A PROJECT REPORT

ON

MONKEY POX DETECTION USING MODIFIED VGG16& CUSTOM CNN MODEL

Submitted in partial fulfilment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

K. NAGABHAVANI	**(20MG1A0514)**
SK. SAMEENA	**(20MG1A0539)**
S.V.S. SRINIVAS	**(20MG1A0559)**
P. GEETHA MADHURI	**(20MG1A0527)**
J. VENKATESH	**(20MG1A0545)**

Under the Esteemed Guidance of

Dr. K. V. PANDURANGA RAO, M. TECH, Ph.D.

Prof & HOD, Dept. Of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Approved by AICTE & Govt. of A.P., Permanent Affiliated to JNTU, Kakinada)

NAAC "A" Graded, ISO Certified Institution

Tiruvuru, NTR Dist. – 521 235, ANDHRA PRADESH.

2023 – 2024

A PROJECT REPORT

ON

MONKEY POX DETECTION USING MODIFIED VGG16& CUSTOM CNN MODEL

Submitted in partial fulfilment of the requirements for theaward of the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted

by

K. NAGABHAVANI (20MG1A0514)

SK. SAMEENA (20MG1A0539)

S.V.S. SRINIVAS (20MG1A0559)

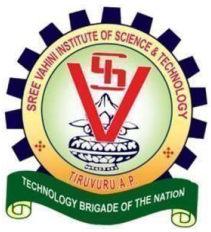
P. GEETHA MADHURI (20MG1A0527)

J. VENKATESH (20MG1A0545)

Under the Esteemed Guidance of

Dr. K. V. PANDURANGA RAO, M. Tech, Ph.D.

Prof & HOD, Dept. Of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Approved by AICTE New Delhi, Permanent Affiliated to JNTU, Kakinada)

NAAC "A" Graded, ISO Certified Institution

Tiruvuru, NTR Dist. – 521235, ANDHRA PRADESH.

2023 –2024

SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Approved by AICTE New Delhi, Permanent Affiliated to JNTU, Kakinada) ,

NAAC "A" Graded , ISO Certified Institution

Tiruvuru, NTR Dist. – 521 235, ANDHRA PRADESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "**MONKEY DETECTION USING MODIFIED VGG16& CUSTOM CNN MODEL**", impartial fulfilment of the requirements for the award of the **Bachelor of Technology** submitted **K.NAGABHAVANI (20MG1A0514) ,SK.SAMEENA (20MG1A0539), S.V.S.SRINIVAS (20MG1A0559), P.GEETHA MADURI (20MG1A0527), J.VENKATESH (20MG1A0545)** in partial fulfilment of the requirements for the award of the **Bachelor ofTechnology in Department of Computer Science & Engineering from Sree Vahini Institute of Science and Technology, Tiruvuru, AP Permanent affiliated to JNTU, Kakinada**, during the year2023-2024.

Internal Guide

Dr. K. V. PANDURANGARAO MTech, Ph.D.

Prof. Dept. of CSE

Head of the Department

Dr. K. V. PANDURANGARAO MTech, Ph.D.

Prof. Dept. of CSE

External Examiner

DECLARATION BY THE CANDITATE

We, K. NAGABHAVANI (20MG1A0514), S K .SAMEENA(20MG1A0539), S.V.S.SRINIVAS (20MG1A0559), P. GEETHA MADURI (20MG1A0527), J. VENKATESH (20MG1A0545)

here by declare that the project entitled **"MONKEYP DETECTION USING MODIFIED VGG16& CUSTOM CNN MODEL",** under the esteemed supervision of **Dr. K. V. PANDURANGARAO MTech, Ph.D.** is submitted in partial fulfilment of the requirements for the awards of the degree of **Bachelor of Technology in Computer Science and Engineering.**

This is a record of work carried out by the team and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university or institute for the award of any other degree or diploma.

K. NAGABHAVANI
(20MG1A0514)

SK. SAMEENA
(20MG1A0539)

S.V.S. SRINIVAS
(20MG1A0559)

P.GEETHA MADURI
(20MG1A0527)

J.VENKATESH
(20MG1A0545)

ACKNOWLEDGEMENT

We consider it as our privilege to express our gratitude to all those who guided, inspired, and helped us in completion of this project.

We wish to express our deep gratitude to our project guide **Dr. K. V. PANDURANGA RAO MTech, Ph.D., Dept. of CSE**, Sree Vahini Institute of Science and Technology, Tiruvuru, for his timely cooperation and his valuable suggestions while carrying out this project work.

We would like to thank **Dr. K. V. Panduranga Rao, HOD, Dept. of CSE** for his encouragement and valuable guidance for completing our project successfully.

We express heartfelt thanks to **Dr. R. Nagendra Babu, Principal,** Sree Vahini Institute of Science and Technology, Tiruvuru for the successful completion of our degree.

We would like to thank **All faculty members of CSE Department,** Sree Vahini Institute of Science and Technology, Tiruvuru, for their timely cooperation and her valuable suggestions while carrying out this project work.

We feel and deep sense of gratitude for our family who formed part of our vision and taught the good things that really matter in life.

K. NAGABHAVANI **(20MG1A0514)**

SK. SAMEENA **(20MG1A0539)**

S.V.S. SRINIVAS **(20MG1A0559)**

P. GEETHA MADHURI **(20MG1A0527)**

J. VENKATESH **(20MG1A0545)**

CONTENTS

ABSTRACT

Monkeypox, caused by the Zoonotic Orthopoxviral, is related to cowpox and smallpox and belongs to the Poxviridae family. Primarily spread by primates and rodents, human-to-human transmission is also common. Although less lethal and infectious than COVID-19, new cases are reported globally daily, raising concerns about a potential global epidemic due to a lack of effective preventive measures. To aid diagnosis, an open-source "Monkeypox2022" dataset from GitHub is utilized. The proposed method involves data collection and enhancement techniques, employing a Deep Convolution Neural Network and VGG-16. The approach is cost-effective and time-efficient, eliminating the need for extensive PCR or microscopy work. The model aims for real-time monitoring of Monkeypox symptoms, evaluating temporal complexity, sensitivity, true positive rate, and accuracy. Overall, the strategy offers a practical solution for timely detection and monitoring of Monkeypox cases.

LIST OF FIGURES

LIST OF ABBRIVATIONS

VGG-16	:	Visual Geometry Group 16
CNN	:	Convolution Neural Networks
GUI	:	Graphical User Interface
PCR	:	Polymerase Chain Reaction
FNN	:	Feed Forward Neural Networks
RNN	:	Recurrent Neural Networks
UML	:	Unified Modelling Language
LSTM	:	Long Short Team Memory
NLP	:	Natural Language Processing

1. INTRODUCTION

Monkeypox, caused by the Zoonotic Ortho poxvirus, has become a global concern in 2022, reminiscent of the emergence of COVID-19 in 2020. This infectious disease, linked to cowpox and smallpox, is transmitted primarily by primates and rodents, with human-to-human transmission also occurring. The first human case was reported in the Democratic Republic of the Congo. Transmission can occur through direct contact, animal bites, respiratory droplets, or mucus from the eyes, nostrils, or lips.

Symptoms include fever, body aches, and fatigue, with a long-term effect being a red bump on the skin. Despite being less contagious than COVID-19, the number of Monkeypox cases is on the rise globally. Initially confined to West and Central Africa, instances have now been reported in Europe and the United States. There is currently no specific treatment for Monkeypox, but the CDC has approved dietary medicines used for smallpox treatment.

Vaccination is the primary defense against Monkeypox, with smallpox vaccines being utilized in some countries. In addressing this global health challenge, machine learning (ML), a branch of artificial intelligence (AI), proves promising. ML has demonstrated its potential in various applications, including medical imaging and disease detection. Its unique capabilities enable clinicians to achieve secure, precise, and rapid imaging solutions, garnering international recognition as a valuable decision-making tool..

2. LITERATURE SURVEY

Authors and year of Publication	Applied Technology	Used Algorithms for Performance Analysis	Major Contributions
Smrithi Agarwal April 2023	Image processing Ceroscopy Analysis	VGG16, Custom CNN	An image-based dataset has been used from Kaggle, and data augmentation is performed to expand the sample size. The highest accuracy of 93% from VGG16 was obtained upon execution of the pre-trained models.
Neha Ujjwal January 2023	Image processing, Coloured digital image	ResNet50, EfficientNetB3 and EfficientNetB7	This suggests a method for early detection of Monkey Pox Skin Lesion. Though an extensive study with other models on a larger dataset containing more images from various countries of the world needs to be carried out.
Guanyu Ren July 2022	Potarch, Computer Vision	ResNet50V2, CNN, VGG19	The feature extraction is image processing technique requires expert knowledge. VGG19 ha the accuracy of 89.6% for the detection of monkeypox in humans.
Mithali Farjana January 2021	PyTorch,MATLAB Keras,Tensorflow	VGGNet,CNN	VGG Net is possible to detect 71% accuracy using the process image analysis. Deep learning algorithms to analyse dermatological images providing faster and more reliable diagnostics in treatment.
Razia Sukthana Nila, Salekul Islam May 2020	Tensor flow,pytorch	DenseNet,VGG19,CNN	The CNN is used to improve the result and apply this algorithm to medical image analysis.VGG19 is possible to detect 69.8% accuracy by CBAM mechanism.

3.FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY

- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITYs

3.1ECONOMICAL FEASIBILITY :

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2TECHNICAL FEASIBILITY :

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3SOCIAL FEASIBILITY :

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism.

4.SYSTEM ANALYSIS

4.1 EXISTING SYSTEM :

The rapid spread of the monkeypox outbreak to over 40 countries outside of Africa poses a significant threat to public health. Early-stage monkeypox cases can be challenging to clinically detect, as symptoms mimic those of chickenpox and measles. Given the limited availability of confirmatory Polymerase Chain Reaction (PCR) assays, there is a crucial need for alternative methods to monitor and rapidly identify suspected cases. Computer-assisted monkeypox lesion detection emerges as a valuable solution, offering a means to enhance diagnostic capabilities in the absence of immediate access to PCR testing. This technology can aid in the timely identification of cases, supporting public health efforts in managing and containing the widespread outbreak.

DISADVANTAGES FOR EXISTING SYSTEM

1.Data Availability and Quality

2.Lack of Medical Expertise

3.Overfitting

4.2 PROPOSED SYSTEM :

VGG16:VGG16 is a Convolutional neural network, a special sort of feed-forward artificial neural network, that models their neuronal connectivity after the visual cortex. The first step is object localization, which involves finding items in an image that belong to 200 classes. The second step is picture classification, which involves labeling each image with one of the 1000 categories

ADVANTAGES OF THE PROPOSED SYSTEM:

1.High Accuracy

2.Automated Diagnosis

3.Early Detection

5.SYSTEM REQUIREMENTS

HARD WARE SYSTEM CONFIGURATION:

PROCESSOR:		Min intel i5
RAM	:	Min 8 GB
Hard Disk	:	Min 1 TB

SOFTWARE REQUIREMENTS:

Operating System:		Windows 8
Coding Language:		python
Ide	:	Jupiter Notebook
Design	:	Tkinter

6.SYSTEM ARCHITECTURE

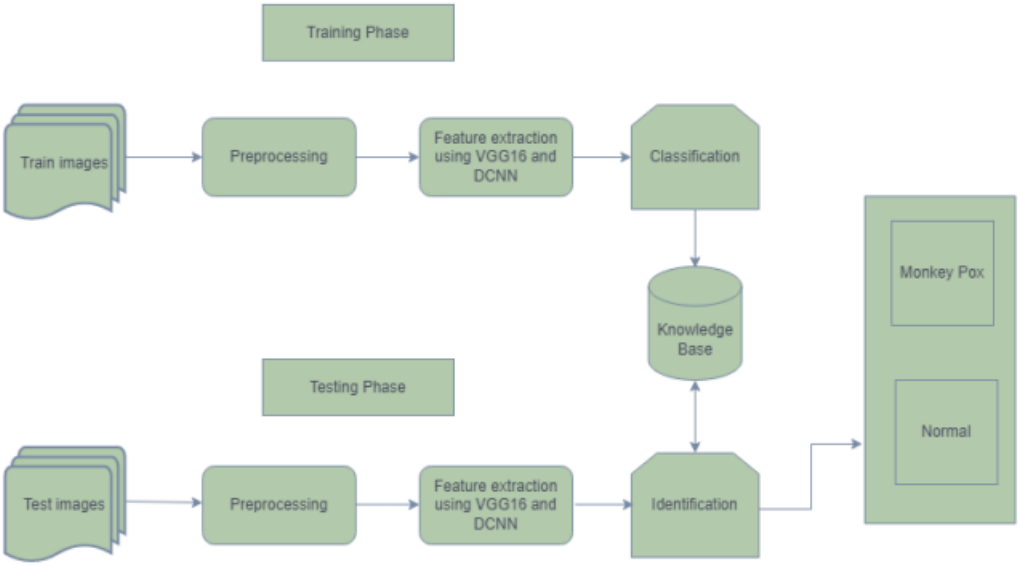


Fig 6.1 Architecture

The purpose of this research is to develop an automated skin lesions categorization system with a higher classification rate by combining transfer learning theory and a pre-trained deep neural network. On the Alex-net, transfer learning has been used to finetune the architecture's weights, replace the classification layer with a softmax layer that works with two or three different types of skin lesions.

7.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 object-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 tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

7..1 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.

