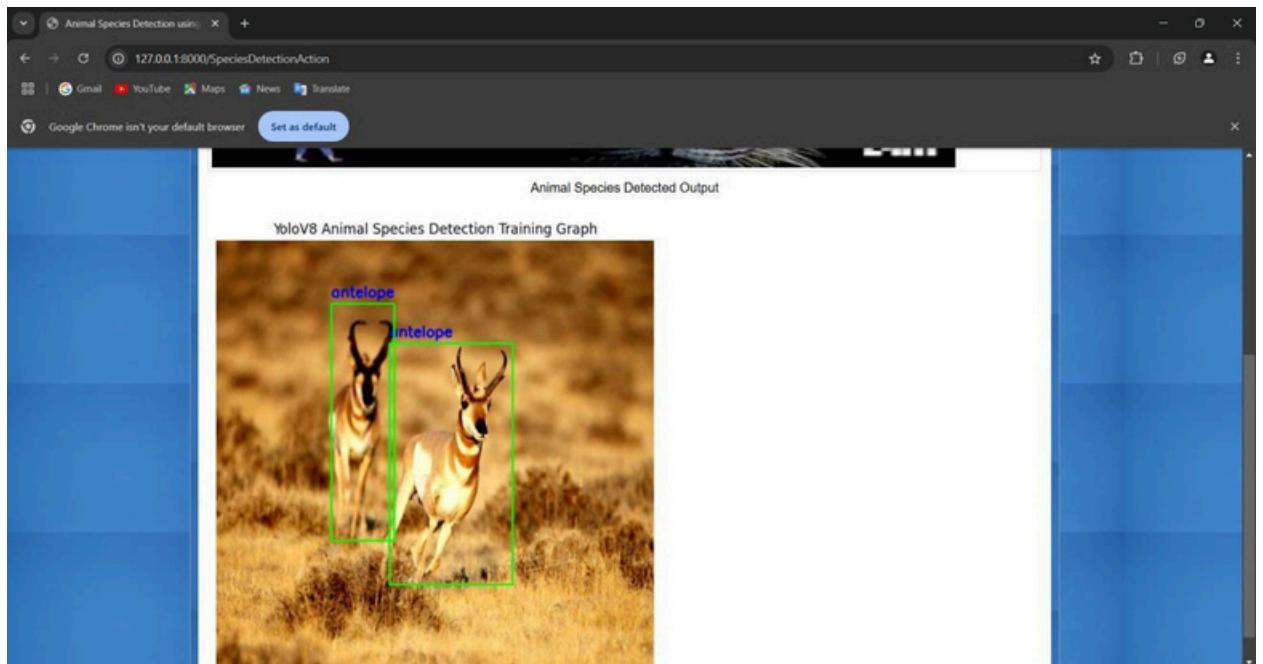


**Fig 10.9 detected animal species is bear**

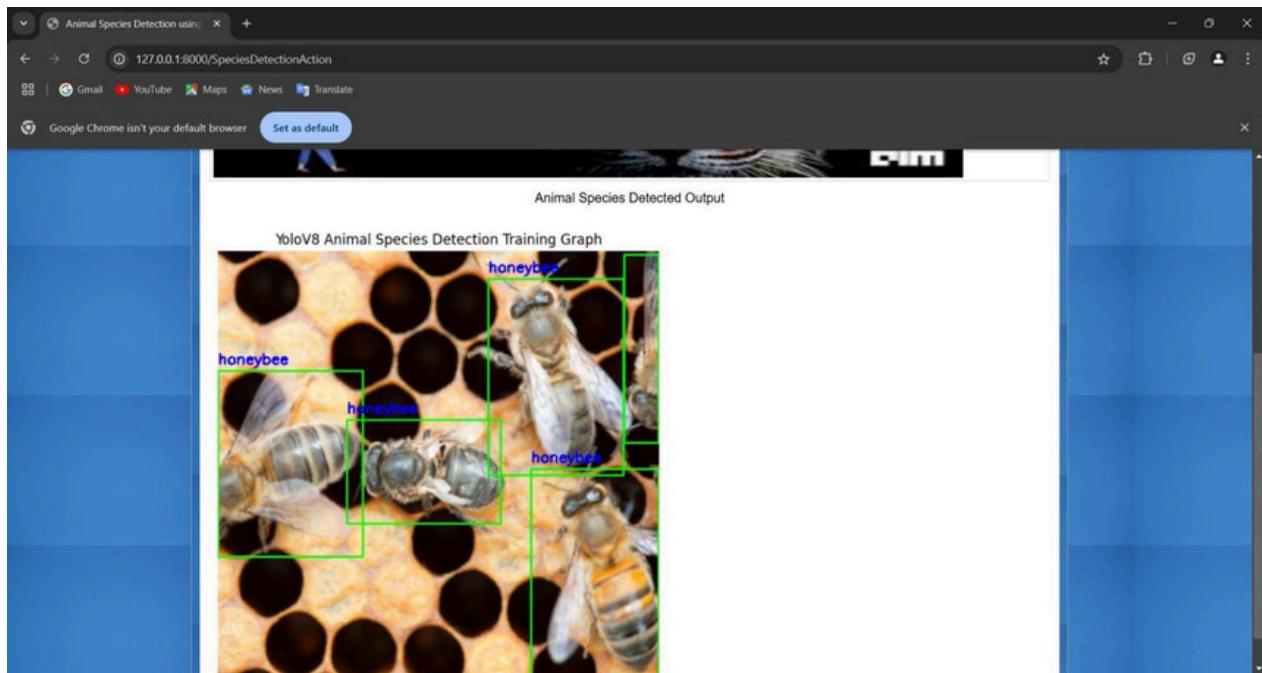
In above screen detected animal