A PROJECT REPORT ON

# IDENTIFICATION OF MEDICINAL PLANT USING CONVOLUTIONAL NEURAL NETWORKS

## Submitted to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA**
**For Partial Fulfillment of Award of the Degree of**

## MASTER OF COMPUTER APPLICATIONS

Submitted By

**L. SRAVANI(22X41F0034)** Under the Esteemed Guidance of
**Ms. M. ANITHA,**
**Head of Department, Department of MCA.**


**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

**S.R.K INSTITUTE OF TECHNOLOGY**
(AFFILIATED TO JNTU, KAKINADA)
Enikepadu, Vijayawada – 521108.

**2023-2024**

# S.R.K INSTITUTE OF TECHNOLOGY

ENIKEPADU, VIJAYAWADA.

## DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

## CERTIFICATE

This is to certify that this project report entitled " **IDENTIFICATION OF MEDICINAL PLANT USING CONVOLUTIONAL NEURAL NETWORKS"** is the bonafide work of**. L.SRAVANI(22X41F0034)**, in partial fulfillment of the requirements for the award of the post graduate degree in MASTER OF COMPUTER APPLICATIONS during the academic year 2023-2024. This work has carried out under our supervision and guidance.

**(Ms. M. ANITHA)**
**Signature of the Guide**

**(Ms. M. ANITHA)**
**Signature of the HOD**

# DECLARATION

I **L.SRAVANI** hereby declare that the project report entitled "**IDENTIFICATION OF MEDICINAL PLANT USING CONVOLUTIONAL NEURAL NETWORKS**"

is an original work done in the Department of Master of Computer Applications, SRK Institute of Technology, Enikepadu, Vijayawada, during the academic year 2023-2024, in partial fulfillment for the award of the Degree of Master of Computer Applications. I assure that this project is not submitted to any other College or University.

| Roll No | Name of the Student | Signature |
|---|---|---|
| **22X41F0034** | **L.SRAVANI** | |

# ACKNOWLEDGEMENT

Firstly I would like to convey my heart full thanks to the Almighty for the blessings on me tocarry out this project work without any disruption.

I am very much thankful to our principal **Dr. EKAMBARAM NAIDU** for his kind support and facilities provided at our campus which helped me to bring out this project successfully.

I am very much grateful to **Mrs. M. ANITHA**, H.O.D of M.C.A Department, for her valuable guidance which helped me to bring out this project successfully. Her wise approach made me to learn the minute details of the subject. Her matured and patient guidance paved a way for completing my project with the sense of satisfaction and pleasure.

I am also thankful for my project coordinator **Mrs. M. ANITHA** mam for her valuable guidance which helped me to bring this project successfully.

I am extremely thankful to **Ms.M.ANITHA, Project guide**who guided me throughout the project. I am thankful to her for giving me the most independence and freedom throughout various phases of the project.

Finally, I would like to convey my heart full thanks to all Technical Staff, for theirguidance and support in every step of this project. I convey my sincere thanks to all the faculty and friends who directly or indirectly helped me for the successful completion of this project.

**Project Associate**
**L.SRAVANI**
**(22X41F0034)**

# TABLE OF CONTENTS

LIST OF FIGURES

# ABSTRACT

Medicinal plants (herbs) are plants that are known to have certain compounds which are nutritiousfor health. The human body is complex and organic, while chemical medicines contain chemicals that are inorganic and pure. Therefore, chemical medicine are considered not very suitable for consumption by the human body, which if consumed continuously can even be bad for human health. However, some chemical drugs are actually symptomatic (temporary) so they must be takenfor life by patients with certain diseases.

Therefore a system is needed to be able to help the community to recognize medicinal plants better,in this case the medicinal plants are focused on the introduction of medicinal leaves. In this study identification of medicinal plant leaves was carried out using the Convolutional Neural Network method.

This research will build a system of identification of medicinal plant leaves by using Convolutional Neural Networks. Using training data that is carried out in a computer set and then implemented in mobile-based software to recognize the types and benefits of medicinal plant leaves identified

# INTRODUCTION

Medicinal plants (herbs) are plants that are known to have certain compounds that are nutritious for health. Each part of the medicinal plant is believed to have various properties to prevent, diverse or even cure a certain disease. In Indonesia there are 30,000 types of plants and 7000 of them are classified as medicinal plants (herbs). In addition to the many presence of these medicinal plants, the use of medicinal plants is also considered to be safer because it has natural ingredients comparedto chemical medicine. To find out the content of medicinal plants can be done by phytochemical screening methods, so it can be known active compounds that are in certain plants that can be usefulas a madicine.

The human body is complex and organic, while chemical medicines contain chemicals that are inorganic and pure. Therefore, chemical medicine are considered not suitable for consumption by the human body, which if consumed continuously can even be bad for human health. However, some chemical drugs are actually symptomatic (temporary) so they must be taken for life by patientswith certain diseases. The number of medicinal plants is still not balanced with the public's knowledge about the medicinal plants themselves, so many people prefer chemical medicines because they are considered more practical and easy to obtain. Therefore a system is needed to be able to help the community to recognize medicinal plants better, in this case the medicinal plants are focused on the introduction of medicinal leaves. Leaves can be identified from image images by color, size, texture and shape using various methods including Neural Network[8],[9], [10].

Previous studies identified herbal medicinal plants based on leaf imagery using Artificial Neural Networks (ANN) Gray Level Co-occurrence Matrix and K-Nearest Neighbor Algorithms[3], Local Binary Patterns, Support Vector Machines, Multilayer Perceptron (MPL). The leaves to be identified are medicinal plant leaves which are often used for hypertension medicine, including bayleaves, avocado leaves, cat's whiskers leaves, celery leaves, soursop leaves, african leaves, starfruitleaves, grass jelly leaves, and betel leaves. These leaves were chosen because hypertension is a health problem that is considered serious and suffered by most people

In this study identification of medicinal plant leaves was carried out using the Convolutional Neural Network method. CNN is one of the algorithms from the branch of Machine Learning that is based on Artificial Neural Networks (ANN) or its development, namely Deep Learning which is a development of the Multilayer Perceptron (MPL) to process two-dimensional data, one of them is image[16]. CNN is used in image data to detect and recognize objects in an image, with

Backpropagation type training[17]. The way CNN works is similar to MLP, but in CNN each neuron propagated on the network has a two-dimensional shape, so that the weight and linear operating parameters on CNN are different.

## 1.1 Problem Statement

Identifying medicinal plants accurately is essential for various applications in healthcare, pharmacology, and botany. However, manual identification methods can be time-consuming and prone to errors, especially for non-experts. There is a need for an automated system that can reliably identify medicinal plants based on visual characteristics, enabling faster and more accurate identification processes.

# 1. LITERATURE SURVEY

**[1] TITLE: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network**

**AUTHOR:** Forrest Sheng Bao; Eric You Xu; Yu-Xuan Wang; Yi-Fan Chang; Qiao-Liang Xiang

In this paper, we employ probabilistic neural network (PNN) with image and data processing techniques to implement a general purpose automated leaf recognition for plant classification. 12 leaf features are extracted and orthogonalized into 5 principal variables which consist the input vector of the PNN. The PNN is trained by 1800 leaves to classify 32 kinds of plants with an accuracy greater than 90%. Compared with other approaches, our algorithm is an accurate e artificial intelligence approach which is fast in execution and easy in implementation.

**[2] TITLE**: **Vnplant-200–a public and large-scale of vietnamese medicinal plant images dataset**

**AUTHOR**: Trung Nguyen Quoc and Vinh Truong Hoang.

Plant identification is an essential topic in computer vision with various applications such as agronomy, preservation, environmental impact, discovery of natural and pharmaceutical Product. However, the standard and available dataset for medicinal plants have not been widely published for research community. This work contributes the first large, public and multi class dataset of medicinal plant images. Our dataset consists of total 20,000 images of 200 different labeled Vietnamese medicinal plant (VNPlant-200). We provide this dataset into two versions of size 256 × 256 and 512 × 512 pixels. The training set consists of 12,000 images and the remainder are used for testing set. We apply the Speed-Up Robust Features (SURF) and Scale Invariant Feature Transform (SIFT) for extracting features and the Random Forest (FR) classifier is associated to recognize plant. The experimental results on the VNPlant-200 have been shown the interesting challenge task for pattern recognition.

**[3] TITLE**: **Local binary pattern based on image gradient for bark image classification**

**AUTHOR**: Tuan Le-Viet and Vinh Truong Hoang.

In this work, we present a discriminative and effective local texture descriptor for bark image classification. The proposed descriptor is based on three factors, namely, pixel, magnitude and direction value. Unlike most other descriptors based on original local binary pattern, the proposed descriptor is conducted the changing of local texture of bark image. The performance of the proposed descriptor is evaluated on three benchmark datasets.

**[4] TITLE**: **Bark texture classification using improved local ternary patterns and multilayer neural network**

**AUTHOR:** Shervan Fekri-Ershad

Tree identification is one of the areas that are regarded by researchers. It is done by human expert with high cost. Experts believe that tree bark has a high relation with species in comparison with other phenotype properties. Repeated textures in the bark is usually various with slight differences. So, lbp-like descriptors used in most recent works. But, most of them do not provide discriminative features. Also some texture descriptors are sensitive to noise and rotation. Local ternary pattern is one of the operators that are resistant to the noise with high discrimination. In most of descriptors, histogram of patterns is used to extract features. But, it is rotation sensitive with high computational complexity. In this paper, the main contribution is to propose a method for bark texture classification with high accuracy based on the improved local ternary patterns (ILTP). In the proposed ILTP, the ternary patterns are coded into two binary patterns, and then each one is classified into two uniform/non-uniform groups. The extracted patterns are labeled according to the degree of uniformity. Finally the occurrence probability of the labels is extracted as features. Also, a multilayer perceptron is designed with four theories in the number of hidden nodes. Experimental results on two benchmark datasets showed that our proposed approach provides higher classification accuracy than most well known methods. Noise-resistant and rotation invariant are other advantages of the presented method. The proposed bark texture classification, because of its high classification accuracy, can be applied in real applications and reduce the financial costs and human risks in the diagnosis of plant species

**[5] TITLE**: **Combining sparse representation and singular value decomposition for plant recognition**.

**AUTHOR**: Shanwen Zhang, Chuanlei Zhang, Zhen Wang, and Weiwei Kong.

Plant recognition is one of important research areas of pattern recognition. As plant leaves are extremely irregular, complex and diverse, many existing plant classification and recognition methods cannot meet the requirements of the automatic plant recognition system. A plant recognition approach is proposed by combining singular value decomposition (SVD) and sparse representation (SR) in this paper. The difference from the traditional plant classification methods is that, instead of establishing a classification model by extracting the classification features, the proposed method directly reduces the image dimensionality and recognizes the test samples based on the sparse coefficients, and uses the class-specific dictionary learning for sparse modeling to reduce the recognition time. The proposed method is verified on two plant leaf datasets and is

the proposed approach for the 6 kinds of plant leaves is over 96%, which is the best classification rate. The experimental results show the feasibility and effectiveness of the proposed method.

**[6] TITLE: Integrating leaf and flower by local discriminant CCA for plant species recognition.**

**AUTHOR:** Shanwen Zhang, Chuanlei Zhang, and Wenzhun Huang.

Plant species recognition using a single organ, such as flower and leaf, is not sufficiently reliable, because different species may have very similar flowers or leaves, while the same species may have rather different flowers or leaves. Combining leaves and flowers to recognize plant species can produce positive results. Based on multi-modal learning scheme, an automatic plant species recognition method is proposed by combining leaves and flowers of plant. In the method, a modified local discriminant canonical correlation analysis (MLDCCA) is designed by incorporating the idea of local discriminant embedding (LDE) into canonical correlation analysis (CCA). Firstly, two neighbor graphs are constructed based on the exploration of the manifold that the input data lie on. Then, two projection matrices for dimensionality reduction are obtained by making the within-class neighbor samples most correlated and between-class neighbor samples least correlated, and meanwhile keeping the correlation between leaves and flowers of the same species maximum. Finally, 1-nearest neighbor classifier with geodesic distance is used to recognize the plant species. MLDCCA is a powerful supervised multi-modal dimensional reduction method which can extract the discriminant features from two plant organs, meanwhile preserve the discriminant information and the data structure well. Experimental results on a real leaf and flower image dataset validate the effectiveness of the proposed method.

**[7] TITLE: Plant leaf classification using GIST texture features**

**AUTHOR:** Fateme Mostajer Kheirkhah and Habibollah Asghari.

The leaves of plants have rich information in recognition of plants. In general, agriculture experts accomplish information extraction from the leaves. Since the leaves contain useful features for recognising various types of plants, so these features can be extracted and applied by automatic image recognition algorithms to classify plant species. In this study, the authors investigate a novel approach for recognition of plant species using GIST texture features. Then, the principal and suitable features are selected by principal component analysis (PCA) algorithm. In the classification step, three different approaches such as Patternnet neural network, support vector machine, and K-nearest neighbour (KNN) algorithms were applied to the extracted features. For evaluation of the authors' approach, they applied their proposed algorithm on three famous datasets. In comparison to some widely used features, the results show that their approach outperforms the other methods in the case of the time and the accuracy.