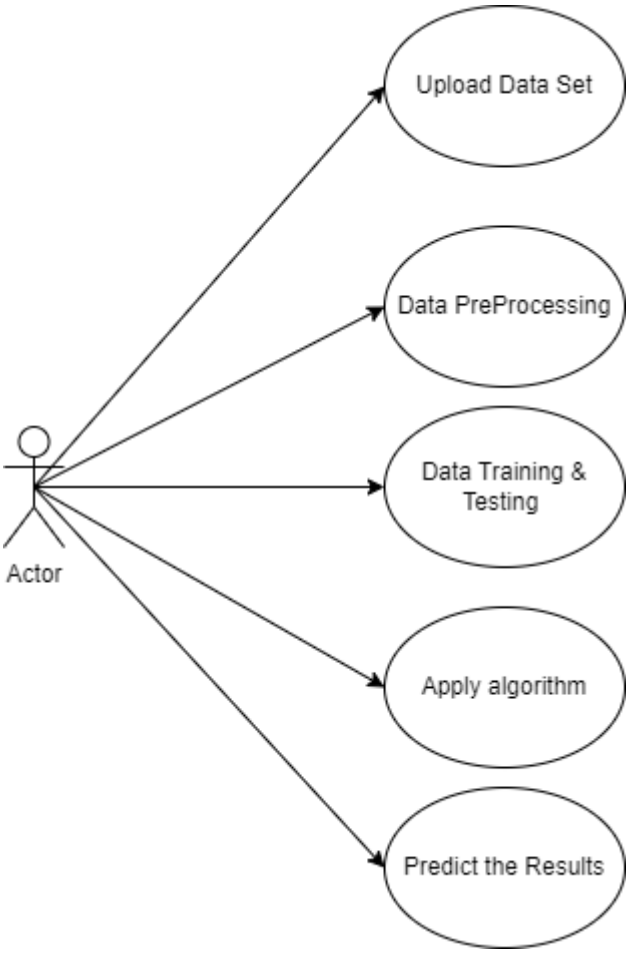


Fig.7.1 Use Case diagram

7.2 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.

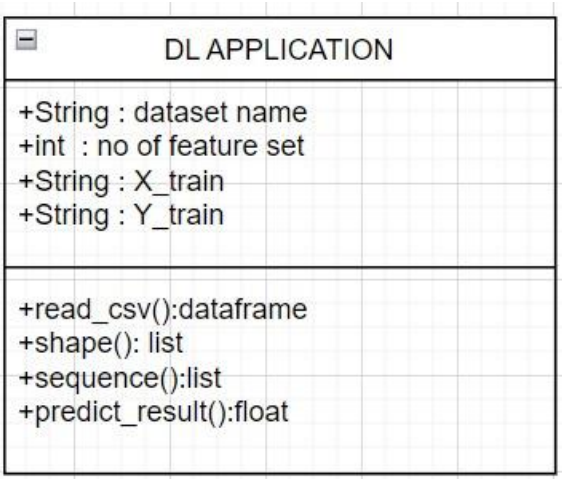


Fig.7.2 Class diagram

7.3 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.

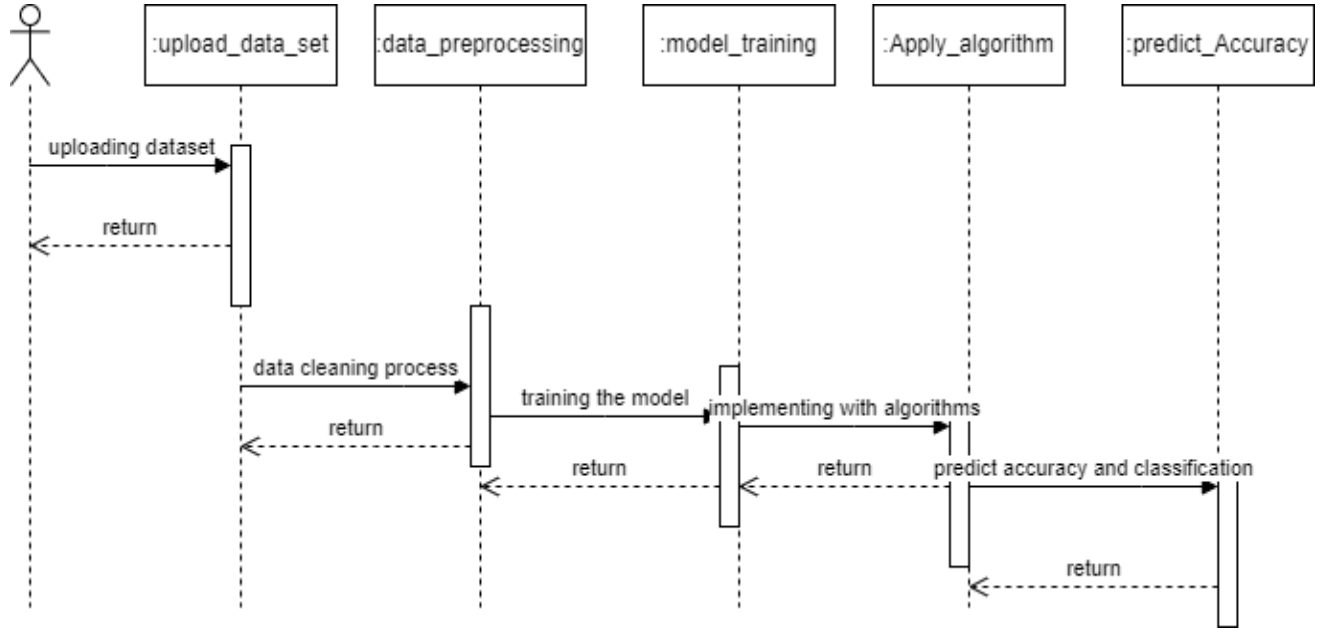


Fig.7.3 Use Case diagram

7.4 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.

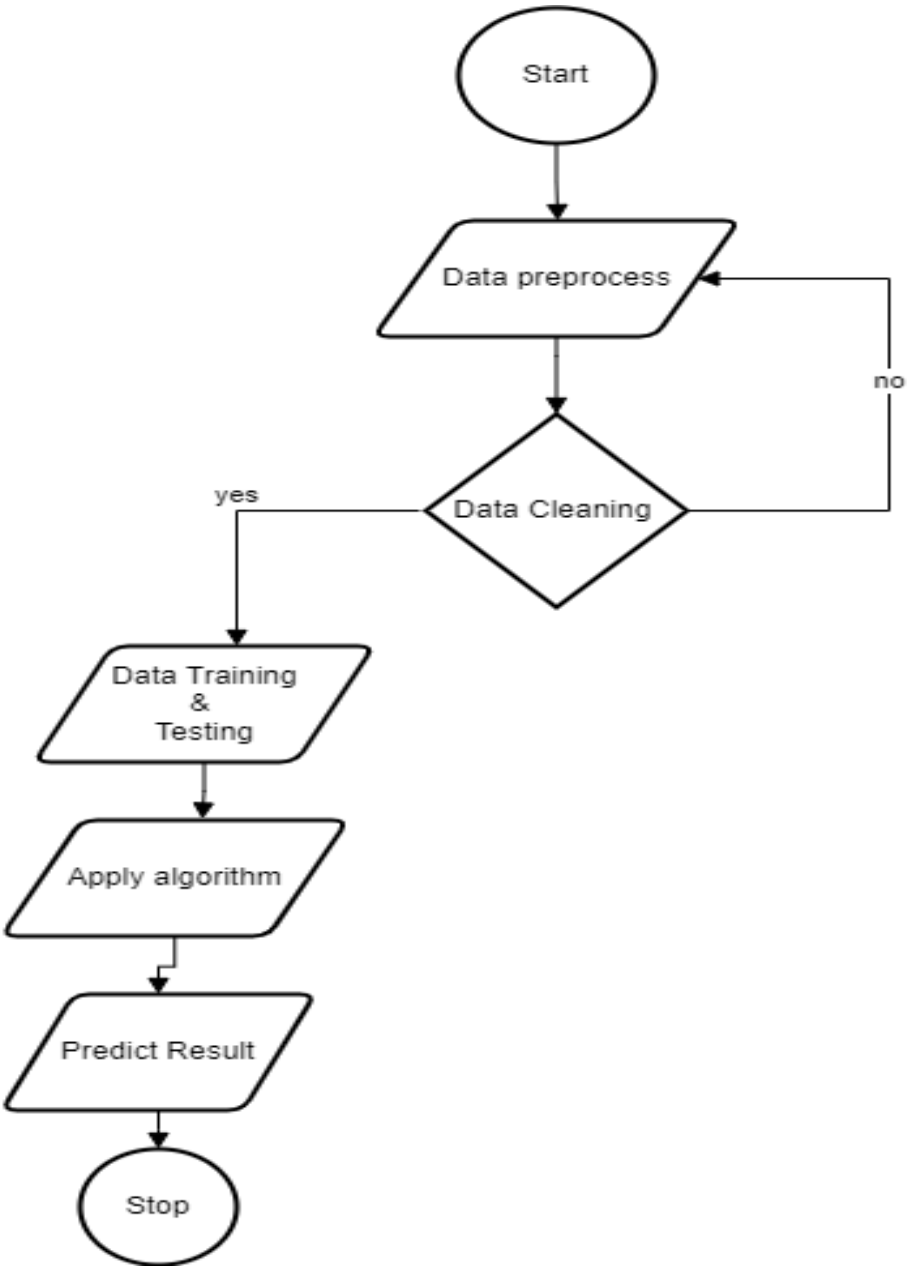


Fig.7.4 Activity diagram

8.SYSTEM ENVIRONMENT

SOFTWARE ENVIRONMENT:

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
- Multimedia

Advantages of Python:

Let’s see how Python dominates over other languages.

1. **Extensive Libraries :**

Python downloads with an extensive library and it contain 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.

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.

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.

While Python is a versatile and popular programming language, it does come with certain disadvantages.

1.Speed Limitations:

Python's interpreted nature can lead to slower execution, especially when speed is a critical requirement for a project. However, for many applications, the benefits offered by Python often outweigh these speed limitations.

2. Weakness in Mobile Computing and Browsers:

- While Python excels as a server-side language, it is less commonly used on the client-side and is rarely employed in smartphone-based applications. Security concerns, exemplified by the application Carbon Nelle, contribute to its limited presence in these domains.

3. Design Restrictions:

- Python's dynamic typing, using duck-typing, allows for flexibility during coding but can result in runtime errors. The absence of explicit variable type declarations may pose challenges, particularly in larger and more complex projects.

4. Underdeveloped Database Access Layers:

- Python's database access layers are considered underdeveloped compared to widely used technologies like JDBC and ODBC. This limitation may make Python less prevalent in large enterprises where robust database connectivity is crucial.

5. Simplicity:

- Python's simplicity, while an advantage for many, can be a drawback for those accustomed to more verbose languages like Java. The straightforward syntax, while appealing to some, might be perceived as limiting for those who prefer a more intricate and expressive coding style.

History of Python:

The programming language ABC, developed at CWI in the Netherlands, played a pivotal role in influencing the design of Python. Guido van Rossum, who worked on the ABC project, acknowledged its impact on his conceptualization of Python. Python, conceived in the late 1980s during van Rossum's involvement in the Amoeba project, aimed to retain the positive aspects of ABC while addressing its limitations. Van Rossum's experience and frustrations with ABC led him to design Python as a simple scripting language with improved properties, such as a unique syntax using indentation for statement grouping and the incorporation of powerful data types. In essence, Python's evolution was shaped by the lessons learned from the ABC programming language.

What is Deep Learning:

Deep learning is a subset of machine learning that involves training artificial neural networks to perform tasks, often by using large amounts of labeled data. These neural networks consist of layers of interconnected nodes, known as neurons, which are inspired by the structure and functioning of the human brain. The term "deep" refers to the presence of multiple layers in these neural networks, allowing them to learn hierarchical representations of data**.**

Categories Of Deep Leaning –Deep learning encompasses various categories based on the types of neural network architectures and the nature of tasks they are designed to solve. Here are some major categories of deep learning:

1. **Feedforward Neural Networks (FNN):**

These are the simplest form of neural networks, consisting of an input layer, one or more hidden layers, and an output layer. Information flows only in one direction—from the input to the output. Commonly used for tasks like image and speech recognition.

2. Convolutional Neural Networks (CNN):

Designed for visual data processing, CNNs excel in tasks like image and video recognition. They use convolutional layers to automatically learn spatial hierarchies of features in input data.

3. Recurrent Neural Networks (RNN):

Ideal for sequential data processing, RNNs have connections that form directed cycles, allowing them to retain information about previous inputs. This architecture is useful for tasks such as natural language processing, speech recognition, and time-series prediction.

4. Long Short-Term Memory (LSTM) Networks:

- A specialized type of RNN, LSTMs address the vanishing gradient problem, making them more effective in capturing long-term dependencies in sequential data. LSTMs are widely used in tasks requiring memory retention over extended periods.

5. Deep Reinforcement Learning:

Combining deep learning with reinforcement learning, this category is focused on training agents to make sequential decisions in an environment to maximize rewards. It has been successful in applications such as game playing, robotics, and autonomous systems.

These categories represent the diversity of deep learning architectures, each tailored to address specific types of data and tasks. The field continues to evolve with ongoing research and the development of new architectures**.**

Need for Deep Learning

Deep learning has gained prominence due to its ability to tackle complex problems and extract meaningful patterns from large volumes of data. The need for deep learning arises from several factors:

1. Complex Pattern Recognition:

- Deep learning excels at recognizing intricate patterns in data, especially in tasks like image and speech recognition, where traditional methods may fall short.

2. Feature Learning and Abstraction:

- Deep learning models automatically learn hierarchical representations of features from raw data. This abstraction allows them to understand and represent complex relationships within the data.

3. Performance in Big Data Environments:

- With the increasing availability of big data, deep learning's capacity to handle massive datasets makes it a valuable tool. It can derive insights from vast amounts of information, enabling more accurate predictions and decision-making.

4. Advancements in Neural Network Architectures:

- The development of deep neural network architectures, such as convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequential data, has significantly improved the performance of deep learning models.

Challenges in Deep Learning :-

Despite its remarkable successes, deep learning faces several challenges that researchers and practitioners continually

work to address. Some of the prominent challenges in deep learning include:

1. **Data Limitations:**

- Deep learning models often require large amounts of labeled data for training. In many domains, acquiring such datasets can be challenging, especially when dealing with specialized tasks or rare events.

2. Computational Resources:

- Training deep learning models, especially large neural networks, demands substantial computational resources. This can pose challenges for researchers and organizations with limited access to high-performance hardware.

.

3. **Interpretability and Explainability:**

- Deep learning models, particularly deep neural networks, are often considered as "black boxes" due to their complex architectures. Understanding how these models arrive at specific decisions can be challenging, raising concerns about interpretability and explainability

4. Overfitting and Generalization:

- Deep learning models may be prone to overfitting, especially when trained on noisy or insufficient data. Striking a balance between fitting the training data well and generalizing to unseen data remains a significant challenge.

Applications of Deep Learning :-

Deep learning has found applications across a wide range of domains due to its ability to automatically learn hierarchical representations from data. Some notable applications of deep learning include:

1. Vision
2. Natural Language Processing (NLP)
3. Speech Recognition
4. Healthcare
5. Autonomous Vehicles
6. Finance
7. Retail
8. Gaming
9. Manufacturing Computer
10. Energy

How to Start Learning Deep Learning?