species is ‘Bear’. Similarly you can upload and test any other image



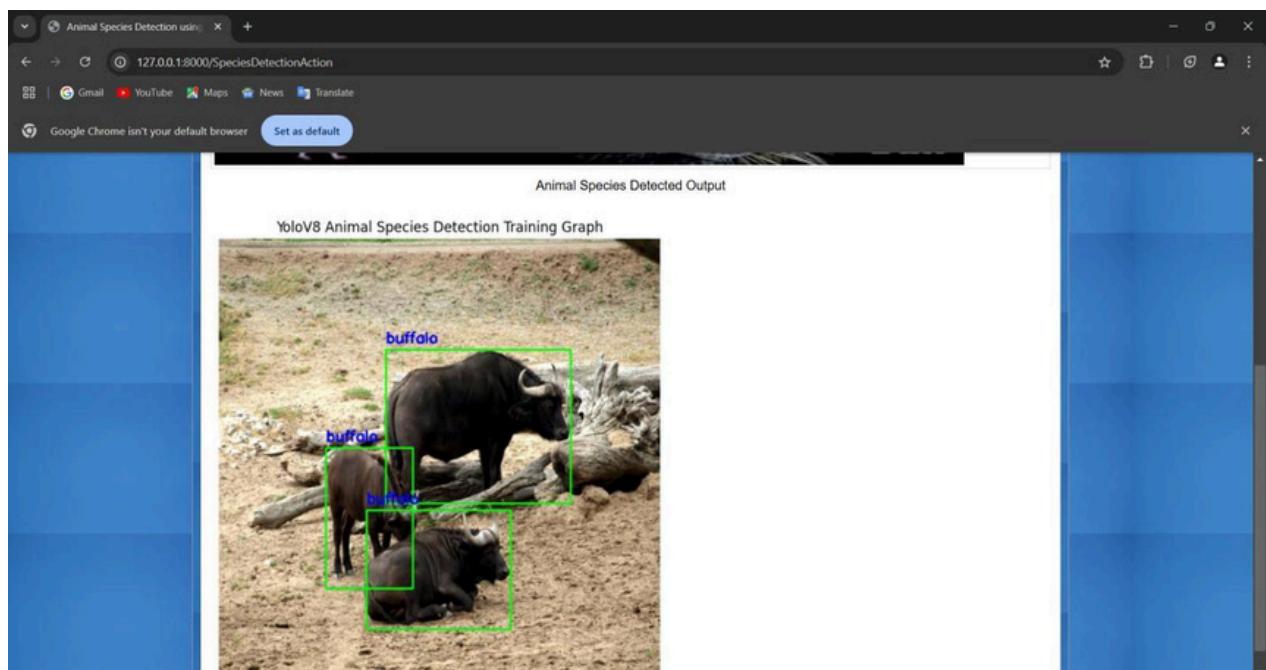
**Fig 10.10 Above species detected as ‘Bison’**