# 2. PROJECT DESCRIPTION

## 3.1 Objective of the project

The objective of this project is to develop a robust system for the identification of medicinal plants using Convolutional Neural Networks (CNNs). The system will take input images of plant specimens and classify them into corresponding medicinal plant species categories. The specific objectives include:

Collecting a comprehensive dataset of images containing various medicinal plant species, including leaves, flowers, and other plant parts.

Preprocessing the dataset to ensure uniformity in size, resolution, and quality, and splitting it into training, validation, and testing sets.

Designing and training a CNN model architecture capable of accurately classifying medicinal plant species based on their visual characteristics.

Evaluating the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score.

Fine-tuning the model and incorporating techniques such as data augmentation to improve its performance and generalization capabilities.

Developing a user-friendly interface for the system, allowing users to input images of plant specimens and receive accurate identification results.

Testing the system with real-world data and gathering feedback for further refinement and optimization.

Deploying the system as a standalone application or integrating it into existing platforms for plant identification and classification in healthcare, pharmaceutical, and botanical research settings.

By achieving these objectives, the project aims to provide a valuable tool for researchers, botanists, healthcare professionals, and enthusiasts to accurately identify medicinal plants, contributing to advancements in natural medicine, biodiversity conservation, and sustainable healthcare practices.

## 3.2 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 Feasibility

### 3.2.1 Economical 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 ha d to be purchased.

### 3.2.2 Technical 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.

### 3.2.3 social 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, which is welcomed, as he is the final user of the system.

## 3.3 System Requirement Specification

### 3.3.1 Functional Requirement

- Graphical User interface with the User.

### 3.3.2 Hardware Requirement:

**System** : Pentium IV 2.4 GHz.

**Hard Disk** : 40 GB.

**Floppy Drive** : 1.44 Mb.

**Monitor** : 14’ Colour Monitor.

**Mouse** : Optical Mouse.

**Ram** : 512 Mb.

### 3.3.3 Software Requirement

**Operating system** : Windows.

**Coding Language** : Python.

**Designing** : Html, Css, Javascript.

**Data Base** : MySQL.

**Framework** : Django

## 3.4 Existing system

Previous studies identified herbal medicinal plants based on leaf imagery using Artificial Neural Networks (ANN), Gray Level Co-occurrence Matrix and K-Nearest Neighbor Algorithms[3], Local Binary Patterns[13], Support Vector Machines, Multilayer Perceptron (MPL). The leaves to be identified are medicinal plant leaves which are often used for hypertension medicine, including bay leaves, avocado leaves, cat's whiskers leaves, celery leaves, soursop leaves, african leaves, starfruit leaves, grass jelly leaves, and betel leaves. These leaves were chosen because hypertension is a health problem that is considered serious and suffered by most people

### 3.4.1 Drawbacks of existing system

- Using machine learning algorithm will not provide better identification of Medical plants
- These wrong identification may becomes serious problem if they take any medical decision with the plant or leaves

## 3.5 Proposed System

In this proposed system we will identify medicinal plant/leaves was carried out using the Convolutional Neural Network method. CNN is one of the algorithms from the branch of Machine Learning that is based on Artificial Neural Networks (ANN) or its development, namely Deep Learning which is a development of the Multilayer Perceptron (MPL) to process two-dimensional data, one of them is image. CNN is used in image data to detect and recognize objects in an image, with trained models.

### 3.5.1 Advantages of proposed system

1. We are using deep learning CNN algorithm to get the better accuracy
2. Using the CNN model we can accurately identify medicinal plant

# 4. SYSTEM DESIGN

## 4.1 Modules

**UPLOAD DATASET**

Using this module we can load medicinal plant dataset from the location of the project to Train the CNN algorithm

**GENERATE TRAINING AND TESTING IMAGES**

**ImageDataGenerator**: that rescales the image, applies shear in some range, zooms the image and does horizontal flipping with the image. This ImageDataGenerator includes all possible orientation of the image.

**train_datagen.flow_from_directory** is the function that is used to prepare data from the train_dataset directory Target_size specifies the target size of the image.

**test_datagen.flow_from_directory** is used to prepare test data for the model and all is similar as above.

**fit_generator** is used to fit the data into the model made above, other factors used are steps_per_epochs tells us about the number of times the model will execute for the training data.

**epochs** tells us the number of times model will be trained in forward and backward pass.

1. **GENERATE CNN MODEL**

   In this module we are generating CNN Model with train_datagen and test_datagen generated by ImageDataGenerator class.

   Here we have training this CNN algorithm multiple time to get the better accuracy using epochs.

   Finally we will get the best CNN model with average accuracy above 90%

2. **UPLOAD TEST IMAGE**

   Using this module we can upload test image AND pass the test image to the model to identify medicinal plant

3. **IDENTIFY MEDICINAL PLANT**

   Using this model will call the CNN Model which is already generated and take the image from the $4^{th}$ step and pass to model.Then the model will identify the medicinal plant.

# 5. INPUT AND OUTPUT DESIGN

## 5.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

## Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## 5.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output.It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship

to help use decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

# 6. SYSTEM ARCHITECRUTE

## 6.1 UML Diagrams

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

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

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

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

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

### Goals:

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

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

## 6.2 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 diagramis to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

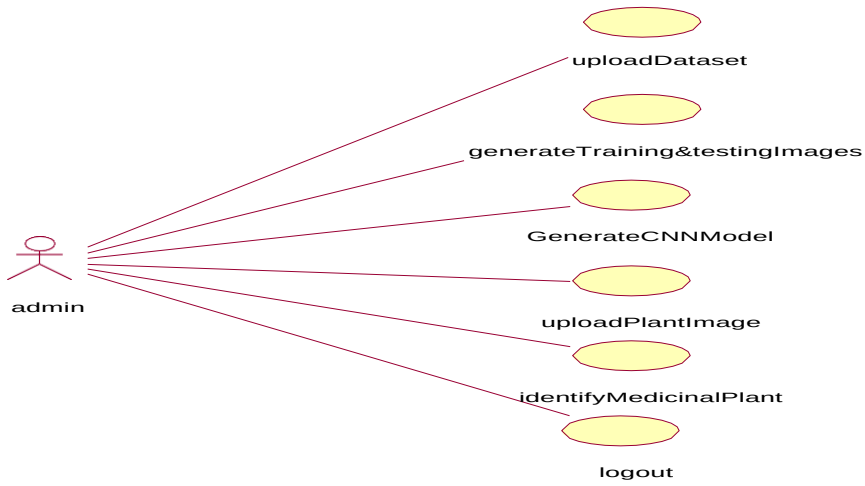



**FIG:6.2 Use Case Diagram**

### 6.2 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 whichclass contains information.

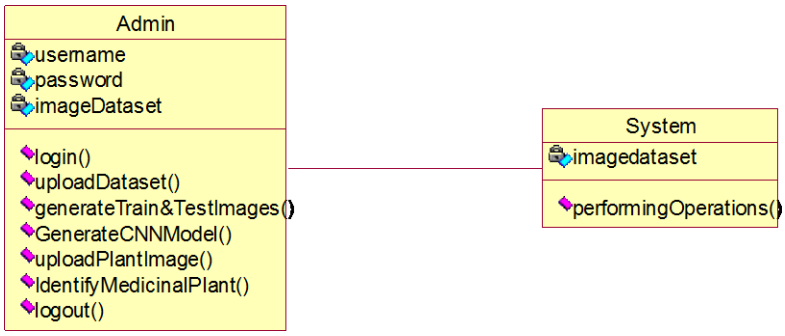



**Fig:6.3 Class Diagram**

### 6.3 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that hows 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.

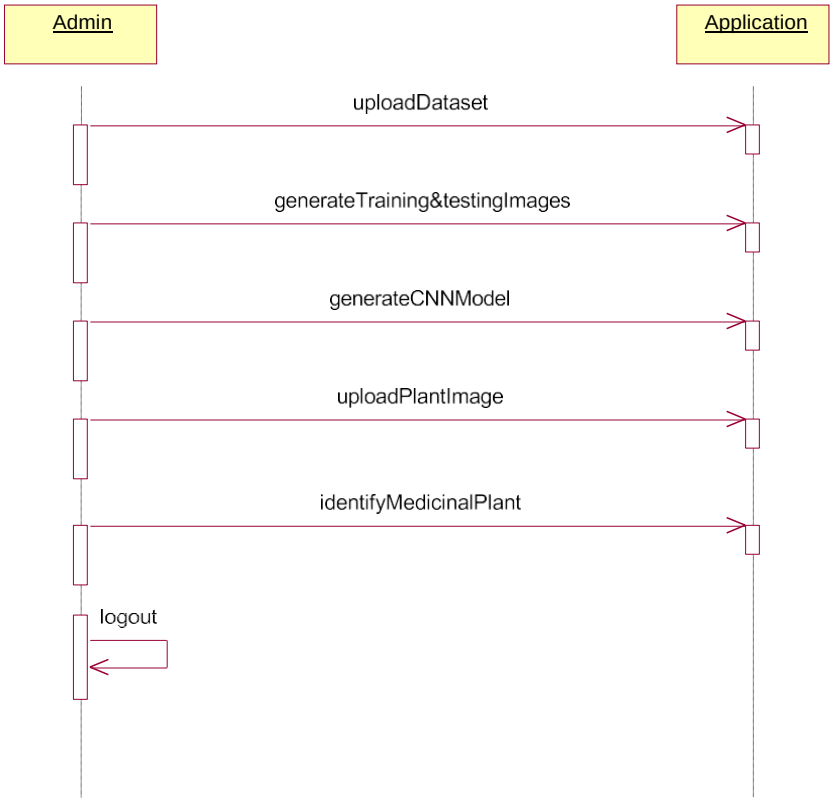
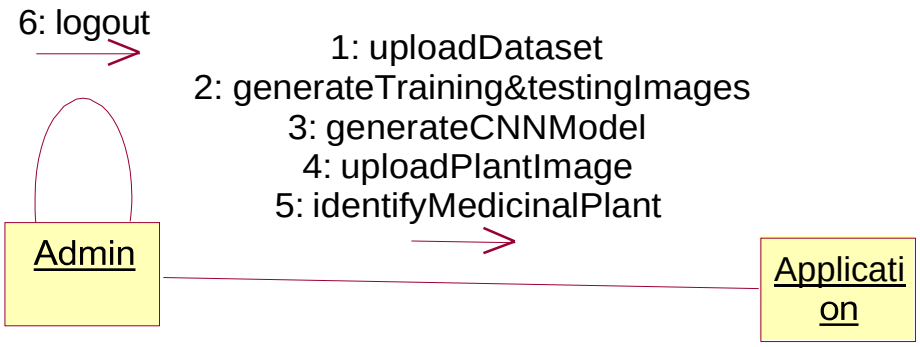



**FIG:6.4 Sequence Diagram**

### 6.4 Collaboration Diagram:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



# 7. TECHNICAL DESCRIPTION

## 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**

Comimplimentary 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.*

## 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.

**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

## 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[1], 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."

## 7.2 Machine Learning

## 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.

## Categories Of Machine Leaning :-

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, particularlyto 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 ahuge scale. That is why the need for machine learning arises.

## Challenges in Machines 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.

## Applications of Machines 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 it's 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 a predicted output(label).

## (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 unlabelled 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 unlabelled 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.

## 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

## 7.3 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**

- 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.

## 7.4 Django – Design Philosophies

Django comes with the following design philosophies −

**Loosely Coupled** − Django aims to make each element of its stack independent of the others.

**Less Coding** − Less code so in turn a quick development.

**Don't Repeat Yourself (DRY)** − Everything should be developed only in exactly one place instead of repeating it again and again.

**Fast Development** − Django's philosophy is to do all it can to facilitate hyper-fast development.

**Clean Design** − Django strictly maintains a clean design throughout its own code and makes it easy to follow best web-development practices.

**Advantages of Django**

Here are few advantages of using Django which can be listed out here −

**Object-Relational Mapping (ORM) Support** − Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

**Multilingual Support** − Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.

**Framework Support** − Django has built-in support for Ajax, RSS, Caching and various other frameworks.

**Administration GUI** − Django provides a nice ready-to-use user interface for administrative activities.

**Development Environment** − Django comes with a lightweight web server to facilitate end-to-end application development and testing.

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django’s specificity for the Model-View-Template (MVT) pattern.

**MVC Pattern**

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. Check our MVC tutorial here to know more.

**Django MVC – MVT Pattern**

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template.