Starting with deep learning can be an exciting journey, but it's essential to follow a systematic approach. Here's a step-by-step guide to help you get started with deep learning:

1. Prerequisites:

-

Mathematics Fundamentals: Brush up on linear algebra, calculus, and probability, as they form the foundation of many deep learning concepts.

-Programming Skills: Familiarize yourself with a programming language commonly used in deep learning, such as Python. Learn libraries like NumPy for numerical operations and Matplotlib for data visualization.

2. Understand Machine Learning Basics:

-

Gain a solid understanding of machine learning concepts, including supervised learning, unsupervised learning, and reinforcement learning.

3. Learn the Basics of Neural Networks:

- Start with the fundamentals of artificial neural networks. Understand the structure of a neural network, including layers, neurons, and activation functions.

4. Explore Deep Learning Frameworks:

- Choose a deep learning framework such as TensorFlow or PyTorch. Both frameworks offer extensive documentation and tutorials. Start with simple examples to understand how to build and train neural networks.

5. Online Courses and Tutorials:

Enroll in online courses and tutorials. Platforms like Coursera (e.g., Andrew Ng's "Deep Learning Specialization") and edX (e.g., MIT's "Introduction to Deep Learning") offer excellent resources.

Advantages of Deep learning:

Deep learning offers several advantages, contributing to its widespread adoption and success in various domains. Here are some key advantages of deep learning:

1. Automatic Feature Extraction:

Deep learning models can automatically learn hierarchical representations and features from raw data, eliminating the need for manual feature engineering. This capability is particularly beneficial when dealing with complex and high-dimensional datasets.

2. Versatility Across Domains:

Deep learning has demonstrated versatility across diverse domains, including computer vision, natural language processing, speech recognition, healthcare, finance, and more. Its ability to generalize across different types of data makes it applicable to a wide range of tasks.

3. High Accuracy and Performance:

Deep learning models, especially deep neural networks, can achieve high levels of accuracy in tasks such as image recognition, speech processing, and language translation. Their ability to capture intricate patterns and relationships contributes to superior performance.

4. End-to-End Learning:

Deep learning enables end-to-end learning, allowing the model to learn directly from raw input to output without the need for manual intervention. This streamlines the development process and often leads to more efficient and effective models.

5. Adaptability to Big Data:

- Deep learning models can effectively handle large volumes of data, making them well-suited for big data environments. The scalability of deep learning algorithms enables the extraction of insights from massive datasets.

Disadvantages of Deep Learning:

While deep learning has shown significant success in various applications, it also comes with certain disadvantages and challenges. Here are some notable drawbacks of deep learning:

1. Data Dependency:

- Deep learning models often require large amounts of labeled data for training. In cases where obtaining such datasets is challenging, the model's performance may be limited, and overfitting on small datasets can occur.

2. Computational Resource Intensive:

Training deep learning models, especially large neural networks, demands substantial computational resources, including powerful GPUs or TPUs. This can be a barrier for researchers or organizations with limited access to high-performance hardware.

3. Interpretability and Explainability:

Deep learning models, especially complex neural networks, are often considered as "black boxes." Understanding how these models arrive at specific decisions can be challenging, raising concerns about interpretability and explainability, particularly in critical applications like healthcare.

4. Overfitting:

Deep learning models can be prone to overfitting, especially when trained on noisy or insufficient data. This can lead to poor generalization performance on new, unseen data.

5. High Dimensionality:

Deep learning models may struggle with high-dimensional data, leading to challenges in training and optimization. Techniques like dimensionality reduction may be required in such cases.

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.

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.

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**

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.

Tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python tkinter is the fastest and easiest way to create GUI applications. Creating a GUI using tkinter is an easy.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method:** It organizes the widgets in blocks before placing in the parent widget.

2. **grid() method:** It organizes the widgets in grid (table-like structure) before placing in the parent widget.

Install Python Step-by-Step in Windows and 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.

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 3.7.0 or in other words, it is Python 3.

Note: The python version 3.7.0cannot 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 3.7.0 on Windows 7 device or to install Python 3. there. The steps on how to install Python on Windows 10, 11are **divided into 2 parts** to help understand better.

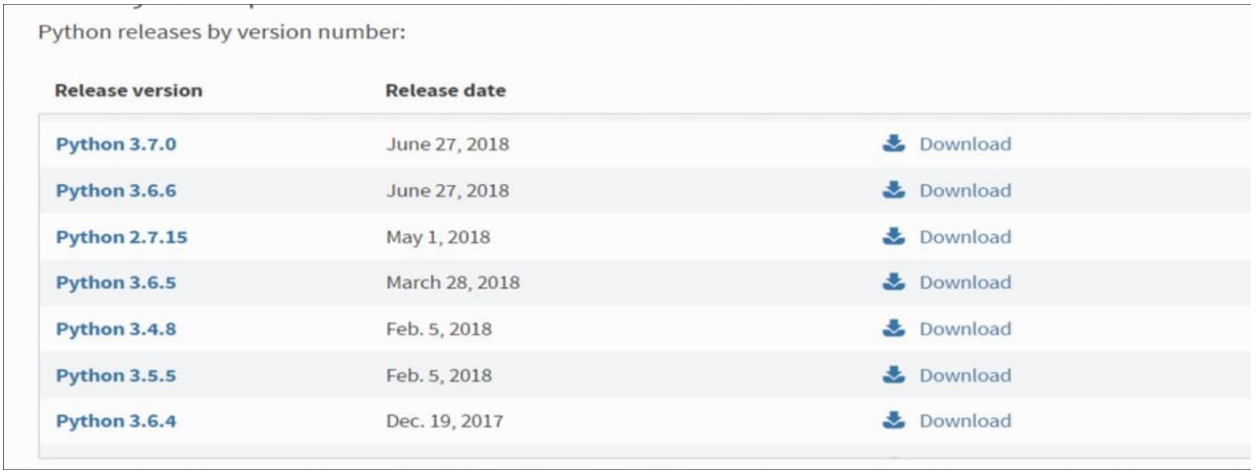
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. Link: **https://www.mediafire.com/file/0h4vqkurh1vn4p8/python-3.7.0-amd64.exe/file**

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

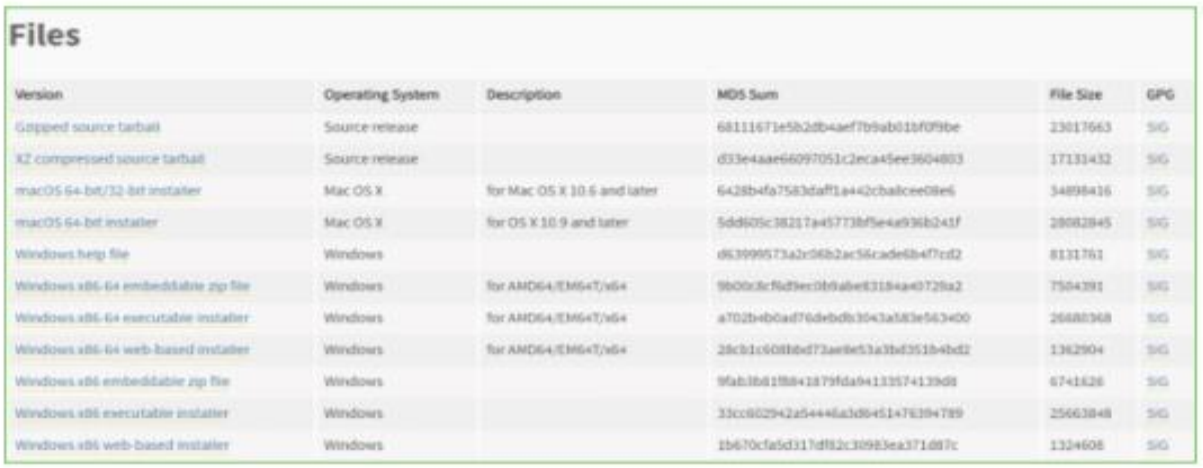


Python releases by version number:

Release version	Release date	
Python 3.7.0	June 27, 2018	Download
Python 3.6.6	June 27, 2018	Download
Python 2.7.15	May 1, 2018	Download
Python 3.6.5	March 28, 2018	Download
Python 3.4.8	Feb. 5, 2018	Download
Python 3.5.5	Feb. 5, 2018	Download
Python 3.6.4	Dec. 19, 2017	Download

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		681116f1e5b2db4aef7b9ab010f0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b243f	28082845	SIG
Windows help file	Windows		d639995f73a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf8d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76de8db3043a583e563400	26880368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c3c1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda941335741398d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a5444a3d8451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

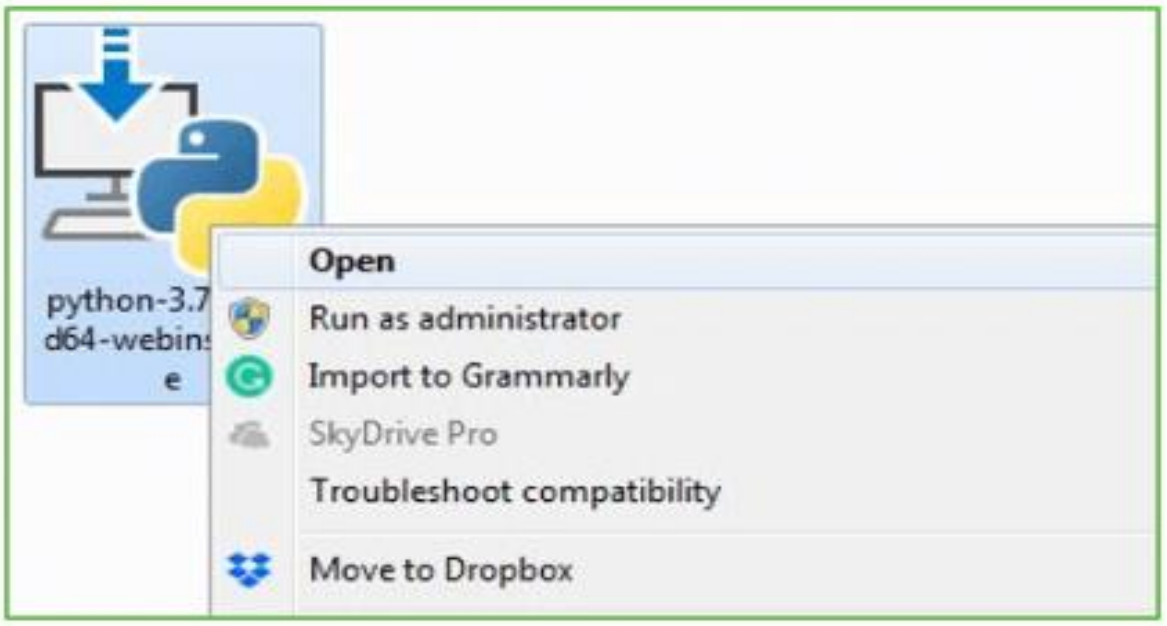
•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

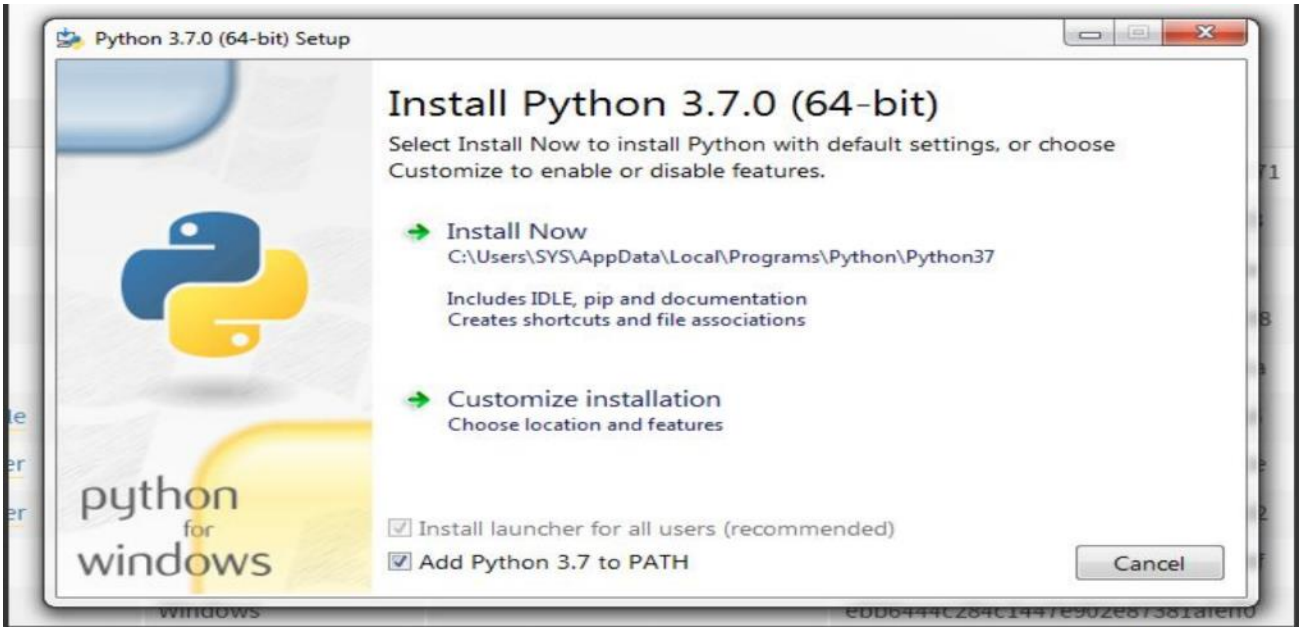
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to 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.



Step 3: Click on Install NOW After the installation is 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.