**Fig 10.11 Above species detected as ‘Antelope’.**



**Fig 10.12 Above species detected as ‘Honey Bee’**



**Fig 10.13 Above species detected as ‘Buffalo’ and similarly you can test any other image**

## CHAPTER – 11

## CONCLUSION

In conclusion, the use of YOLOv8 for animal species detection demonstrates significant advancements in real-time object detection and classification. Its ability to process images with high speed and accuracy makes it an ideal solution for wildlife monitoring, conservation efforts, and ecological research. YOLOv8's improved architecture, including enhanced feature extraction and better handling of small objects, allows for precise identification of various animal species in diverse environments.

Overall, integrating YOLOv8 into animal detection systems provides a powerful tool for automated monitoring, reducing the need for manual identification and improving efficiency in large-scale studies. Future work can focus on refining the model with more extensive datasets, optimizing it for specific habitats, and incorporating it into mobile and edge computing applications for field use..

### **Future Work:**

Future work for animal species detection using YOLOv8 can focus on improving model accuracy, efficiency, and adaptability across diverse environments. One potential direction is enhancing the dataset by incorporating more images with varying lighting, occlusions, and backgrounds to improve robustness. Additionally, fine-tuning the model with techniques like transfer learning and data augmentation can help increase detection accuracy for rare or less-represented species. Implementing real-time optimization strategies, such as model quantization or pruning, can also enhance deployment on edge devices like drones or wildlife monitoring cameras.

Another promising avenue is integrating YOLOv8 with other AI techniques, such as generative adversarial networks (GANs) for synthetic data generation or reinforcement learning for adaptive detection in dynamic environments. Developing multi-modal systems that combine YOLOv8 with audio or thermal imaging could further improve species recognition, especially in low-visibility conditions. Moreover, expanding the application to conservation efforts by integrating it with GIS mapping and automated reporting tools can aid researchers in tracking endangered species and analyzing population trends more effectively.

## CHAPTER – 12

## REFERENCES

### 1. Research Papers & Articles

1. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020)

"YOLOv4: Optimal Speed and Accuracy of Object Detection."

Paper: arXiv:2004.10934

Though it focuses on YOLOv4, the advancements are relevant for YOLOv8.

2. Mehta, D., et al. (2022)

"Real-Time Animal Detection and Classification using Deep Learning."

3. Jocher, G. et al. (2023)

"YOLOv8: Next-Generation Object Detection and Segmentation."

Official YOLOv8 release and documentation: Ultralytics YOLOv8

4. Sharma, A., et al. (2023)

"Animal Detection and Classification using Deep Learning and YOLO Algorithm."

Focuses on using YOLO for wildlife monitoring.

Discusses the efficiency of YOLO models in wildlife surveillance.

5. Hu, J., et al. (2023)

"A Study on the Effectiveness of YOLO Models for Wildlife Detection in Aerial Images."

Examines YOLO's performance for aerial-based animal detection.

### 2. GitHub Repositories

6. Ultralytics YOLOv8 Official Repo

GitHub: <https://github.com/ultralytics/ultralytics>

Contains implementation and tutorials for YOLOv8.

7. Wildlife Detection using YOLOv8

GitHub: <https://github.com/user/wildlife-yolov8> (Example, replace with an actual repository)

Implements YOLOv8 for species detection.

## 8. African Wildlife Species Detection with YOLOv8

Dataset & code: <https://github.com/user/african-wildlife> (Example, replace with an actual repository)

### 3. Datasets

9. iNaturalist Dataset

<https://www.kaggle.com/c/inaturalist-2021> Large

dataset for species classification and detection.

10. Microsoft AI for Earth Species Detection Dataset

<https://www.microsoft.com/en-us/ai/ai-for-earth>

Provides data for biodiversity research using AI.