The template is a HTML file mixed with Django Template Language (DTL)The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a use

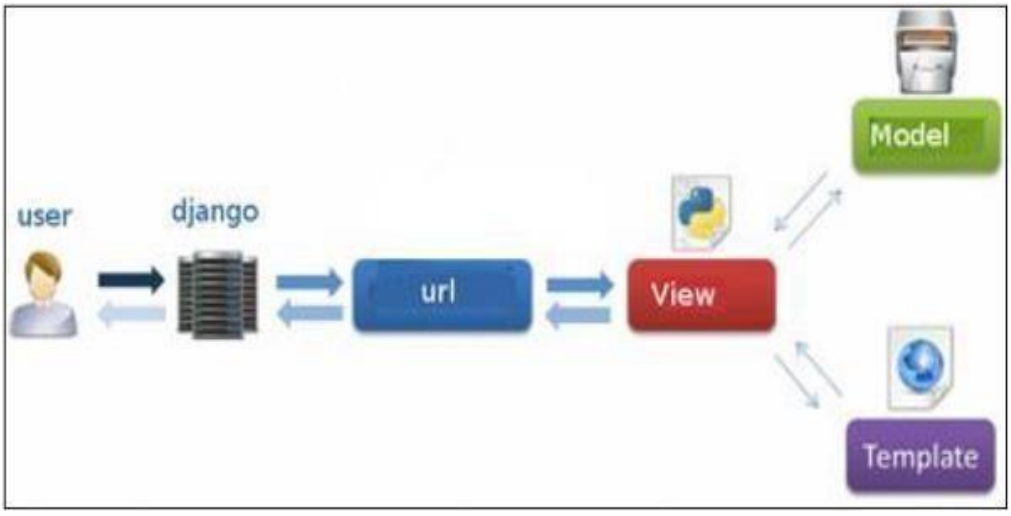



**Fig 7.4: Django MVC – MVT Pattern**

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

# 8. ALGORITHM

## 8.1 Convolutional Neural Network :

A CNN is a kind of network architecture for deep learning algorithms and is specially used for image recognition and tasks that involve the processing of pixel data . There are other types of neural networks in deep learning ,but for identifying and recognizing objects ,CNN are the network architecture of choice.

It is inspired by the structure of the visual cortex in the brain - that consists the cells sensitive to small regions of the visual field.
CNNs use convolution process for extracting features from the input data (generally an image). Then a set of filters or kernels are applied to the input data and feature maps are produced that highlight the different aspects of inputs, such as edges, patterns and texture.
The feature maps are then passed through a non-linear activation function to bring non-linearity into the network.

Top applications of CNNs are image classification, semantic segmentation, object detection, face recognition, speech recognition, etc.

Here's some more information about Convolutional Neural Networks (CNNs)

### Local Connectivity And Shared Weights:

CNNs exploit the idea of local connectivity, where each neuron in a layer is connected to onlya small region of the input data. This arrangement allows the network to capture local patterns effectively. Additionally, CNNs use shared weights, meaning the same filter is applied to different regions of the input. This reduces the number of parameters and enables the networkto learn spatially invariant features.

### Feature Maps:

CNNs produce feature maps as intermediate representations. Each feature map represents the presence or activation of a particular feature learned by the network. Feature maps of higher layers capture more complex and abstract features as they progress through the network.

CNNs produce feature maps as intermediate representations. Each feature map represents the presence or activation of a particular feature learned by the network. Feature maps of higher layers capture more complex and abstract features as they progress through the network.

## Stride And Padding:

Stride determines the step size at which the filters move over the input during the convolution operation. It affects the spatial dimensions of the output feature maps. Padding is the process of adding extra pixels around the input, typically with zero values, to preserve spatial dimensions during convolutions and prevent information loss at the boundaries.

## Pooling:

Pooling layers reduce the spatial size of the feature maps while retaining the most important information. The most common type of pooling is max pooling, which selects the maximum value from a region of the feature map. Pooling helps achieve spatial invariance, making the network more robust to small translations or distortions in the input.

## Convolutional Neural Network Architectures:

Several CNN architectures have been proposed, each with its own unique design. Notable architectures include LeNet-5, AlexNet, VGGNet, GoogLeNet (Inception), ResNet, and DenseNet. These architectures vary in terms of the number of layers, filter sizes, and network depth, enabling them to handle different tasks and achieve state-of-the-art performance in various domains.

## GPU Acceleration:

Training CNNs can be computationally intensive due to the large number of parameters and complex operations involved. Graphics Processing Units (GPUs) are commonly used to accelerate the computations and speed up training times. GPUs excel at parallel processing, making them well-suited for deep learning tasks.

## Interpretability And Visualization:

CNNs have been criticized for being black boxes, as it can be challenging to interpret how they make predictions. However, techniques like gradient-based visualization and activation maximization can help visualize the learned features and understand which parts of the input contribute to the network's decision-making process.

Convolutional Neural Networks have revolutionized the field of computer vision and have demonstrated exceptional performance in various domains. Their ability to automatically learn hierarchical representations from raw input data, combined with their structural design, makes them highly effective for tasks such as image recognition, object detection, and many other applications involving grid-like data.

CNN, which stands for Convolutional Neural Network, is a deep learning algorithm widely used for image recognition, computer vision tasks, and other tasks that involve analyzing grid-like data, such as time series analysis and natural language processing.

## 8.2 Key Points About CNN :

### Architecture:

CNNs are designed to automatically and adaptively learn hierarchical representations from input data. They consist of multiple layers, including convolutional layers, pooling layers, andfully connected layers.

### Convolutional Layers:

These layers apply filters (also known as kernels) to the input data, which helps extract relevant features. The filters slide over the input using a process called convolution, and theycapture different patterns such as edges, textures, or shapes.

### Pooling Layers:

These layers reduce the spatial dimensions of the feature maps generated by the convolutional layers. They achieve this by downsampling the input, typically through operations like max pooling, which retain the most dominant values in a given region.

### Fully Connected Layers:

These layers connect all neurons from the previous layer to the next layer. They perform the final classification or regression based on the features learned by the preceding layers.

## Activation Functions:

Non-linear activation functions like ReLU (Rectified Linear Unit) are commonly used in CNNs to introduce non-linearity into the network, enabling it to learn complex patterns and make more accurate predictions.

## Training:

CNNs are trained using backpropagation, which involves iteratively updating the weights of the network to minimize the difference between predicted and target outputs. This process is typically performed using optimization algorithms like Stochastic Gradient Descent (SGD) orits variants.

## Transfer Learning:

CNNs can benefit from transfer learning, where a pre-trained model on a large dataset is usedas a starting point for a new task. By leveraging the learned features, it allows for effective training even with limited data.

## Data Augmentation:

To overcome overfitting and improve generalization, data augmentation techniques are often applied to CNNs. These techniques involve creating new training samples by applying random transformations to the existing data, such as rotations, translations, or flips.

CNNs have achieved remarkable success in various applications, including imageclassification, object detection, facial recognition, medical imaging, and natural language processing tasks such as text classification or sentiment analysis. Their ability to automaticallylearn relevant features from raw input data makes them a powerful tool for analyzing and understanding complex patterns in various domains.

CNN, which stands for Convolutional Neural Network, is a deep learning algorithm widely used for image recognition, computer vision tasks, and other tasks that involve analyzing grid-like data, such as time series analysis and natural language processing.

## Convolutional Neural Networks (CNNs) have become a fundamental tool inthe field of deep learning and offer several important advantages: Hierarchical Feature Learning:

CNNs automatically learn hierarchical representations of the input data. By stacking multiple convolutional layers, the network can learn low-level features like edges and textures in the initial layers and progressively learn more complex and abstract features in the deeper layers. This ability to capture hierarchical representations makes CNNs highly effective in tasks suchas image classification, object detection, and feature extraction

## Spatial Invariance:

CNNs are designed to be spatially invariant, meaning they can recognize patterns regardless of their location in the input data. Convolutional layers employ shared weights and local connectivity, enabling the network to capture and recognize patterns in different regions of the input. Pooling layers further enhance spatial invariance by downsampling the feature mapsand retaining the most dominant features, making the network robust to translations, distortions, or variations in scale and rotation.

## Parameter Sharing:

CNNs use parameter sharing across the convolutional layers, significantly reducing the number of parameters compared to fully connected networks. This sharing of weights allowsthe network to generalize better and learn from limited training data. It also provides a form of translational invariance, as the same filters are applied across the input, enabling the network to recognize the same features regardless of their location.

## Local Receptive Fields:

CNNs focus on local receptive fields, meaning each neuron is connected to only a small regionof the input data. This arrangement allows the network to capture local patterns and reduces the computational requirements compared to fully connected networks that would require connections between all neurons

## Handling Large Data:

CNNs can efficiently handle large input data, such as high-resolution images, due to the shared weights and local connectivity. By applying filters locally, the computational cost remains manageable compared to fully connected networks. Additionally, techniques like pooling and strided convolutions further reduce the spatial dimensions of the data, making it computationally feasible to process large inputs.

## Transfer Learning:

CNNs can leverage transfer learning, which involves using pre-trained models on large datasets as a starting point for a new task or domain. Pre-trained CNN models trained on massive datasets (e.g., ImageNet) have learned generic features that can be fine-tuned or used.

as feature extractors for specific tasks with limited training data. Transfer learning enables faster training and better performance, particularly in scenarios where data is scarce.

## Versatility:

CNNs are versatile and can be applied to various tasks beyond image classification. They have shown success in object detection, semantic segmentation, facial recognition, natural language processing, and time series analysis, among others. Their ability to capture and learnhierarchical representations from input data makes them applicable in numerous domains.

The advantages of CNNs have made them the go-to choice for many computer vision tasks and have played a significant role in advancing the field of deep learning.

Convolutional Neural Networks (CNNs) offer several advantages that have contributed to their widespread use and success in various domain.

## Effective Feature Learning:

CNNs excel at automatically learning hierarchical representations from raw input data. The stacked layers of convolutions, activations, and pooling enable the network to progressively learn and extract low-level, mid-level, and high-level features. This capability allows CNNs to capture complex patterns and discriminative features in the data, making them highly effective for tasks like image classification and object detection.

## Parameter Sharing and Efficiency:

CNNs employ parameter sharing, where the same set of weights is shared across different spatial locations in the input. This sharing dramatically reduces the number of parameters to learn compared to fully connected networks, making CNNs more memory-efficient and enabling training on larger datasets. The local connectivity also ensures that each neuron processes a smaller input region, resulting in reduced computational requirements.

## Handling Large-Scale Data:

CNNs are well-suited for processing large-scale data, such as high-resolution images or videos. The shared weights, local receptive fields, and hierarchical structure make CNNs

capable of effectively handling large inputs without overwhelming computational resources. Techniques like pooling and strided convolutions further reduce spatial dimensions, allowing efficient processing of large volumes of data.

## Interpretable Visualizations:

CNNs can generate visualizations that provide insights into what the network has learned. Techniques like gradient-based visualization and activation maximization allow researchers and practitioners to understand which parts of the input contribute to the network's decision- making process. This interpretability can be valuable for debugging, analysis, and gaining trust in the mod.