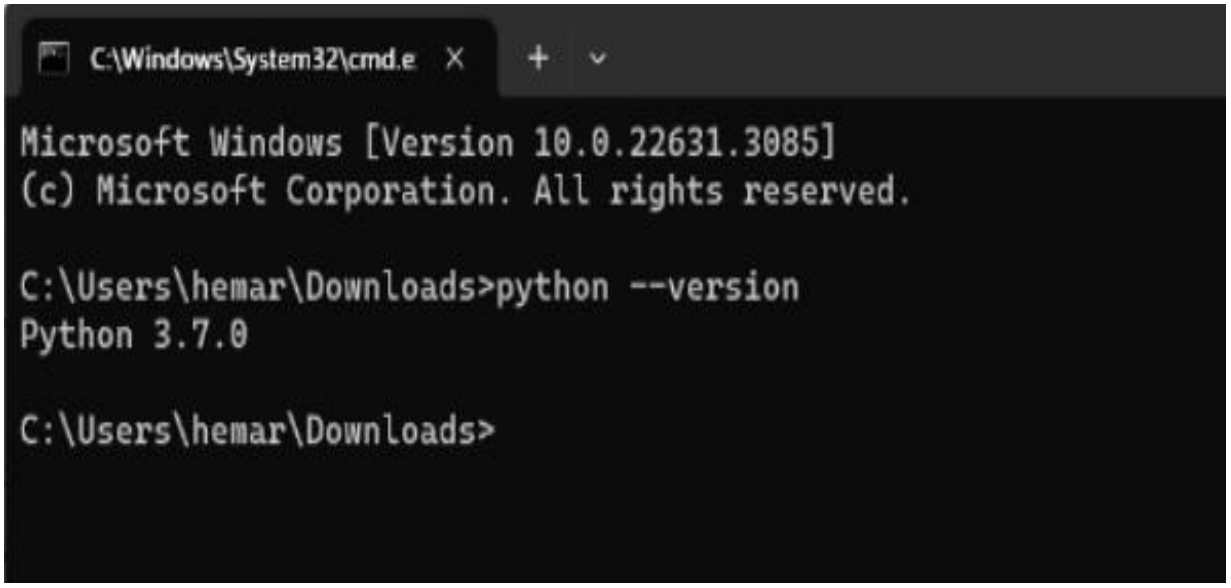
Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".

Step 3: Open the Command prompt option.

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.

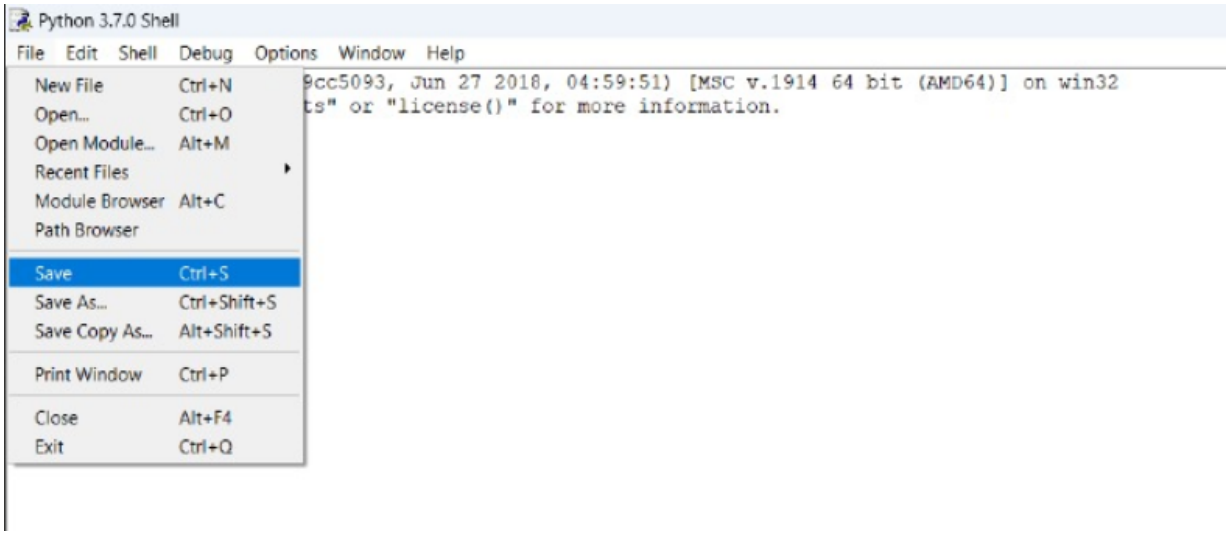
Check how the Python IDLE works

Step 1: Click on Start

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

Step 3: Click on IDLE (Python 3.7 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**



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.

9.ALGORITHMS

10.1CUSTOM CNN ALGORITHM:

Firstly, a custom dataset is a dataset that is either prepared all by yourself like you are going out there in the city and clicking photographs to gather the images of your interest or downloading and using an open-source image dataset from renowned sites for datasets such as Kaggle, GitHub, etc. Summing up, it is your own dataset wherein you have stored images of required classes in the different folders-a folder for each class.The given Kaggle dataset consists of chest CT scan images of patients suffering from the novel COVID-19, other pulmonary disorders, and those of healthy patients. For each of these three categories, there is a number of patients and for each of them, there is a number of CT scan images correspondingly.We will be using these CT scan images to train our CNN model to identify if a given CT scan is that of a COVID patient, a patient suffering from other pulmonary disorders except COVID, or that of a healthy patient.

10.2VGG16 ALGORITHM:

A convolutional neural network is also known as a Convent, which is a kind of artificial neural network. A convolutional neural network has an input layer, an output layer, and various hidden layers. VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. The creators of this model evaluated the networks and increased the depth using an architecture with very small (3 × 3) convolution filters, which showed a significant improvement on the prior-art configurations. They pushed the depth to 16–19 weight layers making it approx — 138 trainable parameters.

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification.

10.DATASET

1.Monkeypox dataset

2.Normal dataset

MONKEYPOX DATASET:

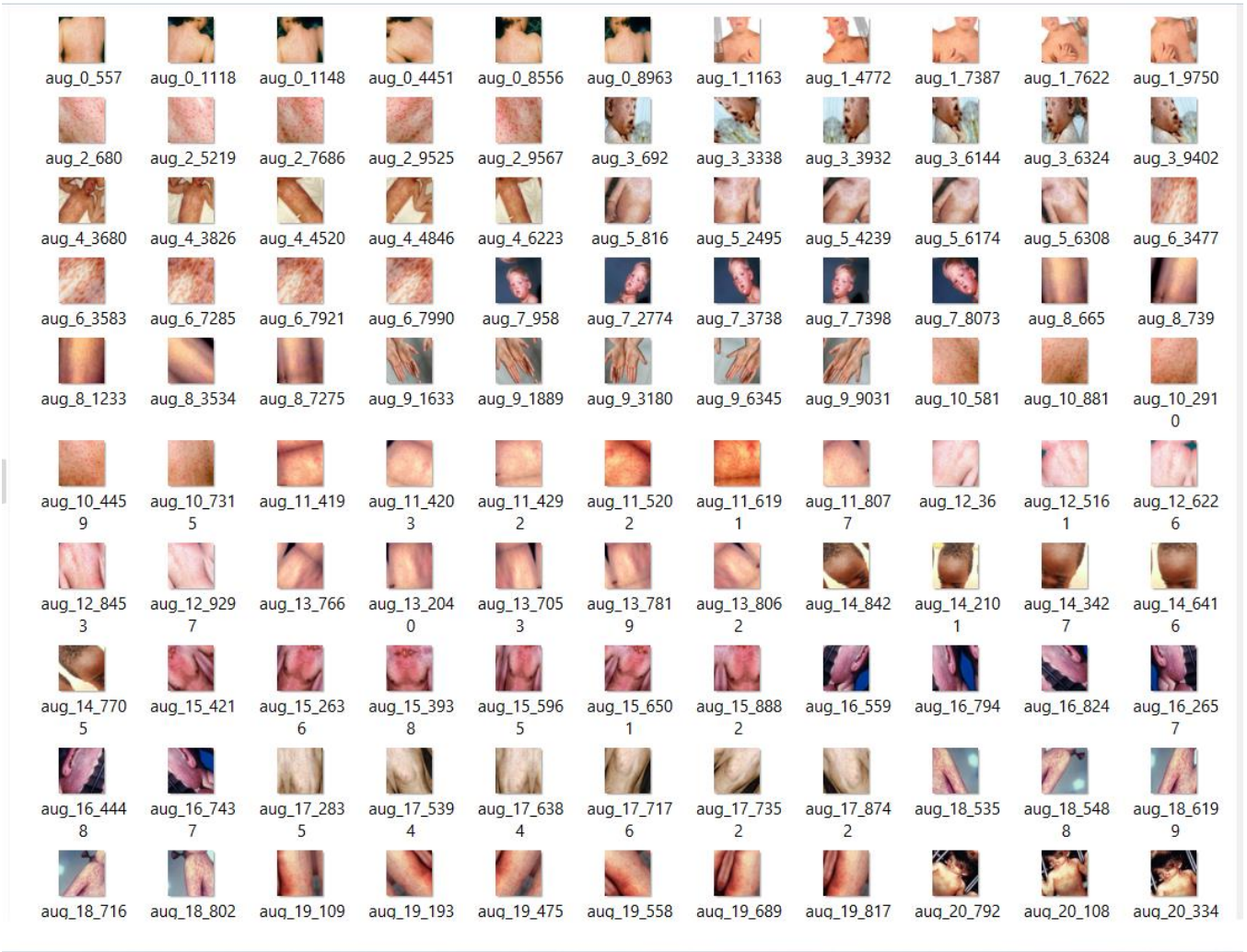


Fig 10.1-MONKEYPOX DATASET

NORMAL DATASET:

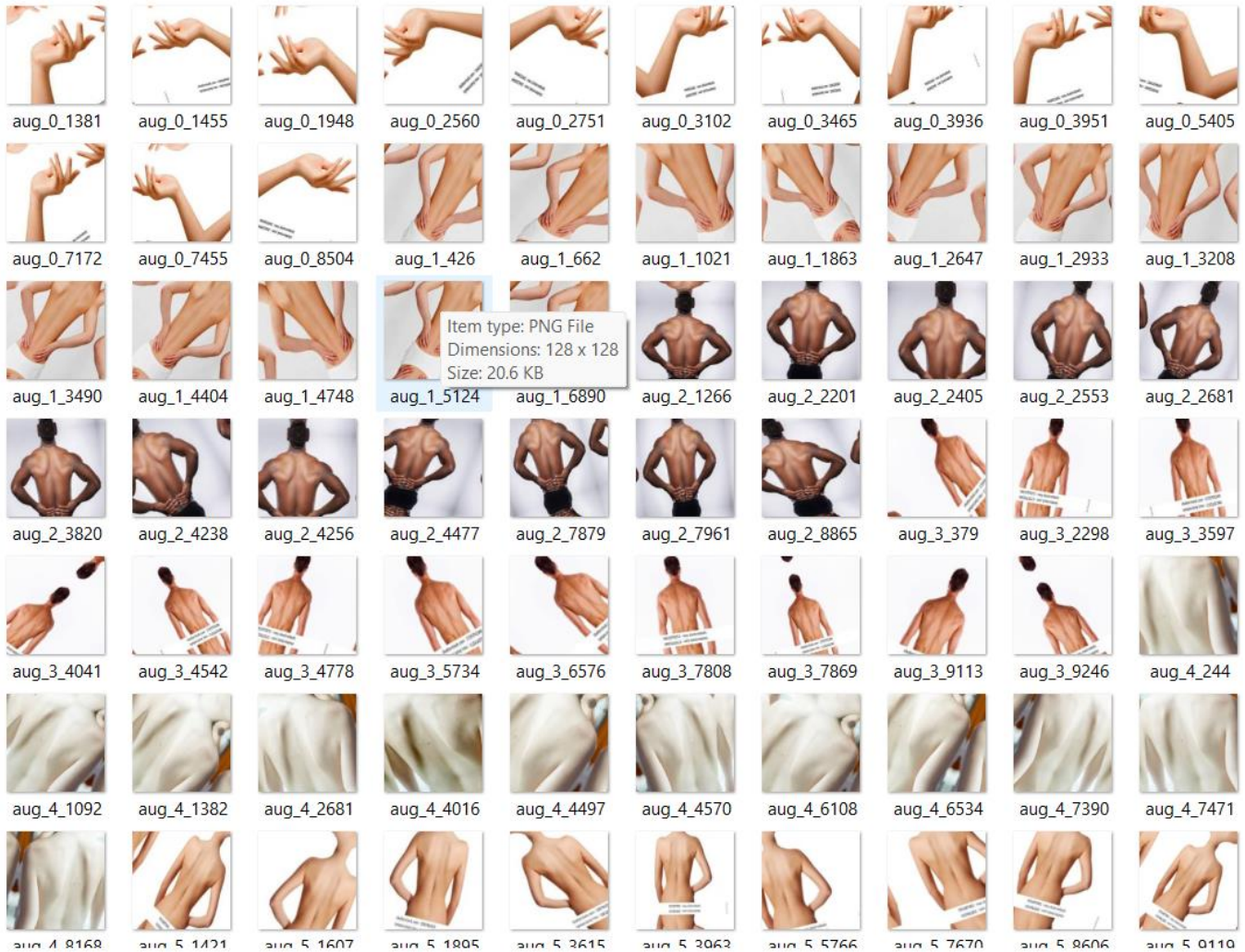


Fig 10.2-NORMAL DATASET

11.SAMPLE CODE:

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
import numpy as np

from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
from sklearn.model_selection import train_test_split
from keras.applications import VGG16
import cv2
import os
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.optimizers import Adam

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
import seaborn as sns
import webbrowser
from sklearn.metrics import roc_curve
from sklearn import metrics
import pandas as pd

main = tkinter.Tk()
main.title("Monkeypox Detection using Modified VGG16 & Custom CNN Model")
main.geometry("1300x1200")

global filename, cnn_model, X, Y
global X_train, X_test, y_train, y_test
global accuracy, precision, recall, fscore
labels = ['Normal', 'Monkeypox']

#function to return integer label for given plat disease name
def getID(name):
```

```
index = 0
    for i in range(len(labels)):
        if labels[i] == name: #return integer ID as label for given monkey pox name
            index = i
            break
    return index
def uploadDataset():
    global filename
    filename = filedialog.askdirectory(initialdir=".")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");
def preprocess():
    global filename, cnn, X, Y
    global X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    if os.path.exists("model/X.txt.npy"):
        X = np.load('model/X.txt.npy') #load X and Y data
        Y = np.load('model/Y.txt.npy')
    else:
        X = []
        Y = []
        for root, dirs, directory in os.walk(filename):
            for j in range(len(directory)):
                name = os.path.basename(root)
                if 'Thumbs.db' not in directory[j]:
                    img = cv2.imread(root+"/"+directory[j]) #read image
                    img = cv2.resize(img, (32,32)) #resize image
                    im2arr = np.array(img)
                    im2arr = im2arr.reshape(32,32,3) #resize as colur image
                    label = getID(name) #get id or label of plant disease
                    X.append(im2arr) #add all image pixel to X array
                    Y.append(label) #add label to Y array
                    print(name+" "+root+"/"+directory[j]+" "+str(label))
        X = np.asarray(X)
        Y = np.asarray(Y)
        np.save('model/X.txt',X) #save X and Y data for future user
        np.save('model/Y.txt',Y)
    X = X.astype('float32') #normalize image pixel with float values
    X = X/255
    indices = np.arange(X.shape[0]) #shuffling the images
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    Y = to_categorical(Y)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train and test
    text.insert(END,"Dataset Preprocessing & Image Normalization Process Completed\n\n")
    text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

```
    text.insert(END,"Dataset Train and Test Split\n\n")
    text.insert(END,"80% images used to train VGG & CNN algorithms : "+str(X_train.shape[0])+"\n")
    text.insert(END,"20% images used to test VGG & CNN algorithms : "+str(X_test.shape[0])+"\n")
    text.update_idletasks()
    test = X[3]
    cv2.imshow("Processed Image",cv2.resize(test,(200,200)))
    cv2.waitKey(0)
def calculateMetrics(algorithm, predict, y_test):
    a = accuracy_score(y_test,predict)*100
    p = precision_score(y_test, predict,average='macro') * 100
    r = recall_score(y_test, predict,average='macro') * 100
    f = f1_score(y_test, predict,average='macro') * 100
    fpr, tpr, _ = roc_curve(y_test,predict,pos_label=1)
    auc = metrics.auc(fpr, tpr) * 100
    accuracy.append(a)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    text.insert(END,algorithm+" AUC  :  "+str(auc)+"\n")
    text.insert(END,algorithm+" Accuracy  :  "+str(a)+"\n")
    text.insert(END,algorithm+" Precision : "+str(p)+"\n")
    text.insert(END,algorithm+" Recall    : "+str(r)+"\n")
    text.insert(END,algorithm+" FScore    : "+str(f)+"\n\n")
    text.update_idletasks()
    conf_matrix = confusion_matrix(y_test, predict)
    plt.figure(figsize =(6, 6))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis"
,fmt ="g");
    ax.set_ylim([0,2])
    plt.title(algorithm+" Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()

def runVGG():
    global X_train, X_test, y_train, y_test
    global accuracy, precision, recall, fscore
    text.delete('1.0', END)
    accuracy = []
    precision = []
    recall = []
    fscore = []
    if os.path.exists('model/vgg_model.json'):
        with open('model/vgg_model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            vgg = model_from_json(loaded_model_json)
        json_file.close()
        vgg.load_weights("model/vgg_model_weights.h5")
```

```
        vgg._make_predict_function()
    else:
        #defining VGG16 object with our own training and test data
        vgg16 = VGG16(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]),
include_top=False, weights="imagenet")
        vgg16.trainable = False
        #defining transfer learning model object
        vgg = Sequential()
        #adding VGG16 to tranfer learning model
        vgg.add(vgg16)
        #defining new layers for transfer learning model with 32 layers to filter images
        vgg.add(Convolution2D(32, 1, 1, input_shape = (X_train.shape[1], X_train.shape[2],
X_train.shape[3]), activation = 'relu'))
        #max pooling collect all filtered data from VGG
        vgg.add(MaxPooling2D(pool_size = (1,1)))
        #another layer to refilter image data
        vgg.add(Convolution2D(32, 1, 1, activation = 'relu'))
        vgg.add(MaxPooling2D(pool_size = (1, 1)))
        vgg.add(Flatten())
        vgg.add(Dense(output_dim = 256, activation = 'relu')) #defining output layer
        vgg.add(Dense(output_dim = y_train.shape[1], activation = 'softmax'))
        vgg.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy']) #compiling
thee model
        hist = vgg.fit(X_train, y_train, batch_size=16, epochs=10, shuffle=True, verbose=2,
validation_data=(X_test, y_test)) #start training model
        vgg.save_weights('model/vgg_model_weights.h5')
        model_json = vgg.to_json()
        with open("model/vgg_model.json", "w") as json_file:
            json_file.write(model_json)
        json_file.close()
        f = open('model/vgg_history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    print(vgg.summary())
    predict = vgg.predict(X_test)
    predict = np.argmax(predict, axis=1)
    testY = np.argmax(y_test, axis=1)
    calculateMetrics("Modified VGG16", predict, testY)
def runCNN():
    global X_train, X_test, y_train, y_test, cnn_model
    global accuracy, precision, recall, fscore
    if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            cnn_model = model_from_json(loaded_model_json)
        json_file.close()
        cnn_model.load_weights("model/model_weights.h5")
        cnn_model._make_predict_function()
```

```
    else:
        cnn_model = Sequential() #define CNN custom model with multiple layers and in this custom CNN we are not using VGG16
        #defining cnn layer with 32 filters and kernel matrix size as 3 X 3
        cnn_model.add(Conv2D(32, (3, 3), padding="same",input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3])))
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization(axis=3))
        cnn_model.add(MaxPooling2D(pool_size=(3, 3))) #max pooling collected filtered image pixels from previous CNN layer
        cnn_model.add(Dropout(0.25))
        cnn_model.add(Conv2D(64, (3, 3), padding="same")) #another CNN layer to refilter images pixles so de define multiple layers to filter image data
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization(axis=3))
        cnn_model.add(Conv2D(64, (3, 3), padding="same"))
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization(axis=3))
        cnn_model.add(MaxPooling2D(pool_size=(2, 2)))
        cnn_model.add(Dropout(0.25))
        cnn_model.add(Conv2D(128, (3, 3), padding="same"))
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization(axis=3))
        cnn_model.add(Conv2D(128, (3, 3), padding="same"))
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization(axis=3))
        cnn_model.add(MaxPooling2D(pool_size=(2, 2)))
        cnn_model.add(Dropout(0.25))
        cnn_model.add(Flatten())
        cnn_model.add(Dense(1024))
        cnn_model.add(Activation("relu"))
        cnn_model.add(BatchNormalization())
        cnn_model.add(Dropout(0.5))
        cnn_model.add(Dense(y_train.shape[1]))
        cnn_model.add(Activation("softmax"))
        cnn_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy']) #compile the model
        hist = cnn_model.fit(X_train, y_train, batch_size=16, epochs=10, shuffle=True, verbose=2, validation_data=(X_test, y_test)) #start traing model
        cnn_model.save_weights('model/model_weights.h5')
        model_json = cnn_model.to_json()
        with open("model/model.json", "w") as json_file:
            json_file.write(model_json)
        json_file.close()
        f = open('model/history.pckl', 'wb')
        pickle.dump(hist.history, f)

    f.close()
```

```
    print(cnn_model.summary())
    predict = cnn_model.predict(X_test)
    predict = np.argmax(predict, axis=1)
    testY = np.argmax(y_test, axis=1)
    calculateMetrics("Custom CNN", predict, testY)


def graph():
    output = "<html><body><table align=center border=1><tr><th>Algorithm
Name</th><th>Accuracy</th><th>Precision</th><th>Recall</th>"
    output+="<th>FSCORE</th></tr>"
    output+="<tr><td>Modified
VGG16</td><td>"+str(accuracy[0])+"</td><td>"+str(precision[0])+"</td><td>"+str(recall[0])+"</td><t
d>"+str(fscore[0])+"</td></tr>"
    output+="<tr><td>Custom
CNN</td><td>"+str(accuracy[1])+"</td><td>"+str(precision[1])+"</td><td>"+str(recall[1])+"</td><td>
"+str(fscore[1])+"</td></tr>"
    output+="</table></body></html>"
    f = open("table.html", "w")
    f.write(output)
    f.close()
    webbrowser.open("table.html",new=2)

    df = pd.DataFrame([['Modified VGG16','Precision',precision[0]],['Modified
VGG16','Recall',recall[0]],['Modified VGG16','F1 Score',fscore[0]],['Modified
VGG16','Accuracy',accuracy[0]],
                       ['Custom CNN','Precision',precision[1]],['Custom CNN','Recall',recall[1]],['Custom
CNN','F1 Score',fscore[1]],['Custom CNN','Accuracy',accuracy[1]],
                      ],columns=['Algorithms','Performance Output','Value'])
    df.pivot("Algorithms", "Performance Output", "Value").plot(kind='bar')
    plt.show()


def predict():
    global cnn_model
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename)
    img = cv2.resize(image, (32,32))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,32,32,3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    preds = cnn_model.predict(img)
    predict = np.argmax(preds)
    score = np.amax(preds)
    img = cv2.imread(filename)
```

```
    img = cv2.resize(img, (600,400))
    cv2.putText(img, 'Classification Result: '+labels[predict]+" Detected", (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (255, 0, 0), 2)
    cv2.imshow('Classification Result: '+labels[predict]+" Detected", img)
    cv2.waitKey(0)
font = ('times', 16, 'bold')
title = Label(main, text='Monkeypox Detection using Modified VGG16 & Custom CNN Model')
title.config(bg='firebrick4', fg='dodger blue')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)
font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Monkeypox Dataset", command=uploadDataset,
bg='#ffb3fe')
uploadButton.place(x=50,y=550)
uploadButton.config(font=font1)
processButton = Button(main, text="Preprocess Dataset", command=preprocess, bg='#ffb3fe')
processButton.place(x=340,y=550)
processButton.config(font=font1)
vggButton1 = Button(main, text="Run VGG16 Algorithm", command=runVGG, bg='#ffb3fe')
vggButton1.place(x=570,y=550)
vggButton1.config(font=font1)
cnnButton = Button(main, text="Run Custom CNN Algorithm", command=runCNN, bg='#ffb3fe')
cnnButton.place(x=50,y=600)
cnnButton.config(font=font1)
graphButton = Button(main, text="Comparison Graph", command=graph, bg='#ffb3fe')
graphButton.place(x=340,y=600)
graphButton.config(font=font1)
predictButton = Button(main, text="Predict Disease from Test Image", command=predict, bg='#ffb3fe')
predictButton.place(x=570,y=600)
predictButton.config(font=font1)
main.config(bg='LightSalmon3')
main.mainloop()
```

12.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.

TYPES OF TESTS

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.

2.Integration 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.

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:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

.

Output = : identified classes of application outputs must be exercised.

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.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

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

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.

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.

Module	Functionality	Test Case	Expected Results	Actual Results	Result	Priority
Upload monkeypox Dataset	Upload process	Click on the upload Monkeypox dataset Button	Path of the dataset Must be visible in the dataset location field	A validation has been populated as Expected	Pass	High
Preprocess Dataset	Data Preprocessing	The desired dataset must be pre-processed	The desire dataset is preprocessing	Total images found in the dataset are returned	Pass	High
Run VGG16 Algorithm	VGG16 commonly used for image classification tasks	Click on the VGG16 Algorithm button so it can train and split the dataset	Image classification process should be completed	Image classification process should be completed and 80% of images used to train.	Pass	Medium
Run Custom CNN Algorithm	CNN Algorithm designing a neural network tailored to the specific characteristics of the disease related images	Click on the Run Custom CNN Algorithm button.	Identify the characteristics of the input and compare with all the images features.	Identify the characteristics of the input and 20% of the images used to test.	Pass	High
Comparison Graph	Comparison should be completed	The graph could be comparing the results of VGG16 &CNN	Give the Comparison of the CNN & VGG16	Give the Comparison of the CNN & VGG16	Pass	High
Predict Disease from test image	Give the Input test image	Give the accurately predict the presence of monkeypox in a given test image	Give the output of the image whether the test image is detected by monkeypox or normal	Give the output for the test whether image detected or normal	Pass	High

13.OUTPUT SCREENS

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

Fig-13.1 OUTPUT SCREEN-1

In above screen click on ‘Upload Monkeypox Dataset’ button to upload dataset and get below output