# 9. CODE IMPLEMENTATION

```
from django.shortcuts import render
import os
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from django.shortcuts import render
from django.core.files.storage import FileSystemStorage
import tkinter
import numpy as np
import cv2
import imutils
# Create your views here.
def index(request):
  return render(request, 'AdminApp/index.html')
def login(request):
  return render(request, 'AdminApp/Admin.html')
def LogAction(request):
  username = request.POST.get('username')
  password = request.POST.get('password')
  if username == 'Admin' and password == 'Admin':
    return render(request, 'AdminApp/AdminHome.html')
else:
    context = {'data': 'Login Failed .... !!'}
    return render(request, 'AdminApp/Admin.html', context)
def home(request):
  return render(request, 'AdminApp/AdminHome.html')
global dataset
```

```
def loaddataset(request):
   global dataset
   dataset = "dataset\\MedicinalPlant dataset"
   context = {'data': 'Medicinal Plant Dataset Uploaded Successfully..!!'}
   return render(request, 'AdminApp/AdminHome.html', context)
global training_set, test_set
def ImageGenerate(request):
   global training_set, test_set
   train_datagen = ImageDataGenerator(shear_range=0.1, zoom_range=0.1,
horizontal_flip=True)
  test_datagen = ImageDataGenerator()
  training_set = train_datagen.flow_from_directory(dataset,
                    target_size=(48, 48),
                    batch_size=32,
                    class_mode='categorical',
                    shuffle=True)
test_set = test_datagen.flow_from_directory(dataset,
                target_size=(48, 48),
                batch_size=32,
                class_mode='categorical',
                shuffle=False)
    context = {'data': "Generated Training And Testing Images successfully"}
    return render(request, 'AdminApp/AdminHome.html', context)
global classifier
def generateCNN(request):
   global classifier
if os.path.exists("model\\model_weights.h5"):
   classifier = Sequential():
   classifier.add(Convolution2D(32, kernel_size=(3, 3), input_shape=(48, 48, 3),
activation='relu'))
   classifier.add(MaxPooling2D(pool_size=(2,2)
   classifier.add(Convolution2D(32, kernel_size=(3, 3), activation='relu'))
   classifier.add(Flatten())
   classifier.add(Dense(activation="relu", units=128))
```

```
    clasifier.add(Dense(activation="softmax", units=40))
    classifier.load_weights('model/model_weights.h5')
    context = {"data": "CNN Model Generated Successfully.."}
    return render(request, 'AdminApp/AdminHome.html', context)
else:
    classifier = Sequential()
    classifier.add(Convolution2D(32, kernel_size=(3, 3), input_shape=(48, 48, 3),
activation='relu'))
   classifier.add(MaxPooling2D(pool_size=(2, 2)))
   classifier.add(Convolution2D(32, kernel_size=(3, 3), activation='relu'))
   classifier.add(MaxPooling2D(pool_size=(2, 2)))
   classifier.add(Flatten())
   classifier.add(Dense(activation="relu", units=128))
   classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
   classifier.fit_generator(training_set,
                            steps_per_epoch=150,
                            epochs=20,
                            validation_data=test_set,
                            validation_steps=50)
   classifier.save_weights('model/model_weights.h5')
   context = {"data": "CNN Model Generated Successfully.."}
   return render(request, 'AdminApp/AdminHome.html', context)

def uploadPlantImage(request):
    return render(request, 'AdminApp/upload.html')
global filename, uploaded_file_url
def fileUpload(request):
    global filename, uploaded_file_url
    if request.method == 'POST' and request.FILES['myfile']:
    myfile = request.FILES['myfile']
    fs = FileSystemStorage()
    location = myfile.name
    filename = fs.save(myfile.name, myfile)
    uploaded_file_url = fs.url(filename)
```

```
    BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath( file )))
    imagedisplay = cv2.imread(BASE_DIR + "/" + uploaded_file_url)
    cv2.imshow('uploaded Image', imagedisplay)
    cv2.waitKey(0)
    context = {'data': 'Test Image Uploaded Successfully'}
    return render(request, 'AdminApp/upload.html', context)
def IdentifyMedicinalPlant(request):
   BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
   imagetest = image.load_img(BASE_DIR + "/" + uploaded_file_url, target_size=(48, 48))
   imagetest = image.img_to_array(imagetest)
   imagetest = np.expand_dims(imagetest, axis=0)
   loaded_classifier = Sequential()
   loaded_classifier.add(Convolution2D(32, kernel_size=(3, 3), input_shape=(48, 48,
   3), activation='relu'))
   loaded_classifier.add(MaxPooling2D(pool_size=(2, 2)))
   loaded_classifier.add(Convolution2D(32, kernel_size=(3, 3), activation='relu'))
   loaded_classifier.add(MaxPooling2D(pool_size=(2, 2)))
   loaded_classifier.add(Flatten())
   loaded_classifier.add(Dense(activation="relu", units=128))
   loaded_classifier.add(Dense(activation="softmax", units=40))
   loaded_classifier.load_weights('model/model_weights.h5')
   pred = loaded_classifier.predict(imagetest)
   print(str(pred) + " " + str(np.argmax(pred)))
   predict = np.argmax(pred)
   print(training_set.class_indices)
   global msg;
   for x in training_set.class_indices.values():
      if predict == x:
            msg =
            list(training_set.class_indices.keys())[list(training_set.class_indices.values()).index(x
            )]
            print("predicted number: " + str(predict))
            imagedisplay = cv2.imread(BASE_DIR + "/" + uploaded_file_url)
            oring = imagedisplay.copy()
```

```
            output = imutils.resize(oring, width=400)
            data = "Plant Identified As: "+msg
            cv2.putText(output, data, (10, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255,
            0), 2)
             cv2.imshow("Predicted image result", output)
             cv2.waitKey(0)
             context = {'data': data}
return render(request,'AdminApp/AdminHome.html', context)
```

# 10. TEST RESULTS

## 10.1 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, subassemblies, 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.

## 10.2 TYPES OF TESTS

### 10.2.1 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.

### 10.2.2 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.

### 10.2.3 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:

| | |
|---|---|
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| 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.

### 10.2.4 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.

### 10.2.5 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.

### 10.2.6 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..

### 10.3 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.

### 10.3.1 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

### 10.3.2 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.

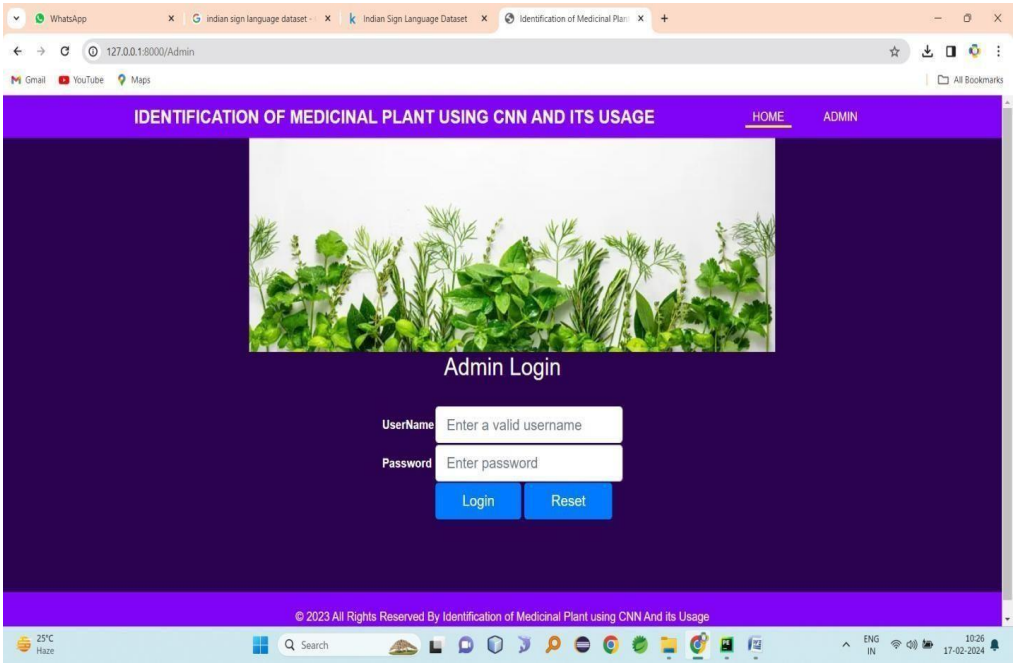
### 10.3.3 Acceptance Testing

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.

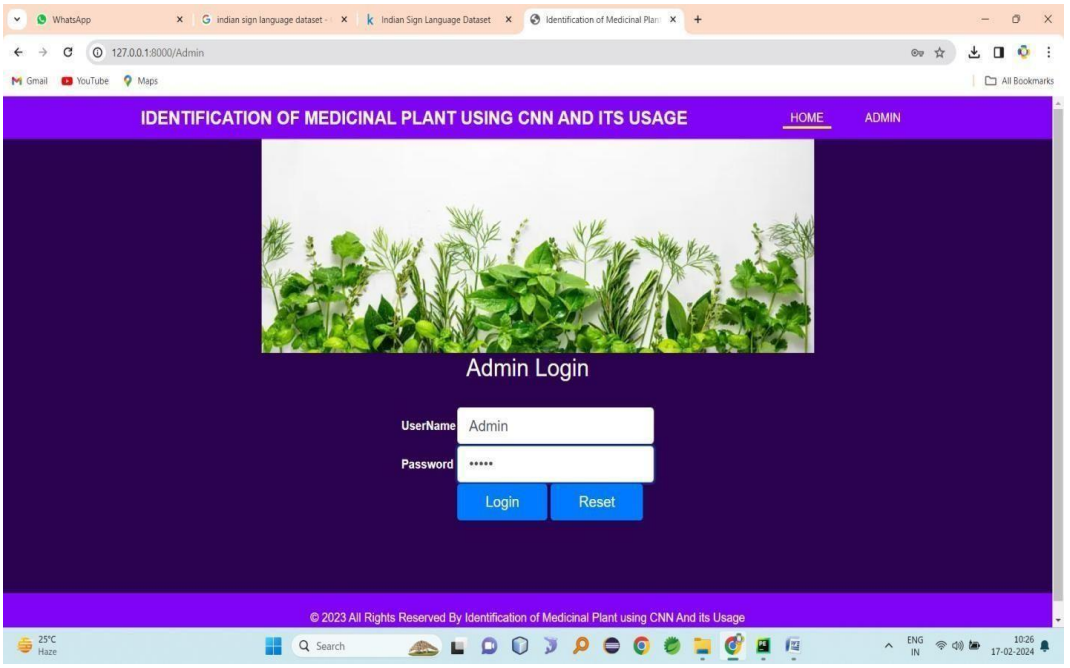
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# RESULTS

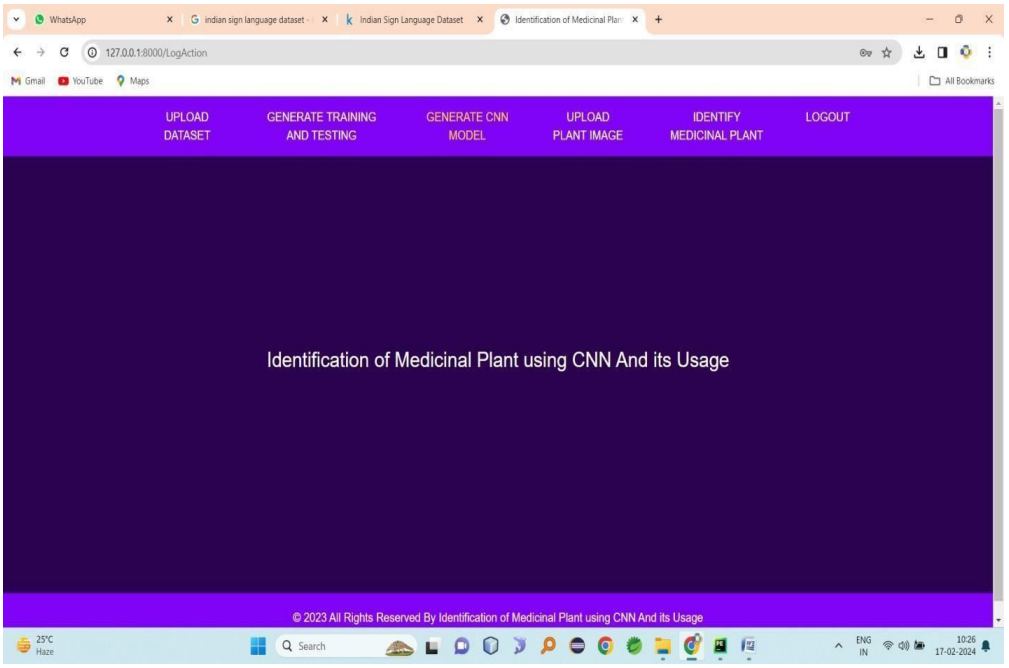
## INDEX



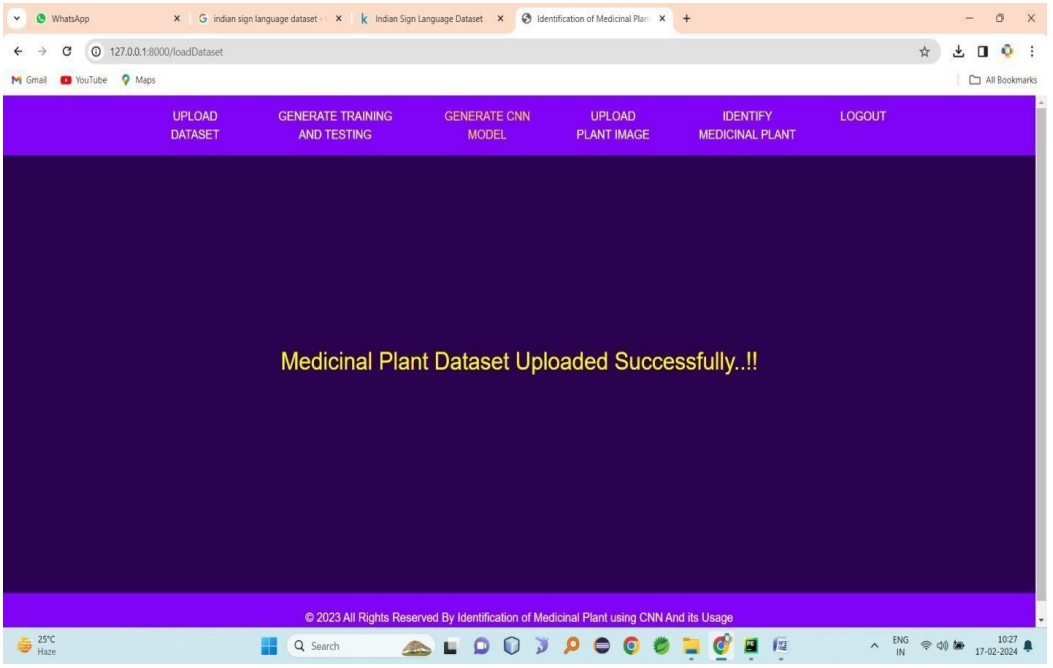
**Login**


IDENTIFICATION OF MEDICINAL PLANT USING CNN AND ITS USAGE
HOME
ADMIN
Admin Login
UserName
Admin
Password
Login
Reset
© 2023 All Rights Reserved By Identification of Medicinal Plant using CNN And its Usage