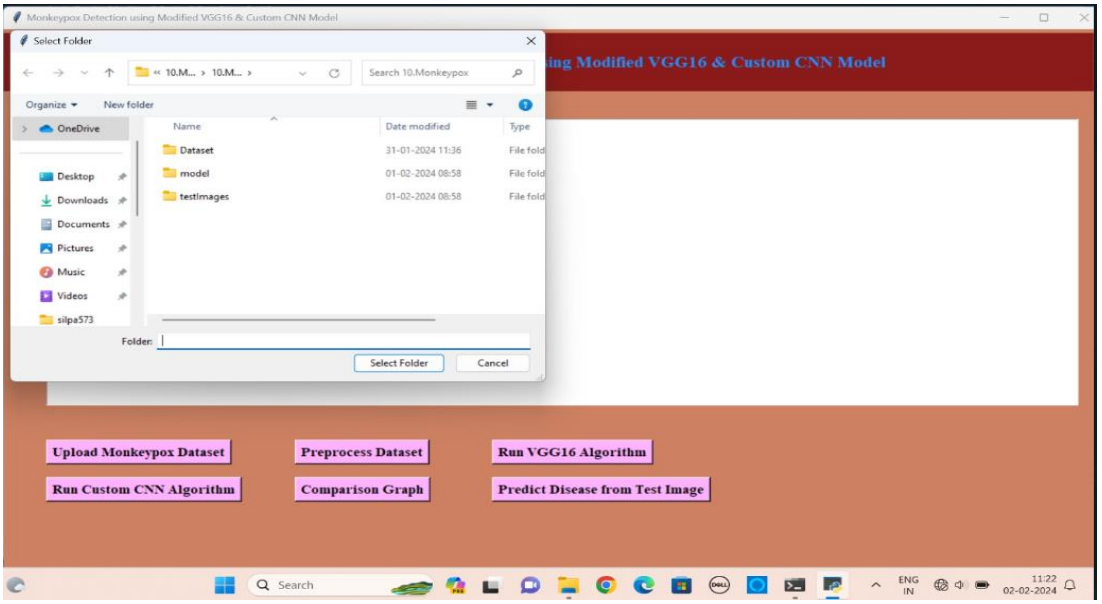


Fig.13.2. DATASAT LOCATION SCREEN-2

In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output

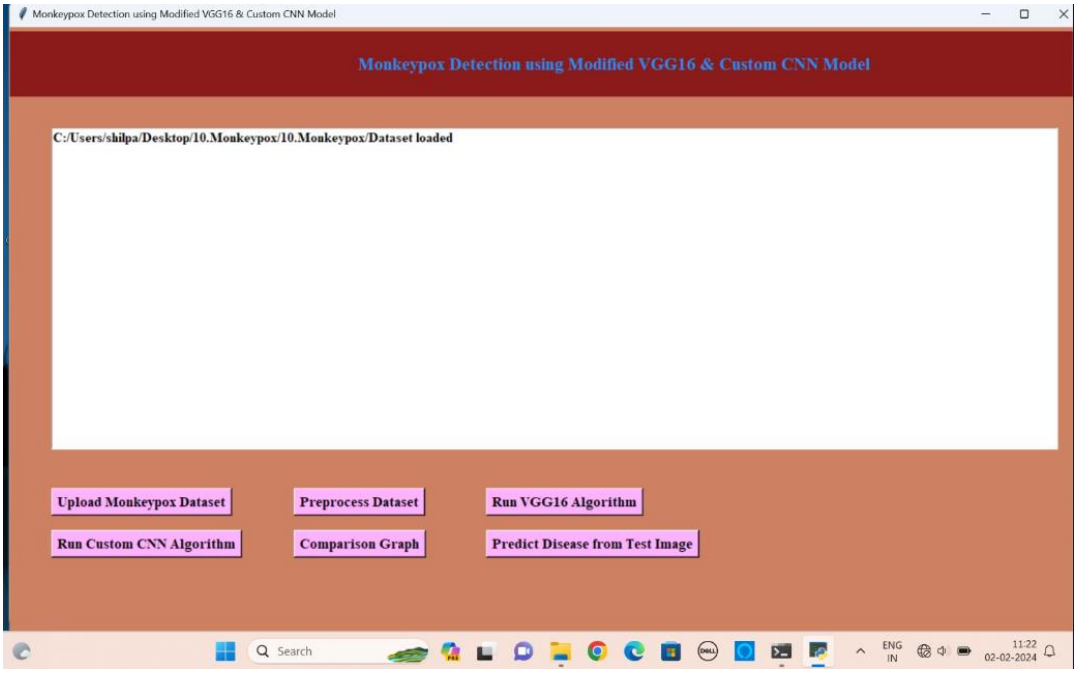


Fig.13.3.DATASET UPLOADED SCREEN-3

In above screen dataset is loaded and now click on 'Preprocess Dataset' button to read all images and then processed them

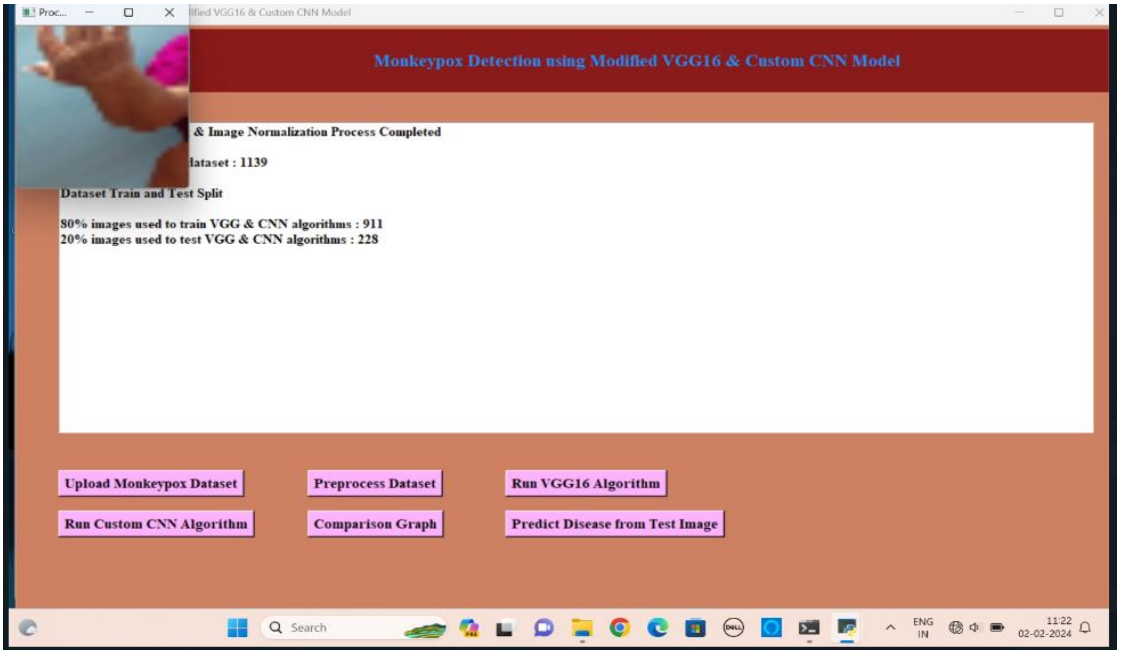


Fig.13.4. PREPROCESSING SCREEN-4

In above screen we can see all images read and then process and we can see dataset contains total 1139 images and application using 911 (80%) images for training and 228 (20%) images for testing and to

check all images are processed properly I am displaying one sample image and now close that image and then click on ‘Run VGG16 Algorithm’ button to train VGG and get below output

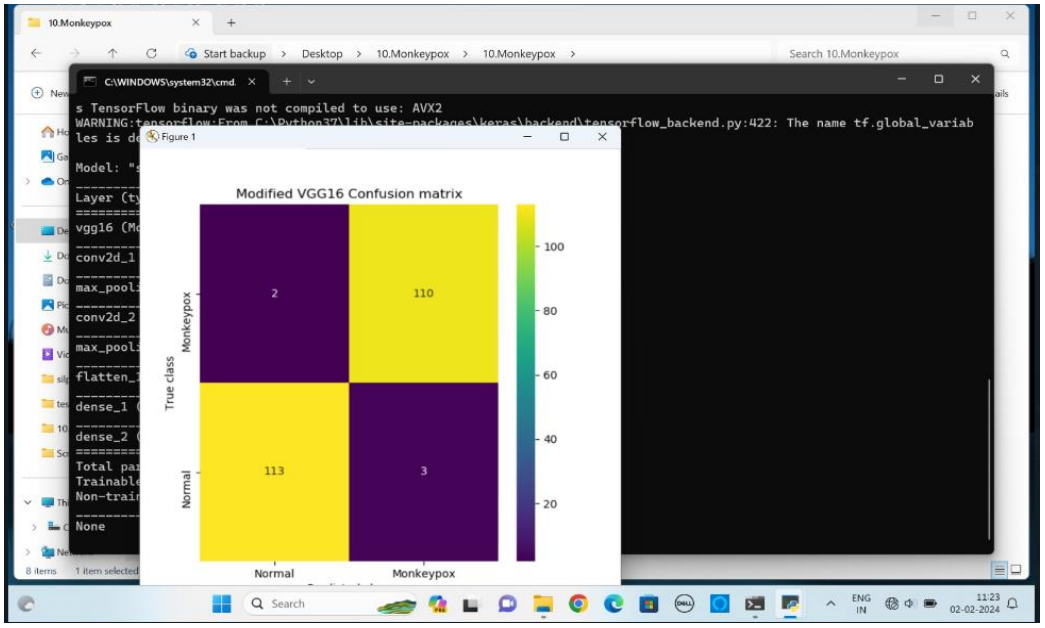


Fig.13.5 MODIFIED VGG16 CONFUSION MATRIX SCREEN-5

In above screen with VGG16 we got 98.24% accuracy and we can see other metric values also and in confusion matrix graph x-axis represents Predicted Classes and y-axis represents True Classes and we can prediction in matching row and column name is correct and un-matched rows and columns names are incorrect prediction and in above graph we can see VGG16 predicted 1 and 3 (total 4 images) incorrectly and now close above graph and then click on ‘Run Custom CNN Algorithm’ button to train Custom CNN and get below output

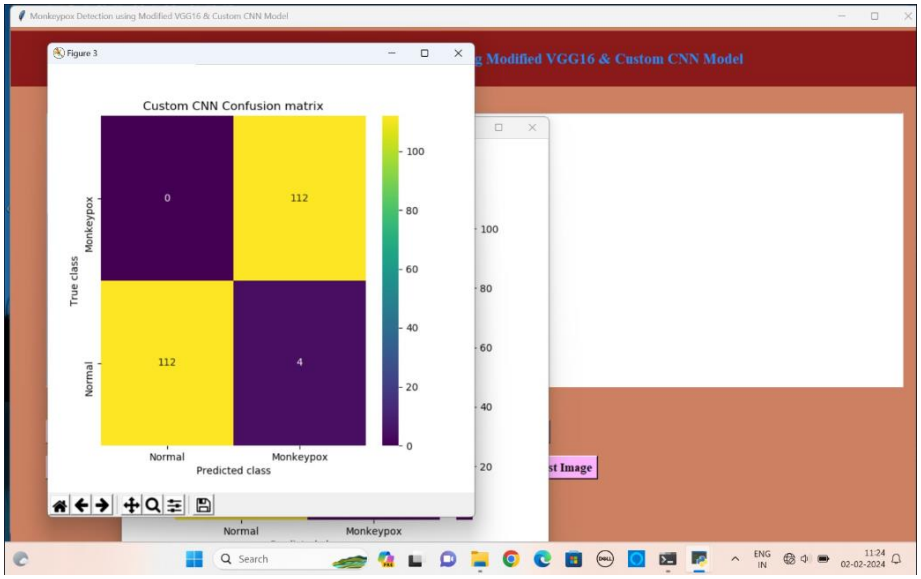


Fig.13.6. CUSTOM CNN CONFUSION MATRIX SCREEN-6

In above screen with Custom CNN we got 99.12% accuracy and in confusion matrix graph we can see only 2 images are incorrectly predicted and now close above graph and then click on ‘Comparison Graph’ button to get below output

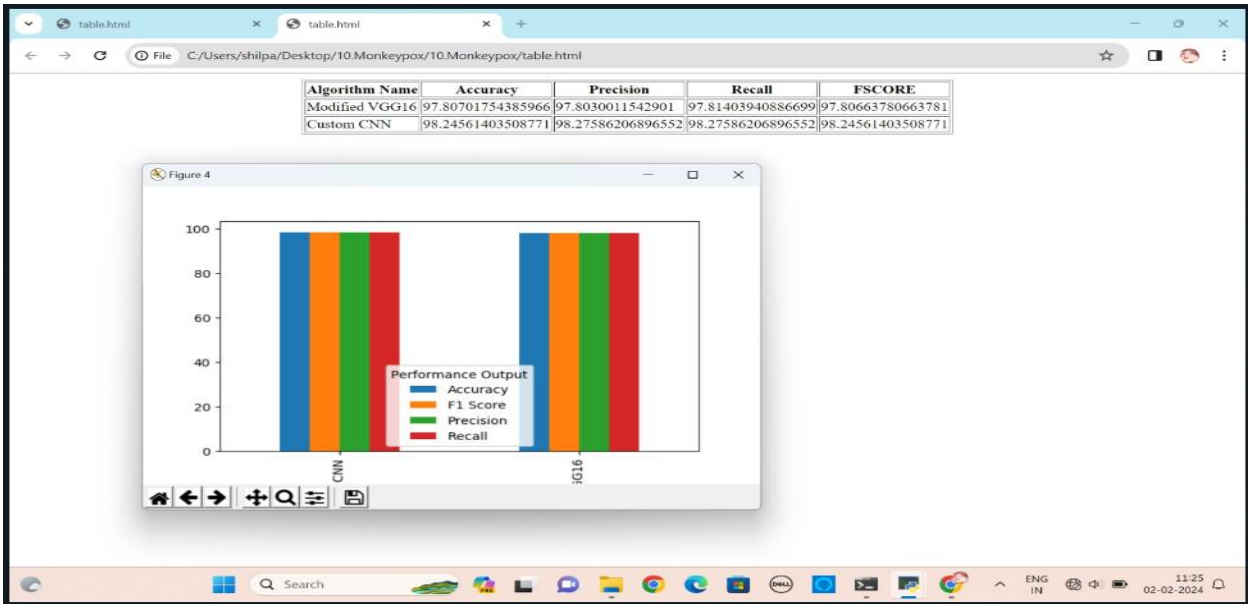


Fig.13.7. COMPARISION GRAPH SCREEN-7

In above screen with tabular format and graph format we can see result of both algorithms and now go back to main application and then click on ‘Predict Disease from Test Image’ button to upload test image and get below output above screen selecting and uploading ‘4.jpg’ image and then click on ‘Open’ button to get below prediction output

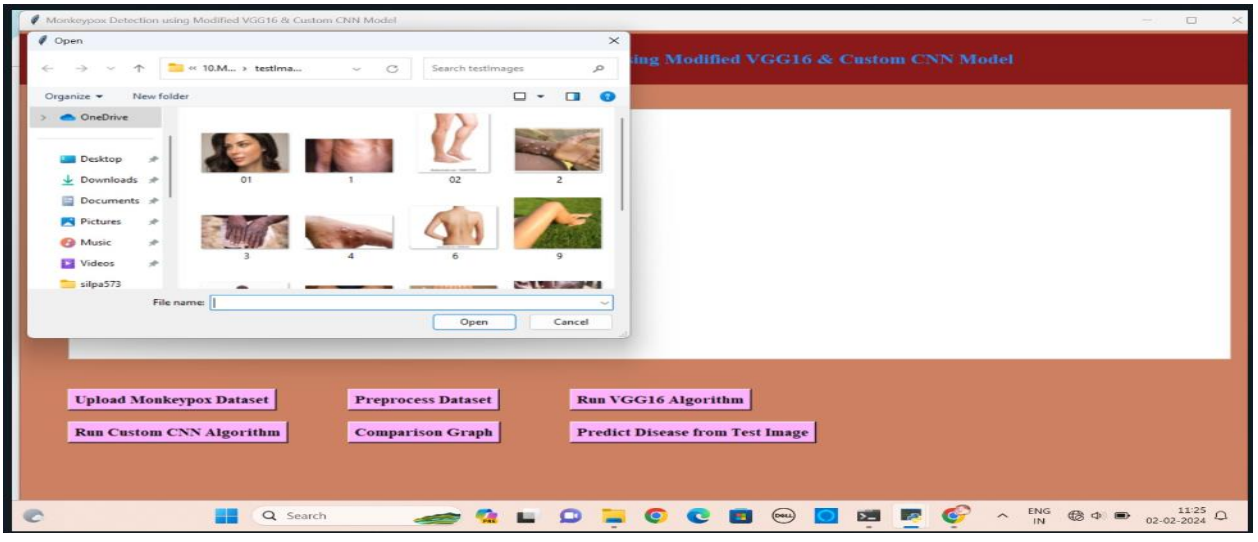


Fig.13.8. SELECTING TEST DATASETS SCREEN-8

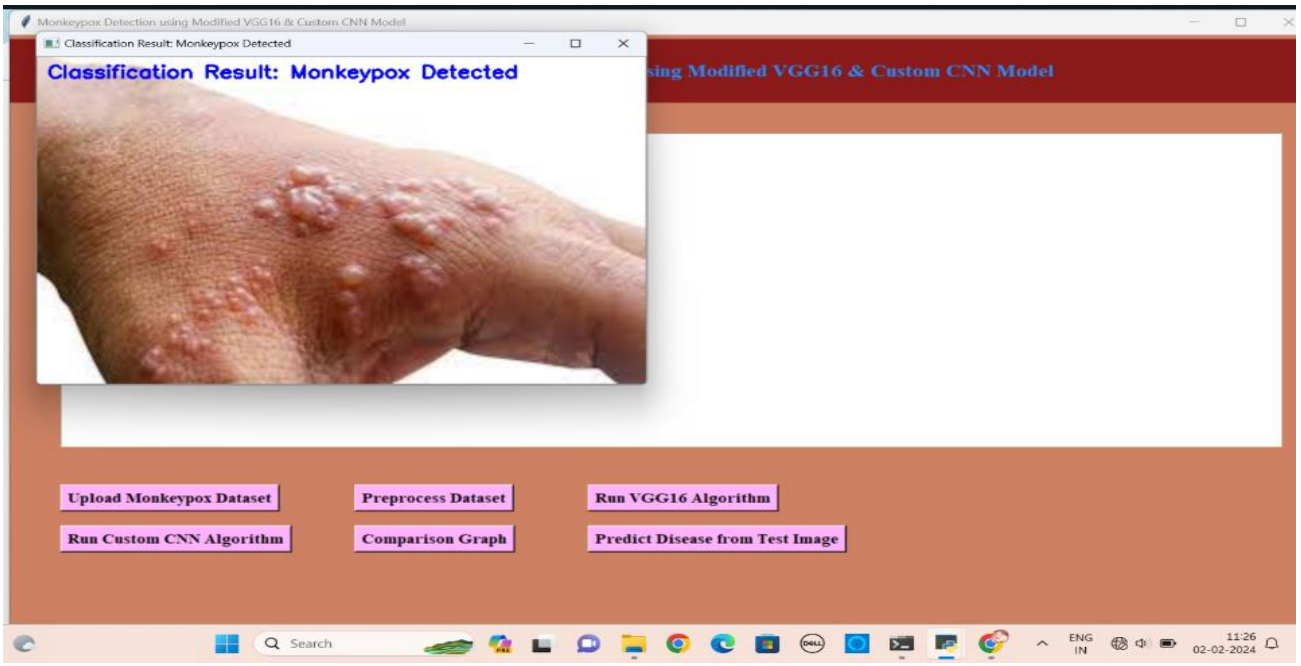


Fig.13.9. OUTPUT SCREEN-9

In above screen monkey pox detected and similarly you can upload other images and test them