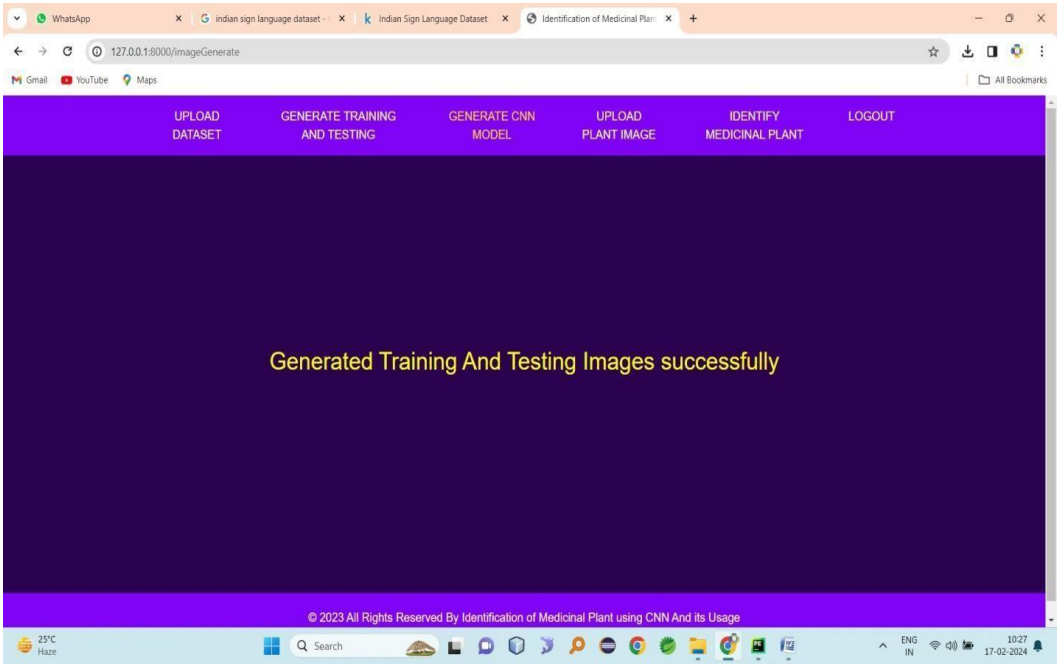
**Home**


WhatsApp
indian sign language dataset
Indian Sign Language Dataset
Identification of Medicinal Plan
127.0.0.1:8000/LogAction
Gmail
YouTube
Maps
All Bookmarks
UPLOAD DATASET
GENERATE TRAINING AND TESTING
GENERATE CNN MODEL
UPLOAD PLANT IMAGE
IDENTIFY MEDICINAL PLANT
LOGOUT
Identification of Medicinal Plant using CNN And its Usage
© 2023 All Rights Reserved By Identification of Medicinal Plant using CNN And its Usage
25°C
Haze
Search
ENG
IN
10:26
17-02-2024

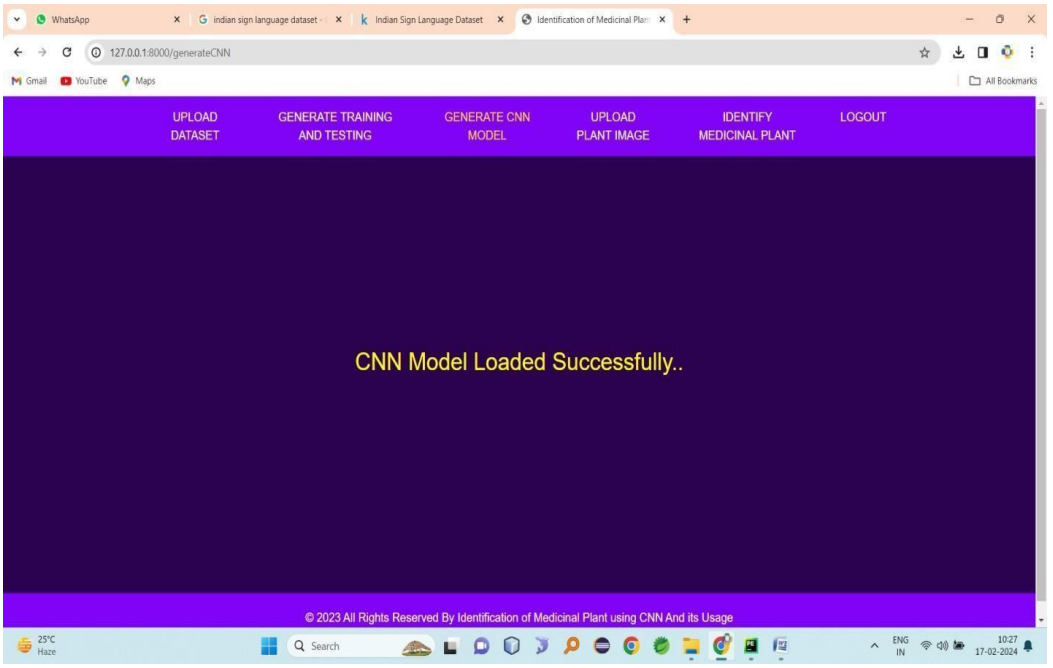
**Upload dataset**



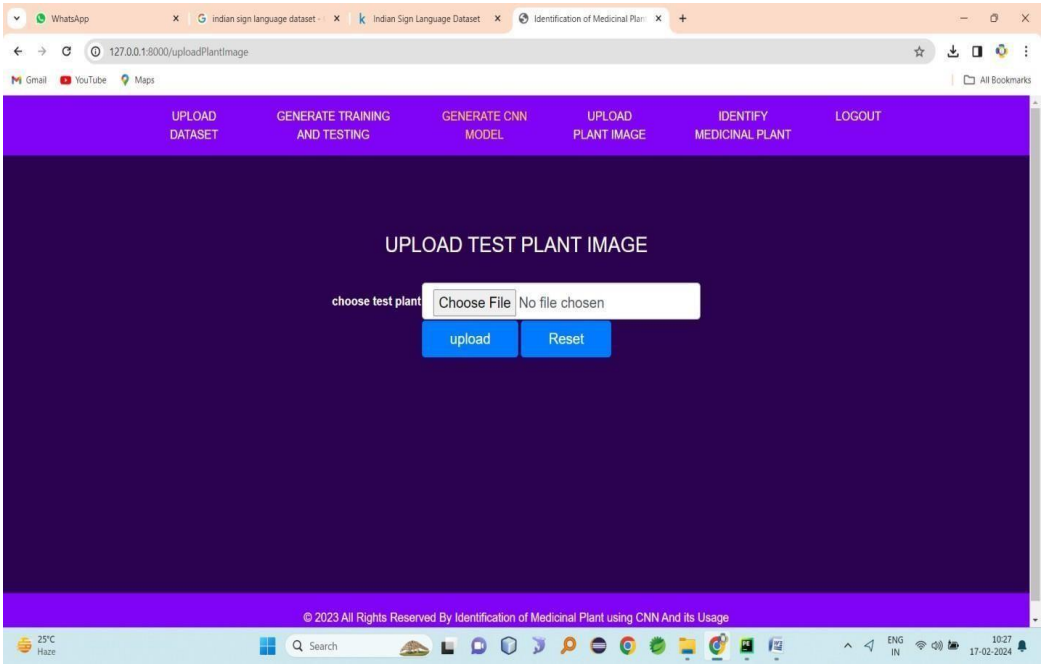
**Generating images as training and testing**



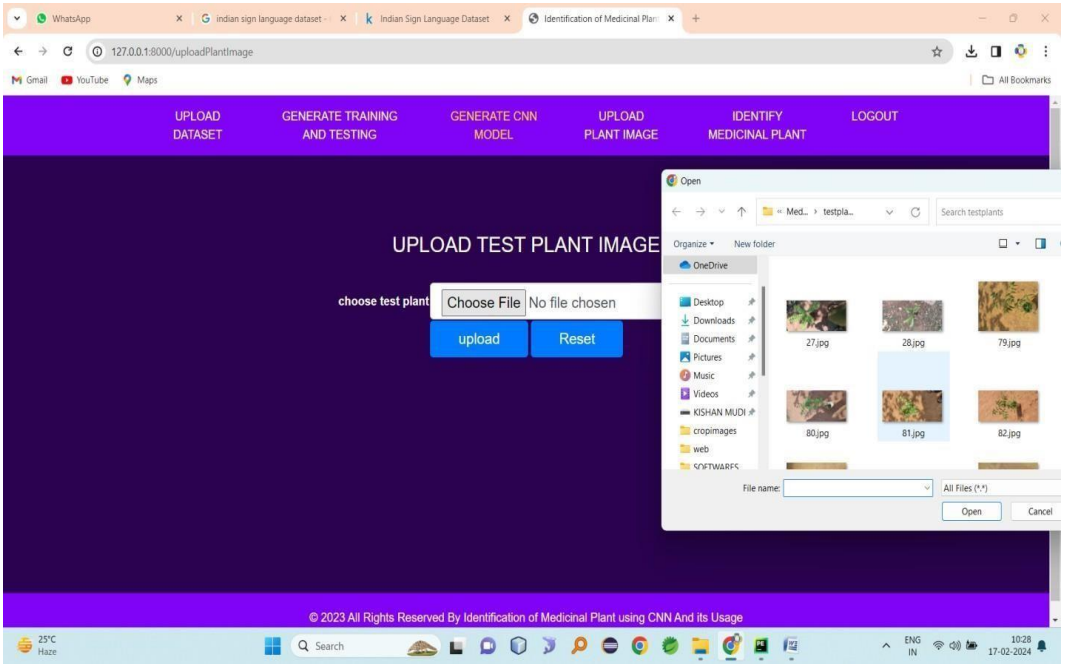
**CNN model generate successfully with training and testing images**