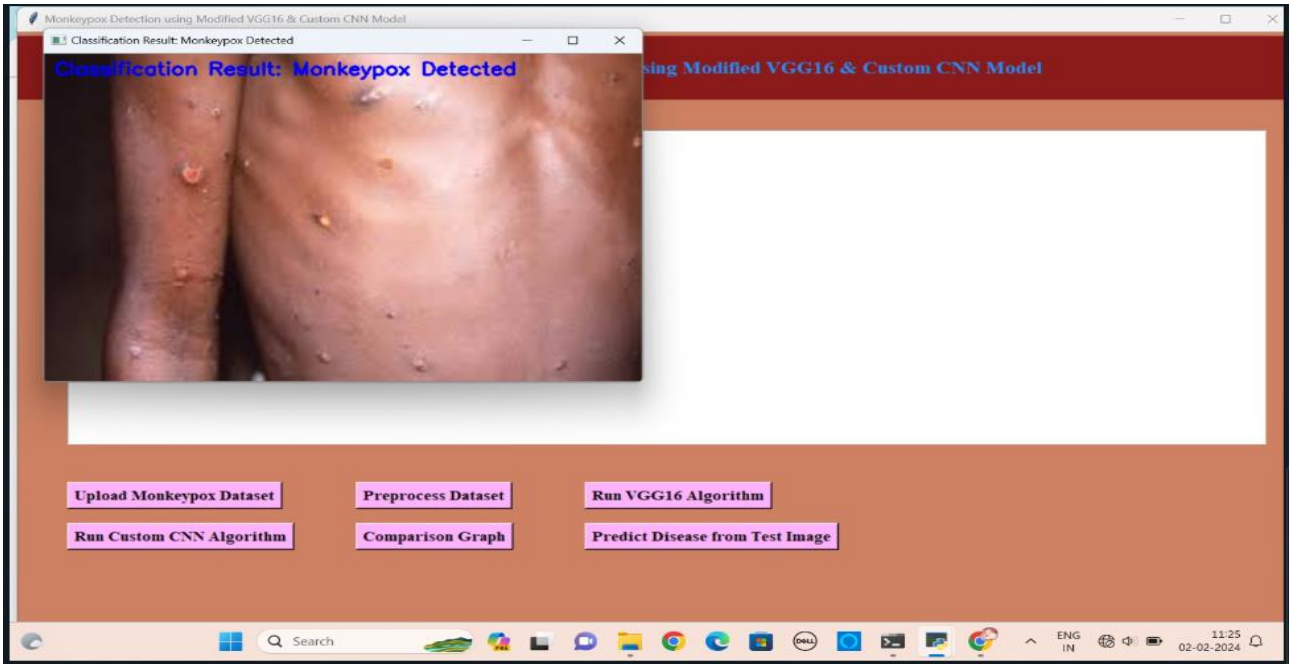


Fig.13.10. OUTPUT SCREEN-10

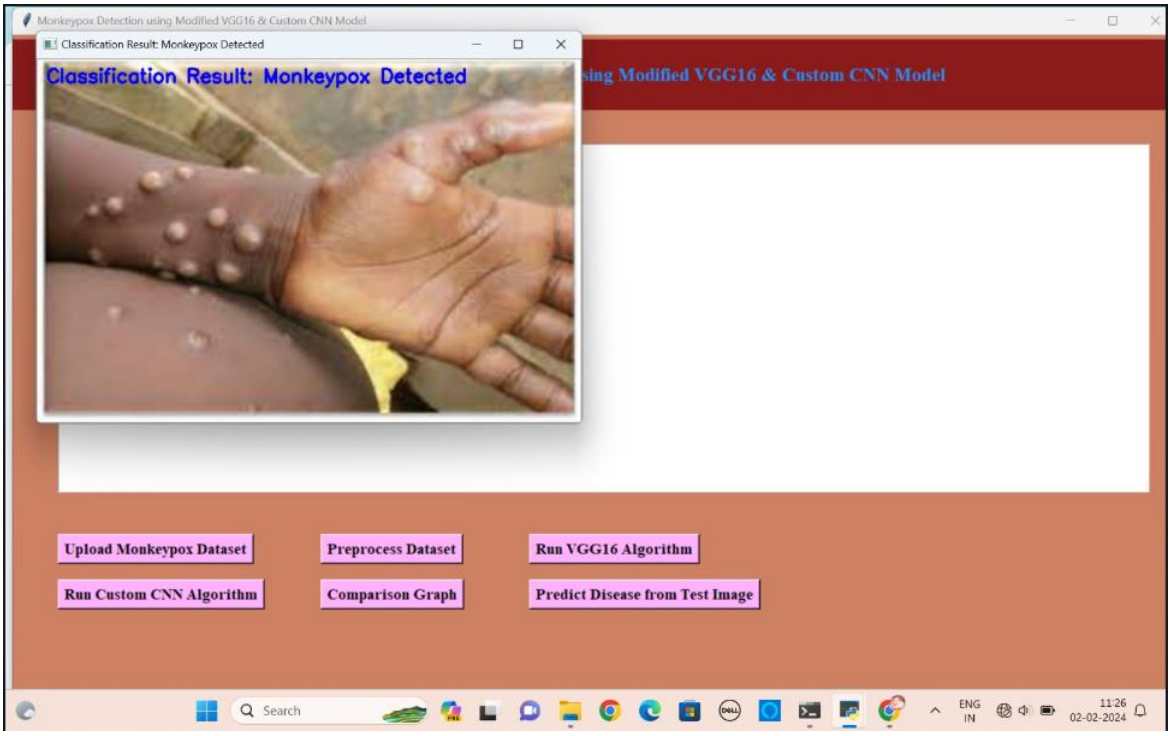


Fig.13.11. OUTPUT SCREEN-11

Fig.13.12. OUTPUT SCREEN-12

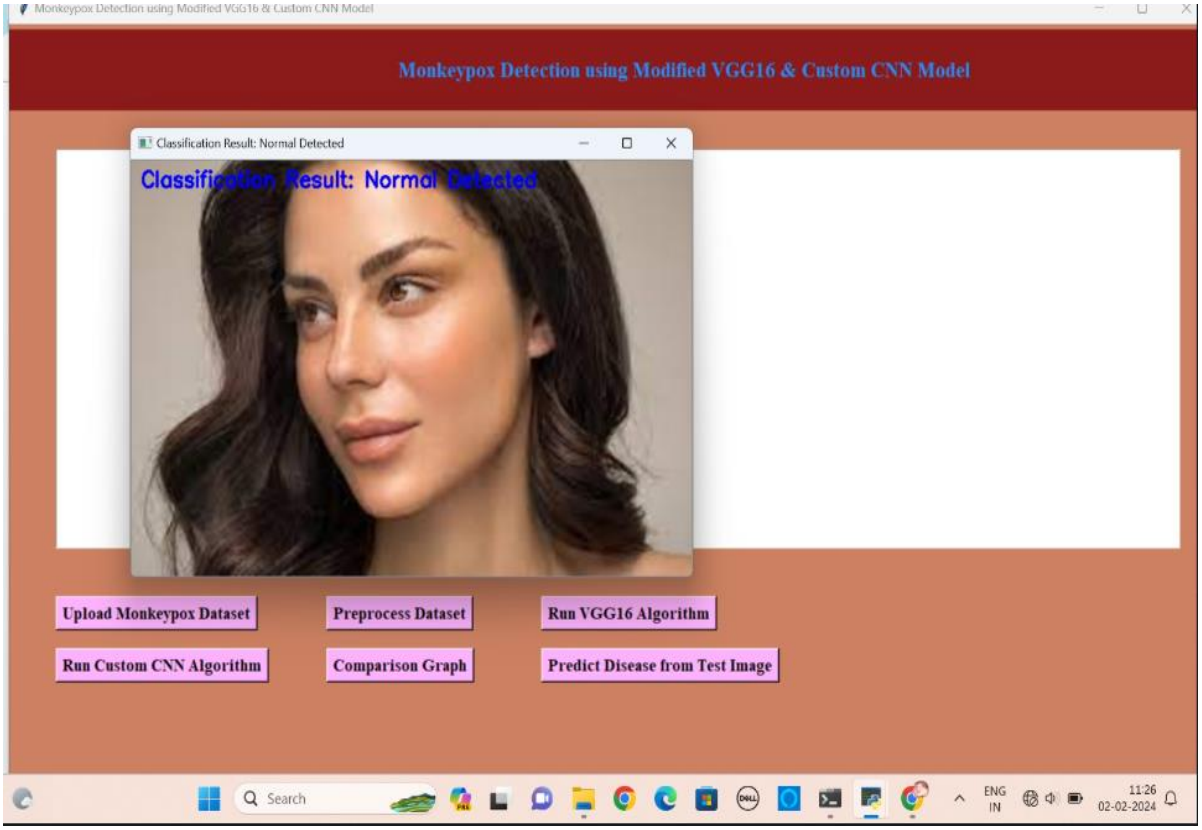


Fig.13.13. OUTPUT SCREEN-13

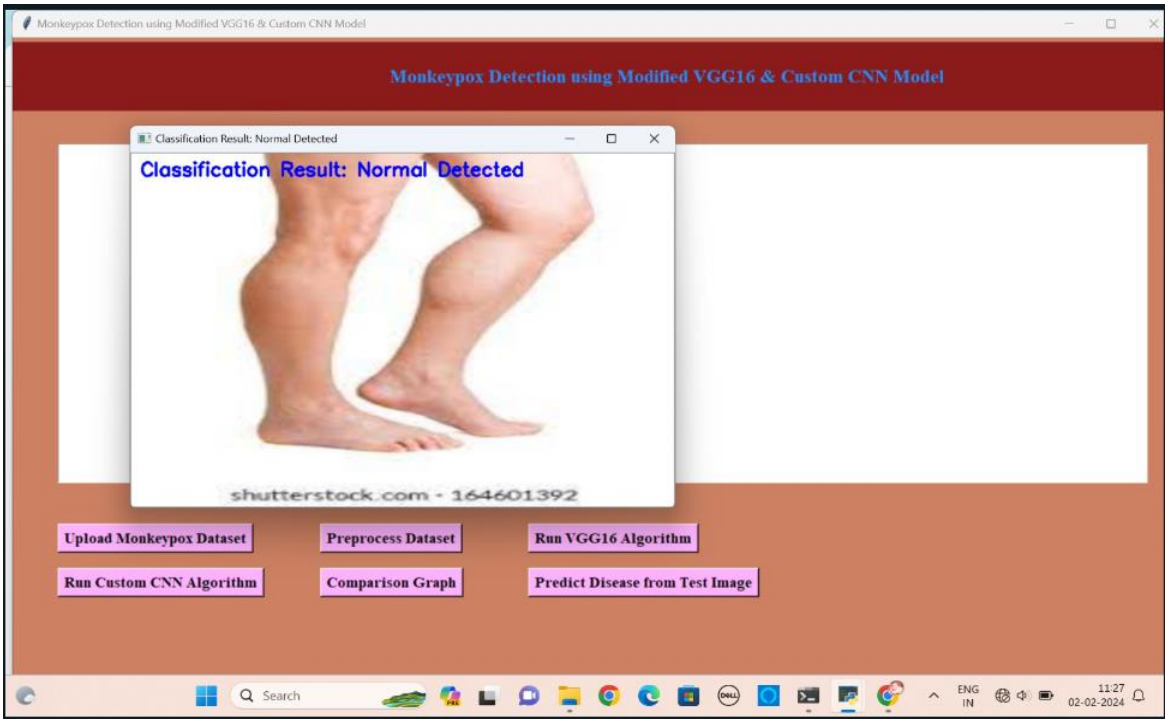


Fig.13.14. OUTPUT SCREEN-14

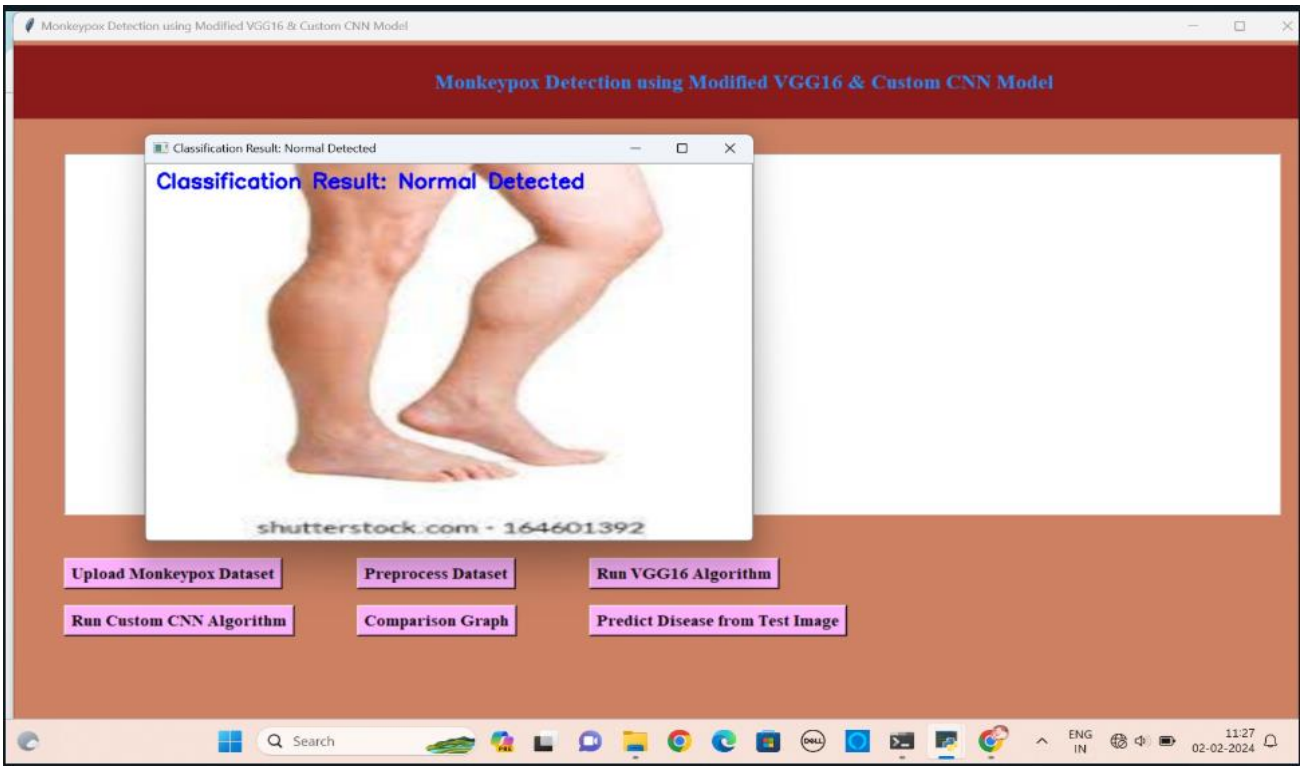


Fig.13.15. OUTPUT SCREEN-15

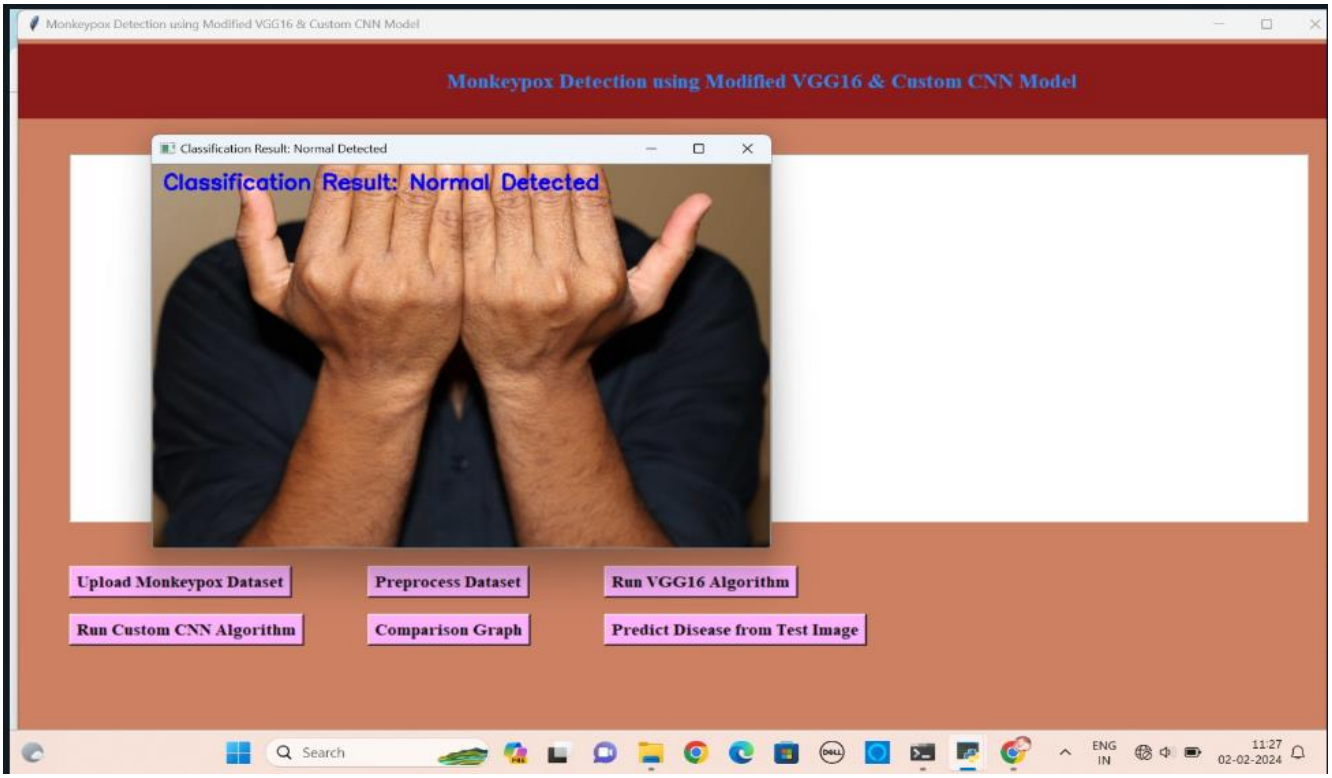


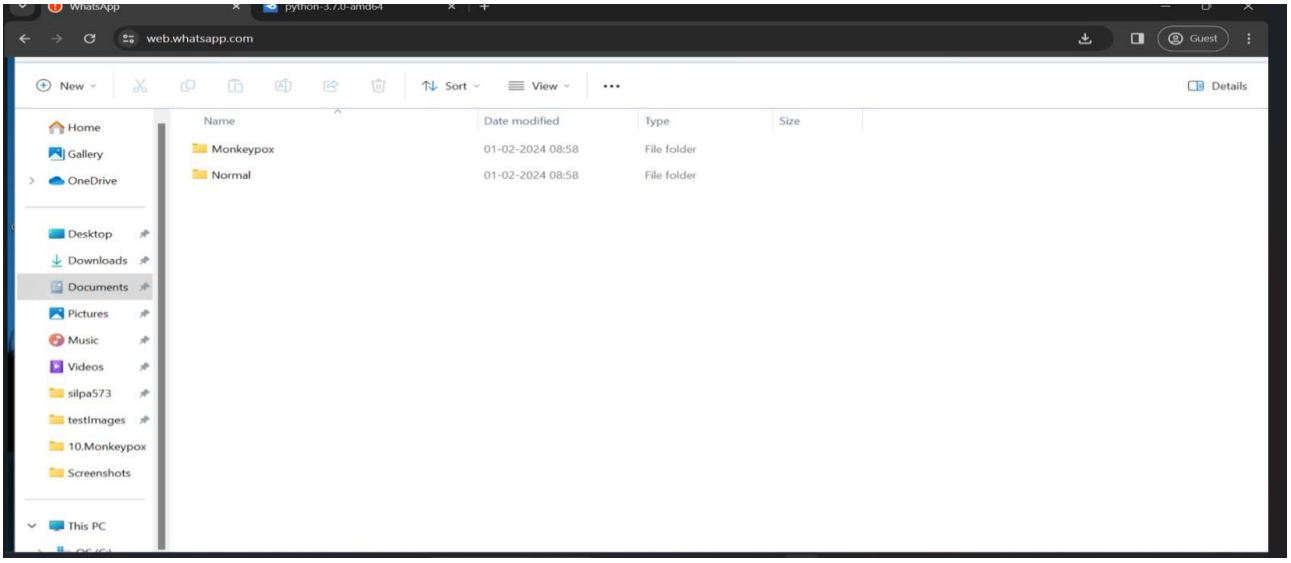
Fig.13.16. OUTPUT SCREEN-16

14.RESULT ANALYSIS

In this project we are designing Transfer Learning Modified VGG16 and Custom CNN algorithm to predict Monkeypox disease as this disease is not deadly but spreading very Fastly. To deal with this disease for timely detection doctors can use these algorithms for detection.

To train both algorithms we have used Normal SKIN and Monkeypox images and in both algorithm VGG 16 giving 98% accuracy and Custom CNN giving 99% accuracy.

Below screen showing images used in this project



In above screen we have two folders and just go inside any folder to view those images

So we are using above images to train VGG 16 and Custom CNN algorithms.

To implement this project we have designed following modules

1) Upload Monkeypox Dataset: using this module we will upload dataset to application.
2) Preprocess Dataset: using this module we will read all images and then resize images to equal size and then normalize image pixel values and then split entire dataset into TRAIN and TEST where application user 80% images for training and 20% for testing. 20% test images will be applied on trained model to calculate correct prediction accuracy.
3) Run VGG16 Algorithm: above processed 80% images will be input to VGG algorithm to trained prediction model and this model will be applied don test images to calculate prediction accuracy.
4) Run Custom CNN Algorithm: above processed 80% images will be input to Custom CNN algorithm to trained prediction model and this model will be applied don test images to calculate prediction accuracy.
5) Comparison Graph: using this module we will plot comparison graph between VGG and Custom CNN algorithms
6) Predict Disease from Test Image: using this module we will upload test image and then Custom CNN will predict weather image is normal or contains Monkeypox disease.

In below screen we are showing code with comments on VGG and Custom CNN algorithms

In above screen read red colour comments to know about VGG16 implementation

ALGORITHM	ACCURACY	PRECISION	RECALL	FSCORE
Modified VGG16	97.80	97.80	97.81	97.80
Custom CNN	98.24	98.24	98.27	98.24

15.CONCLUSION AND FUTURE SCOPE

In this study, we have presented Monkeypox detection using a modified VGG16 & DCNN model. The algorithms have achieved accuracies 97.8 and 98.2 respectively. Despite being a small dataset, the promising results obtained after 3-fold cross-validation reveal the potential to use AI-assisted early diagnosis of this disease. We believe that our proposed concept of the monkeypox suspects is to conduct preliminary screening from the comforts of home at the early stages of the infection. Our model will inspire future researchers and practitioners to use the transfer learning approach in clinical diagnosis. Some of the constraints associated with our work can be overcome by continuously collecting new images of Monkeypox-infected patients, evaluating the proposed VGG16 model's performance on highly imbalanced data, and deploying our proposed model in the development of a mobile-based diagnosis tools.

16.REFERENCES:

[1] Andrea M McCollum and Inger K Damon. Human monkeypox. Clinical infectious diseases, 58(2):260–267, 2014.

[2] Emmanuel Alakunle, Ugo Moens, Godwin Nchinda, and Malachy Ifeanyi Okeke. Monkeypox virus in nigeria: infection biology, epidemiology, and evolution. Viruses, 12(11):1257, 2020.

[3] Marlyn Moore and Farah Zahra. Monkeypox (accessed on may 22, 2022). https://www.ncbi.nlm.nih.gov/ books/NBK574519/, May 2022.

[4] Leisha Diane Nolen, Lynda Osadebe, Jacques Katomba, Jacques Likofata, Daniel Mukadi, Benjamin Monroe, Jeffrey Doty, Christine Marie Hughes, Joelle Kabamba, Jean Malekani, et al. Extended human-to-human transmission during a monkeypox outbreak in the democratic republic of the congo. Emerging infectious diseases, 22(6):1014, 2016.

[5] Phi-Yen Nguyen, Whenayon Simeon Ajisegiri, Valentina Costantino, Abrar A Chughtai, and C Raina MacIntyre. Reemergence of human monkeypox and declining population immunity in the context of urbanization, nigeria, 2017–2020. Emerging Infectious Diseases, 27(4):1007, 2021

[6] Monkeypox signs and symptoms. (accessed on may 30, 2022). https://www.cdc.gov/poxvirus/ monkeypox/symptoms.html, 2022.

[7] Michaeleen Doucleff. Scientists warned us about monkeypox in 1988. here's why they were right. (accessed on may 27, 2022). https://www.npr.org/sections/goatsandsoda/2022/05/27/1101751627/ scientists-warned-us-about-monkeypox-in-1988-heres-whythey-were-right, May 2022.

[8]Deep learning in medicine—promise, progress, and challenges Authors: F. Wang, L. P. Casalino, and D. Khullar

[9]Classification of skin lesions using transfer learning and augmentation with Alex-net Authors: K. M. Hosny, M. A. Kassem, and M. M. Foaud

[10]A survey on transfer learning Authors: S. J. Pan and Q. Yang

[11]A survey on image data augmentation for deep learning Authors: C. Shorten and T. M. Khoshgoftaar

[12] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. arXiv preprint arXiv:2003.11597, 2020.

[13] Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. Pattern Analysis and Applications, 24(3):1207–1220, 202