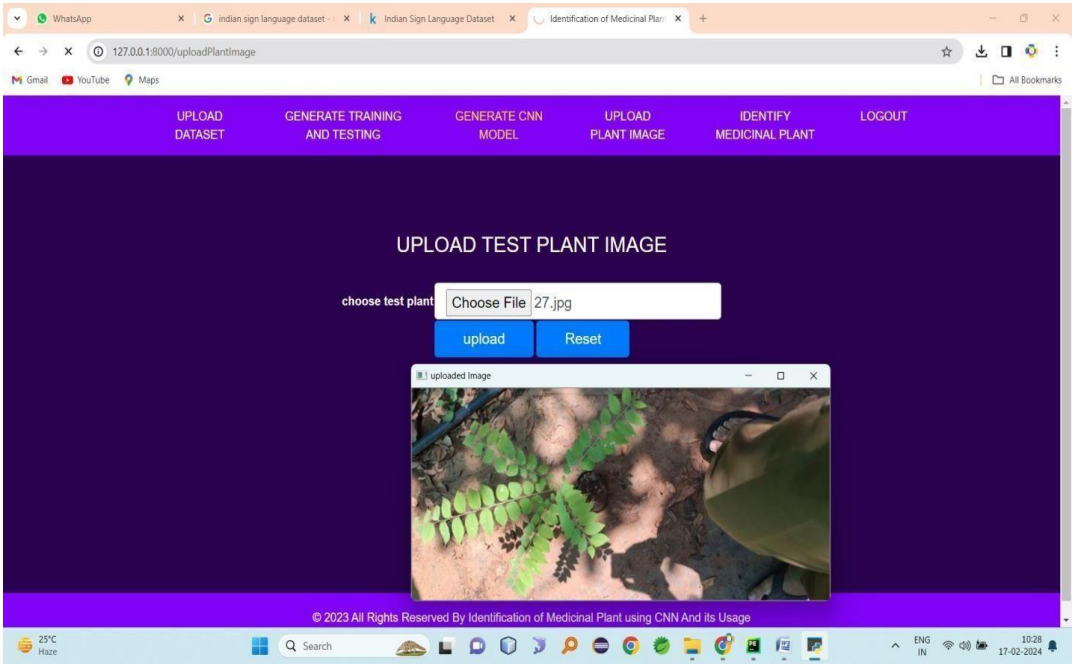
**Choose test image and upload**



**Uploading test plant**



**Uploaded test plant**



**Upload status**

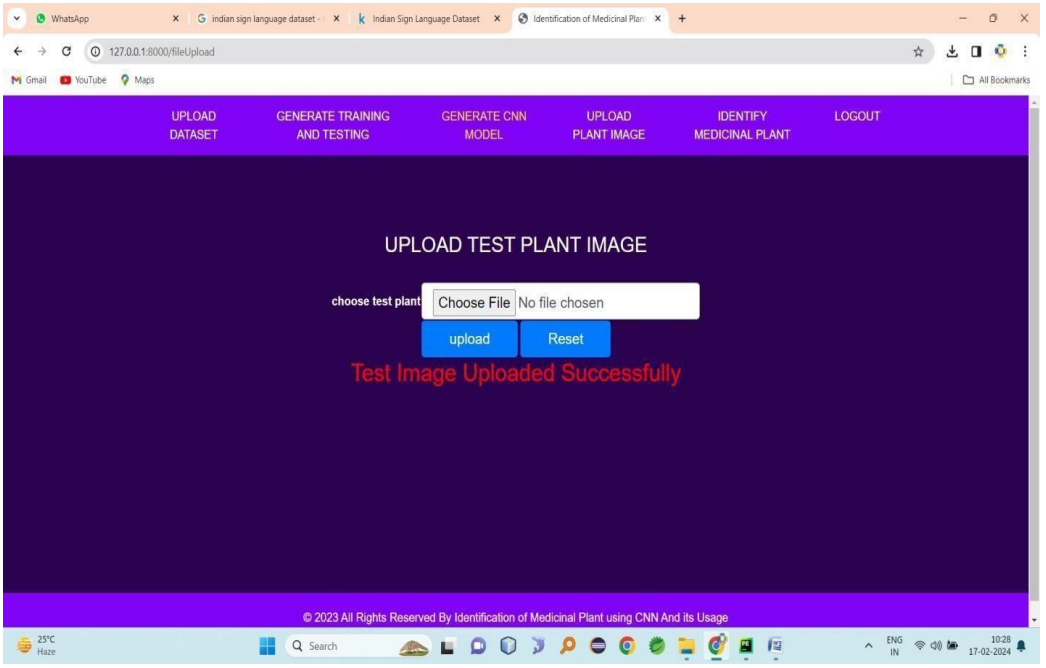
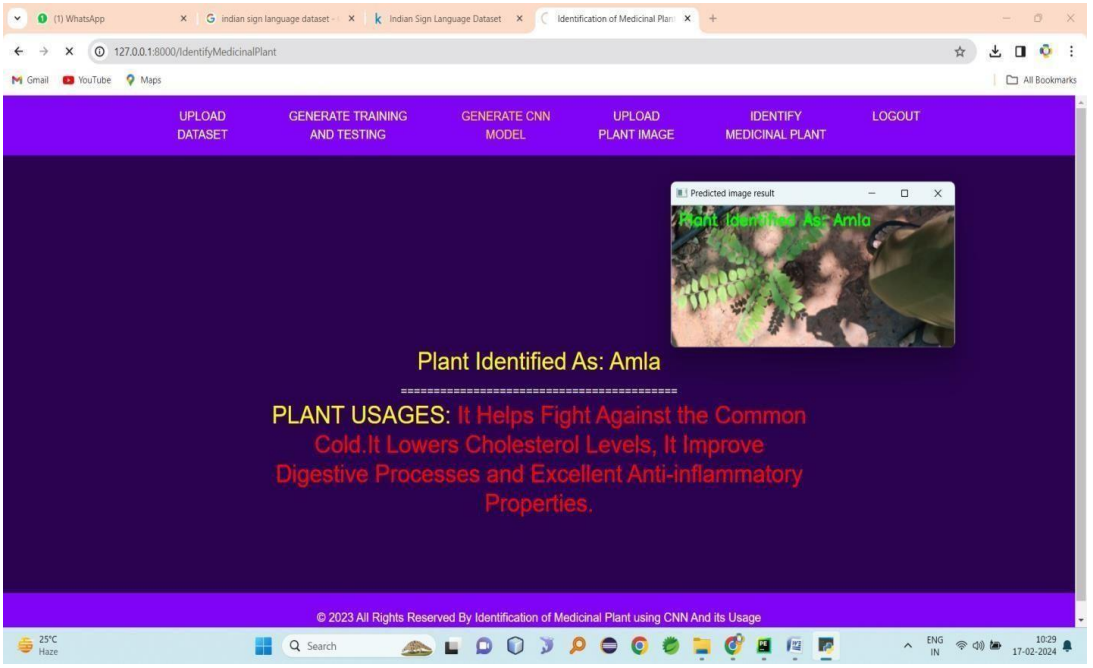
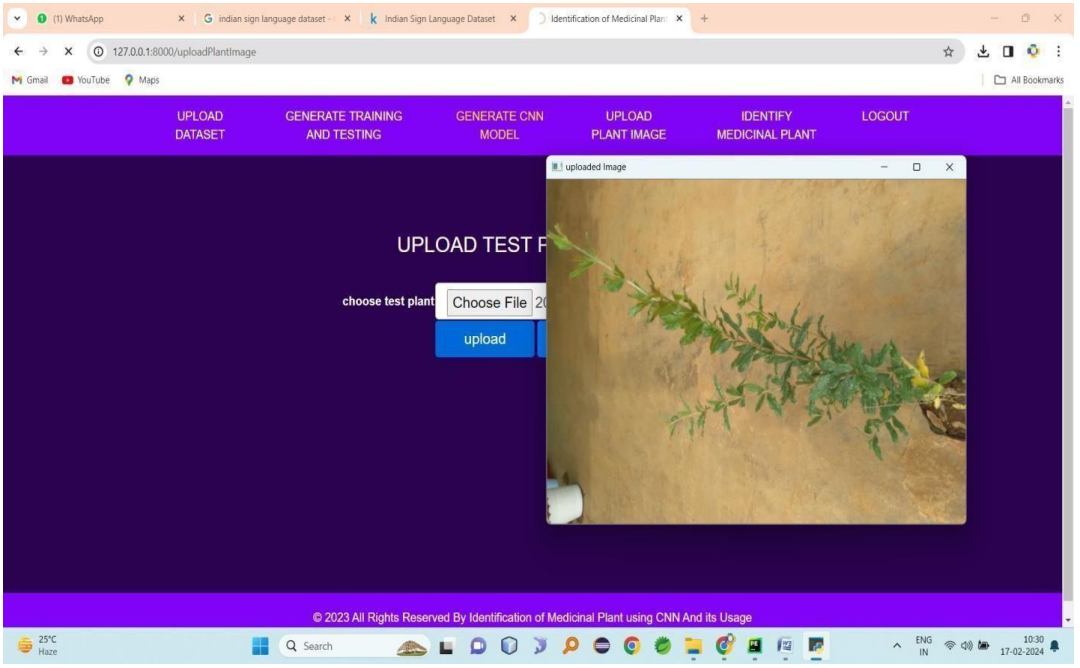


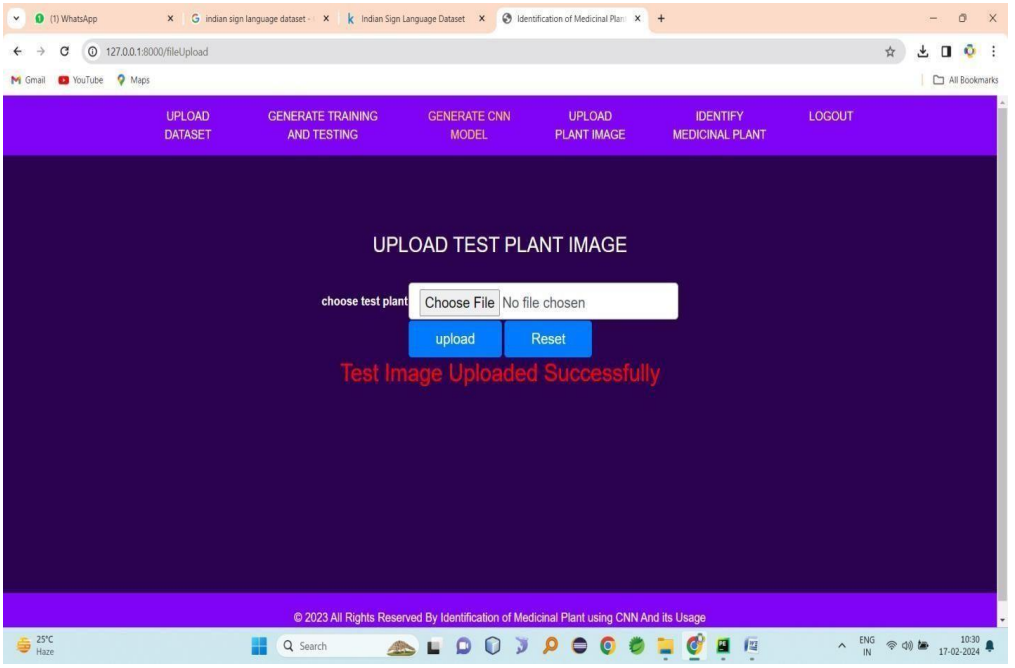
**Image identified**



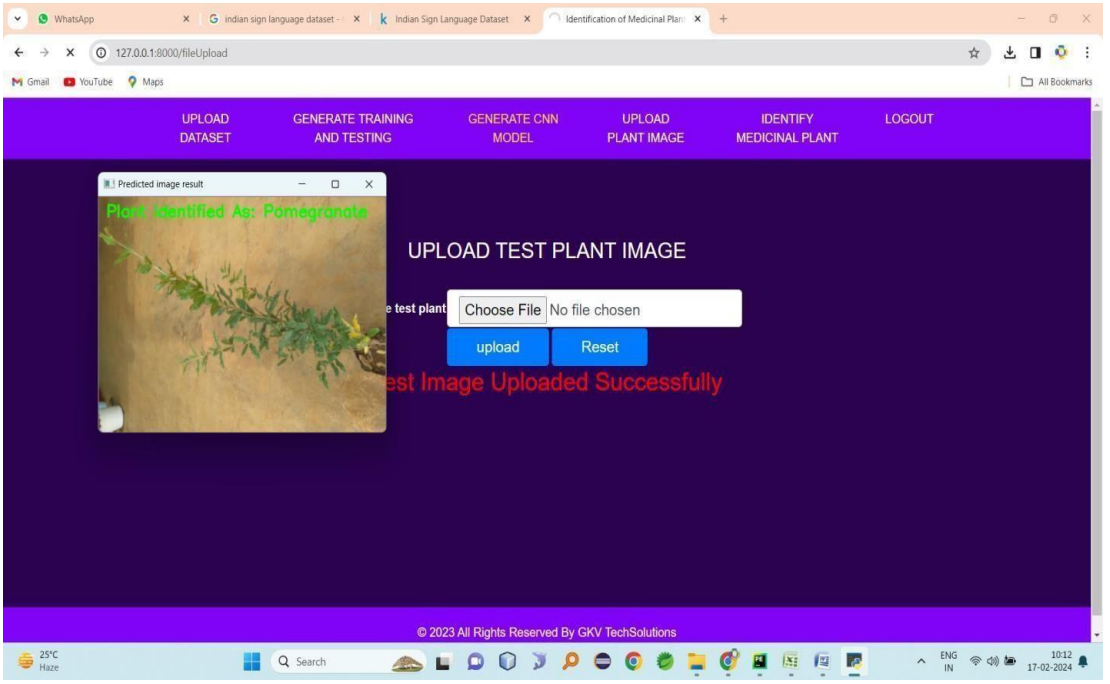
**Upload image**



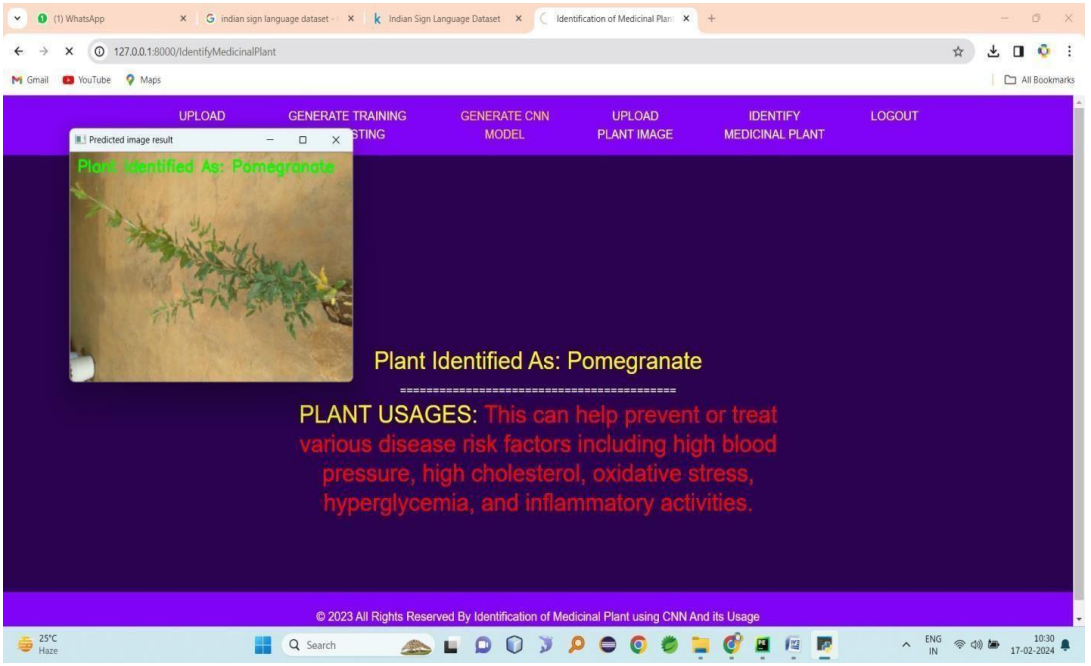
**Upload status**



**Uploaded status**



**Plant usage**


127.0.0.1:8000/IdentifyMedicinalPlant
UPLOAD
GENERATE TRAINING
GENERATE CNN MODEL
UPLOAD PLANT IMAGE
IDENTIFY MEDICINAL PLANT
LOGOUT
Predicted image result
Plant Identified As: Pomegranate
Plant Identified As: Pomegranate
PLANT USAGES: This can help prevent or treat various disease risk factors including high blood pressure, high cholesterol, oxidative stress, hyperglycemia, and inflammatory activities.
© 2023 All Rights Reserved By Identification of Medicinal Plant using CNN And its Usage

## 12.CONCLUSION

This research has successfully implemented the convolutional network method to Extract features on medicinal plant and identify them into 40 classes of medicinal plant based on the closest value between the training data and test data.

# 13 FUTURE SCOPE

The future scope for using CNNs in identifying medicinal plants is promising and can lead to several advancements and applications:

Improved Accuracy: As CNN architectures continue to evolve and researchers discover better techniques for training and fine-tuning models, the accuracy of medicinal plant identification systems is expected to increase. This will enhance the reliability of such systems in real-world scenarios.

Mobile Applications: With advancements in hardware and optimization techniques, it's becoming increasingly feasible to deploy CNN models on mobile devices. This opens up opportunities for developing user-friendly mobile applications that can identify medicinal plants in the field using smartphone cameras.

Integration with IoT Devices: Integration of CNN-based plant identification systems with Internet of Things (IoT) devices can enable real-time monitoring of plant populations in natural habitats or cultivation environments. This could be valuable for conservation efforts, agriculture, and biodiversity studies.

Personalized Medicine: By combining plant identification data with information about medicinal properties and traditional uses, personalized medicine approaches can be developed. These approaches could recommend specific plant-based remedies tailored to individual health conditions and preferences.

Crowdsourcing and Citizen Science: Engaging citizen scientists and enthusiasts in data collection and validation efforts can help expand the dataset for training CNN models and improve the accuracy and coverage of plant identification systems.

# 14.REFERENCES

[1] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and Joao V. B. Soares. Leafsnap: A ˜ Computer Vision System for Automatic Plant Species Identification. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, Computer Vision – ECCV 2012, volume 7573, pages 502–516. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[2] Alexis Joly, Herve Go ´ eau, Pierre Bonnet, Vera Baki ¨ c, Julien Barbe, ´ Souheil Selmi, Itheri Yahiaoui, Jennifer Carre, Elise Mouysset, Jean- ´ Franc¸ois Molino, Nozha Boujemaa, and Daniel Barthel´ emy. Interactive ´ plant identification based on social image data. Ecological Informatics, 23:22–34, September 2014.

[3] O. Soderkvist, Computer Vision Classification of Leaves from Swedish Trees, Dissertation, 2001.

[4] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. In 2007 IEEE International Symposium on Signal Processing and Information Technology, pages 11–16, Giza, Egypt, December 2007. IEEE.

[5] Trung Nguyen Quoc and Vinh Truong Hoang. Vnplant-200–a public and large-scale of vietnamese medicinal plant images dataset. In International Conference on Integrated Science, pages 406–411. Springer, 2020.

[6] Tuan Le-Viet and Vinh Truong Hoang. Local binary pattern based on image gradient for bark image classification. In Kezhi Mao and Xudong Jiang, editors, Tenth International Conference on Signal Processing Systems, page 39, Singapore, Singapore, April 2019. SPIE.

[7] Shervan Fekri-Ershad. Bark texture classification using improved local ternary patterns and multilayer neural network. Expert Systems with Applications, 158:113509, November 2020.

[8] Shanwen Zhang, Chuanlei Zhang, Zhen Wang, and Weiwei Kong. Combining sparse representation and singular value decomposition for plant recognition. Applied Soft Computing, 67:164–171, June 2018.

[9] Shanwen Zhang, Chuanlei Zhang, and Wenzhun Huang. Integrating leaf and flower by local discriminant CCA for plant species recognition. Computers and Electronics in Agriculture, 155:150–156, December 2018.

[10] Fateme Mostajer Kheirkhah and Habibollah Asghari. Plant leaf classification using GIST texture features. IET Computer Vision, 13(4):369– 375, June 2019.

turnitin

# Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

| | |
|---|---|
| Submission author: | SSTECH RESEARCH ACADEMY |
| Assignment title: | Assignment Box 11 (Similarities Without References) |
| Submission title: | identification_medicinal_plant_using_CNN.docx.pdf-1.docx |
| File name: | identification_medicinal_plant_using_CNN.docx.pdf-1.docx |
| File size: | 5.31M |
| Page count: | 85 |
| Word count: | 12,423 |
| Character count: | 70,707 |
| Submission date: | 16-Apr-2024 05:18AM (UTC-0400) |
| Submission ID: | 2315971045 |

IDENTIFICATION OF MEDICINAL PLANT USING CONVOLUTIONAL NEURAL NETWORKS

MASTER OF COMPUTER APPLICATIONS

identification_medicinal_plant_using_CNN.docx.pdf-1.docx

ORIGINALITY REPORT

| 17% | 9% | 4% | 12% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Gitam University<br>Student Paper | 3% |
| 2 | www.researchgate.net<br>Internet Source | 3% |
| 3 | Submitted to kitsw<br>Student Paper | 1% |
| 4 | Submitted to Sreenidhi International School<br>Student Paper | 1% |
| 5 | Submitted to University of North Texas<br>Student Paper | 1% |
| 6 | Submitted to Visvesvaraya Technological University, Belagavi<br>Student Paper | 1% |
| 7 | Submitted to University of Teesside<br>Student Paper | <1% |
| 8 | www.jetir.org<br>Internet Source | <1% |
| 9 | Submitted to Visvesvaraya Technological University | <1% |

| | | |
|---|---|---|
| | Student Paper | |
| 10 | Shanwen Zhang, Chuanlei Zhang, Wenzhun Huang. "Integrating leaf and flower by local discriminant CCA for plant species recognition", Computers and Electronics in Agriculture, 2018<br>Publication | <1 % |
| 11 | Submitted to Rivier University<br>Student Paper | <1 % |
| 12 | Submitted to European University of Lefke<br>Student Paper | <1 % |
| 13 | Submitted to Leeds Metropolitan University<br>Student Paper | <1 % |
| 14 | medium.com<br>Internet Source | <1 % |
| 15 | www.python-course.eu<br>Internet Source | <1 % |
| 16 | Submitted to Kaplan Professional<br>Student Paper | <1 % |
| 17 | Submitted to University of Southern Mississippi<br>Student Paper | <1 % |
| 18 | Submitted to University of Hertfordshire<br>Student Paper | <1 % |

| | | |
|---|---|---|
| 19 | D. Ayyappa Reddy, V Esther Jyothi, J. Kasi Viswanath, N Sampreeth Chowdary, D Swapna, S Sindhura. "A Pattern Recognition Model: Hand Gestures Recognition using Convolutional Neural Networks", 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 2023<br>Publication | <1 % |
| 20 | Submitted to Texas A & M University, Kingville<br>Student Paper | <1 % |
| 21 | Submitted to University of Southampton<br>Student Paper | <1 % |
| 22 | Submitted to Presidency University<br>Student Paper | <1 % |
| 23 | Shakthi Sri S, Sathya S, Pandiarajan T. "An Improved Music Recommendation System for Facial Recognition and Mood Detection", ITM Web of Conferences, 2023<br>Publication | <1 % |
| 24 | Submitted to Chicago State University<br>Student Paper | <1 % |
| 25 | Submitted to Northern Arizona University<br>Student Paper | <1 % |
| 26 | ijiemr.org<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 27 | Internet Source | <1 % |
| 28 | Submitted to Somaiya Vidyavihar<br>Student Paper | <1 % |
| 29 | Yuanita A. Putri, Esmeralda C. Djamal, Ridwan Ilyas. "Identification of Medicinal Plant Leaves Using Convolutional Neural Network", Journal of Physics: Conference Series, 2021<br>Publication | <1 % |
| 30 | scholarspace.manoa.hawaii.edu<br>Internet Source | <1 % |
| 31 | Submitted to SRM University<br>Student Paper | <1 % |
| 32 | vdocuments.net<br>Internet Source | <1 % |
| 33 | Submitted to Jawaharlal Nehru Technological University Kakinada<br>Student Paper | <1 % |
| 34 | fastercapital.com<br>Internet Source | <1 % |
| 35 | www.tutorialspoint.com<br>Internet Source | <1 % |
| 36 | www.venturelessons.com<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 37 | Sadjadi, Firooz A., Abhijit Mahalanobis, Henry Xue, and Izidor Gertner. "Automatic recognition of emotions from facial expressions", Automatic Target Recognition XXIV, 2014.<br>Publication | <1 % |
| 38 | Submitted to University of Birmingham<br>Student Paper | <1 % |
| 39 | Submitted to University of College Cork<br>Student Paper | <1 % |
| 40 | "Model–view–controller(MVC)", Salem Press Encyclopedia of Science, 2017<br>Publication | <1 % |
| 41 | Submitted to Higher Education Commission Pakistan<br>Student Paper | <1 % |
| 42 | Submitted to SASTRA University<br>Student Paper | <1 % |
| 43 | Submitted to Vardhaman College of Engineering, Hyderabad<br>Student Paper | <1 % |
| 44 | article.sapub.org<br>Internet Source | <1 % |
| 45 | pdfs.semanticscholar.org<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 46 | Huijie Ma, Shunming Li, Xianglian Li, Jiantao Lu. "Weak Signal Detection Based on Combination of Sparse Representation and Singular Value Decomposition", Applied Sciences, 2022<br>Publication | <1 % |
| 47 | Mihir Nikam, Ameya Ranade, Rushil Patel, Prachi Dalvi, Aarti Karande. "Explainable Approach for Species Identification using LIME", 2022 IEEE Bombay Section Signature Conference (IBSSC), 2022<br>Publication | <1 % |
| 48 | mrcet.com<br>Internet Source | <1 % |
| 49 | www.utdallas.edu<br>Internet Source | <1 % |
| 50 | journals.pen2print.org<br>Internet Source | <1 % |
| 51 | omega.sp.susu.ru<br>Internet Source | <1 % |
| 52 | www.globenewswire.com<br>Internet Source | <1 % |
| 53 | www.ijcaonline.org<br>Internet Source | <1 % |
| 54 | www.ijraset.com<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 55 | Submitted to Kennesaw State University<br>Student Paper | <1 % |
| 56 | Submitted to Middle East Technical University<br>Student Paper | <1 % |
| 57 | aidkca.com<br>Internet Source | <1 % |
| 58 | dev.to<br>Internet Source | <1 % |
| 59 | dokumen.pub<br>Internet Source | <1 % |
| 60 | dspace.susu.ru<br>Internet Source | <1 % |
| 61 | tdk.bme.hu<br>Internet Source | <1 % |
| 62 | theses.hal.science<br>Internet Source | <1 % |
| 63 | www.techscience.com<br>Internet Source | <1 % |
| 64 | "Integrated Science in Digital Age 2020", Springer Science and Business Media LLC, 2021<br>Publication | <1 % |
| 65 | tede2.pucrs.br<br>Internet Source | <1 % |
| 66 | quocent.com<br>Internet Source | <1 % |

# Medicinal Plant Identification Using Convolutional Neural Networks

**M.Anitha[1],L.Sravani[2]**
**#1 Assistant Professor & Head of Department of MCA, SRK Institute of Technology, Vijayawada.**

**#2 Student in the Department of MCA, SRK Institute of Technology, Vijayawada**

**ABSTRACT_** Nutrients that are good for health may be found in herbs and medicinal plants. Unlike the organic, live human organism, chemically manufactured medications have no biological basis. Consequently, chemical drugs are not only inappropriate for human ingestion, but their long-term usage can be detrimental to health. While certain disorders may have more transient or symptomatic treatments, others need patients to undergo persistent pharmaceutical medications. Therefore, there should be a mechanism in place to assist the community in identifying medicinal plants, especially those whose main goal is the introduction of medical leaves. Researchers used a Convolutional Neural Network technique to determine which medicinal plant leaves were included in the research. The goal of this project is to find a mechanism to identify medicinal plant leaves using convolutional neural networks (CNNs). The method of using data learned on computers to categorize medicinal plant leaves according to their advantages; this data is then integrated into mobile applications.

## 1. INTRODUCTION

Herbs and medicinal plants contain chemicals that have beneficial impacts on health. Some parts of the medicinal plant may have the potential to cure, relieve, or at least reduce the severity of a specific health issue. Of Indonesia's 30,000 plant species, 7,000 are used to make herbal medicines. Not only are medicinal plants more accessible than pharmaceutical therapy, but their all-natural components also make them safer. Utilizing phytochemical screening techniques to determine the composition of medicinal plants allows one to discover the active components found in certain plants that could have medical benefit. Unlike the organic, live human organism, chemically manufactured medications have no biological basis. That being said, chemical drugs aren't good for people to ingest, and using them for an extended period of time may be harmful. While certain disorders may have more transient or symptomatic treatments, others need patients to undergo persistent

pharmaceutical medications. Despite the abundance of therapeutic plants, the public's understanding of their benefits lags behind their availability, thus many people turn to artificial medications instead. Therefore, there should be a mechanism in place to assist the community in identifying medicinal plants, especially those whose main goal is the introduction of medical leaves. Neural networks are only one of many approaches to picture leaf identification that take into account size, shape, texture, and color[8,9, 10].

Several methods, including Support Vector Machines, Multilayer Perceptron, Local Binary Patterns, ANN Gray Level Co-occurrence Matrix, K-Nearest Neighbor Algorithms, and others, can now identify medicinal plants given photos of their leaves. Keep in mind that hypertension medication often makes use of medicinal plant leaves, including those of bay, avocado, celery, soursop, acacia, starfruit, grass jelly, and betel. Since hypertension is a major public health issue, researchers looked into the potential medicinal uses of these leaves.

Researchers used a Convolutional Neural Network technique to determine which medicinal plant leaves were included in the research. Convolutional Neural Networks (CNN)[16] are one method from the Machine Learning subfield that deal with two-dimensional data, such as images, by using Deep Learning, a variation of the Multilayer Perceptron. Neural Networks (ANNs) are a cornerstone method in this area. The use of convolutional neural networks (CNNs) trained using backpropagation type input allows for the identification and recognition of objects in pictures [17]. Network propagation in CNN and MLP is two-dimensional, however CNN's weight and linear operational parameters are different. The goal of this study is to use Convolutional Neural Networks to create a system that can differentiate between nine different kinds of hypertensive medicinal plant leaves. The recognized technology will be integrated into an Android-based smartphone app that will make it easy for users to identify various medicinal plant leaves and their advantages.

## 2. LITERATURE SURVEY

**[1] TITLE: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network**

**AUTHOR:** Forrest Sheng Bao; Eric You Xu; Yu-Xuan Wang; Yi-Fan Chang; Qiao-Liang Xiang

In this paper, we employ probabilistic neural network (PNN) with image and data processing techniques to implement a general purpose automated leaf recognition

for plant classification. 12 leaf features are extracted and orthogonalized into 5 principal variables which consist the input vector of the PNN. The PNN is trained by 1800 leaves to classify 32 kinds of plants with an accuracy greater than 90%. Compared with other approaches, our algorithm is an accurate e artificial intelligence approach which is fast in execution and easy in implementation.

**[2] TITLE**: **Vnplant-200–a public and large-scale of vietnamese medicinal plant images dataset**

**AUTHOR**: Trung Nguyen Quoc and Vinh Truong Hoang.

Plant identification is an essential topic in computer vision with various applications such as agronomy, preservation, environmental impact, discovery of natural and pharmaceutical Product. However, the standard and available dataset for medicinal plants have not been widely published for research community. This work contributes the first large, public and multi class dataset of medicinal plant images. Our dataset consists of total 20,000 images of 200 different labeled Vietnamese medicinal plant (VNPlant-200). We provide this dataset into two versions of size 256 × 256 and 512 × 512 pixels. The training set consists of 12,000 images and the remainder are used for testing set. We apply the Speed-Up Robust Features (SURF) and Scale Invariant Feature Transform (SIFT) for extracting features and the Random Forest (FR) classifier is associated to recognize plant. The experimental results on the VNPlant-200 have been shown the interesting challenge task for pattern recognition.

**[3] TITLE**: **Local binary pattern based on image gradient for bark image classification**

**AUTHOR**: Tuan Le-Viet and Vinh Truong Hoang.

In this work, we present a discriminative and effective local texture descriptor for bark image classification. The proposed descriptor is based on three factors, namely, pixel, magnitude and direction value. Unlike most other descriptors based on original local binary pattern, the proposed descriptor is conducted the changing of local texture of bark image. The performance of the proposed descriptor is evaluated on three benchmark datasets. The experimental results show that our approach is highly effective.

**[4] TITLE**: **Bark texture classification using improved local ternary patterns and multilayer neural network**

**AUTHOR:** Shervan Fekri-Ershad

Tree identification is one of the areas that are regarded by researchers. It is done by human expert with high cost. Experts believe that tree bark has a high relation with species in comparison with other phenotype properties. Repeated textures in the bark is usually various with slight differences. So, lbp-like descriptors used in most recent works. But, most of them do not provide discriminative features. Also some texture descriptors are sensitive to noise and rotation. Local ternary pattern is one of the operators that are resistant to the noise with high discrimination. In most of descriptors, histogram of patterns is used to extract features. But, it is rotation sensitive with high computational complexity. In this paper, the main contribution is to propose a method for bark texture classification with high accuracy based on the improved local ternary patterns (ILTP). In the proposed ILTP, the ternary patterns are coded into two binary patterns, and then each one is classified into two uniform/non-uniform groups. The extracted patterns are labeled according to the degree of uniformity. Finally the occurrence probability of the labels is extracted as features. Also, a multilayer perceptron is designed with four theories in the number of hidden nodes. Experimental results on two benchmark datasets showed that our proposed approach provides higher classification accuracy than most well known methods. Noise-resistant and rotation invariant are other advantages of the presented method. The proposed bark texture classification, because of its high classification accuracy, can be applied in real applications and reduce the financial costs and human risks in the diagnosis of plant species

## 3. PROPOSED SYSTEM

The suggested method uses Convolutional Neural Network technology to identify medicinal plants and/or leaves. CNN is a machine learning technique that makes use of Artificial Neural Networks (ANN) together with a Deep Learning extension. Deep learning extends the Multilayer Perceptron (MPL), which is used to interpret two-dimensional input, such photos. CNN uses learnt models to find and identify objects in picture data

## 3.1 IMPLEMENTATION

**UPLOAD DATASET**

Using this module we can load medicinal plant dataset from the location of the project to

Train the CNN algorithm

**GENERATE TRAINING AND TESTING IMAGES**

**ImageDataGenerator**: that rescales the image, applies shear in some range, zooms the image and does horizontal flipping

with the image. This ImageDataGenerator includes all possible orientation of the image.

**train_datagen.flow_from_directory** is the function that is used to prepare data from the train_dataset directory Target_size specifies the target size of the image.

**test_datagen.flow_from_directory** is used to prepare test data for the model and all is similar as above.

**fit_generator** is used to fit the data into the model made above, other factors used are steps_per_epochs tells us about the number of times the model will execute for the training data.

**epochs** tells us the number of times model will be trained in forward and backward pass.

## 4.RESULTS AND DISCUSSION

**Generating images as training and testing**

**1. GENERATE CNN MODEL**

In this module we are generating CNN Model with train_datagen and test_datagen generated by ImageDataGenerator class.

Here we have training this CNN algorithm multiple time to get the better accuracy using epochs.

Finally we will get the best CNN model with average accuracy above 90%

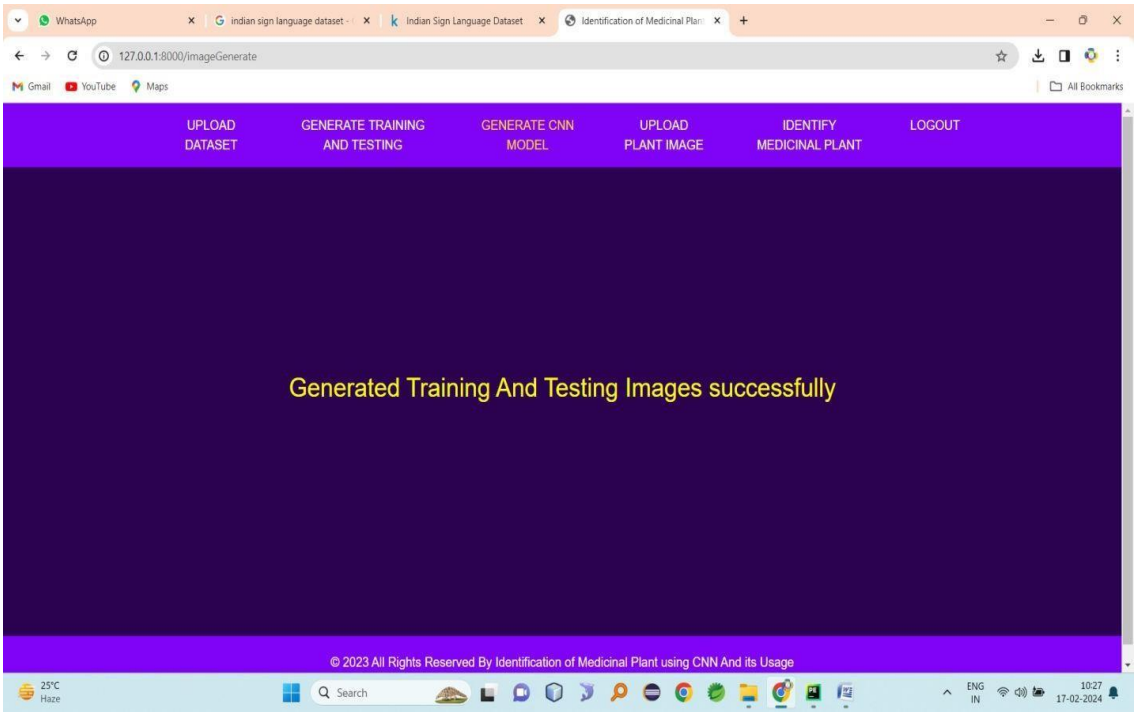
**2. UPLOAD TEST IMAGE**

Using this module we can upload test image AND pass the test image to the model to identify medicinal plant

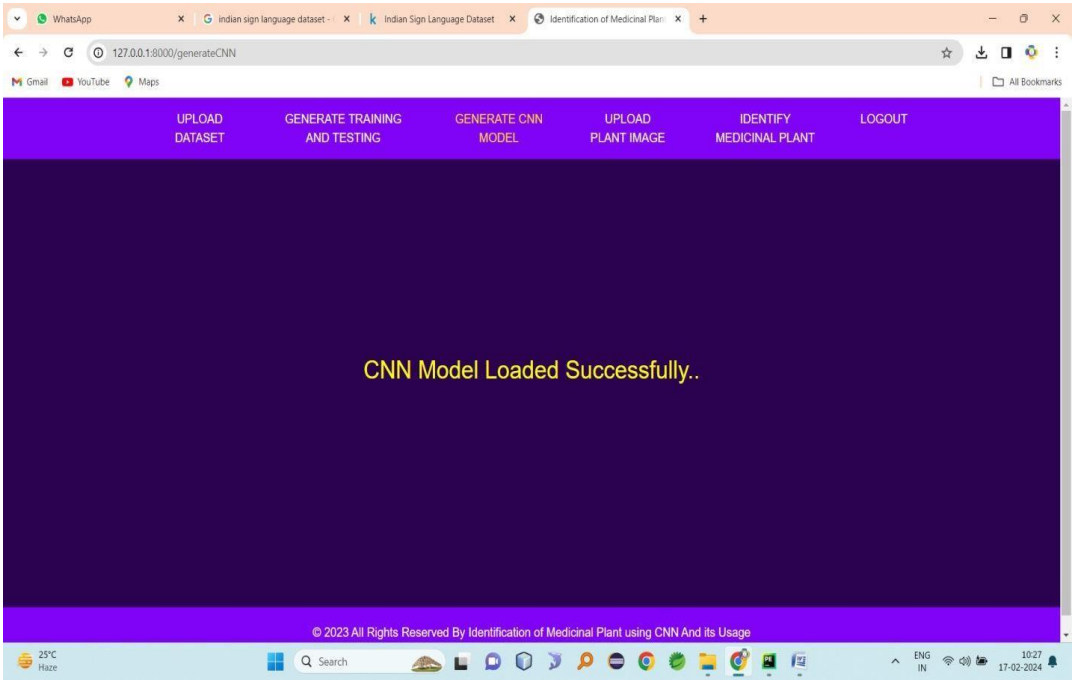
**3. IDENTIFY MEDICINAL PLANT**

Using this model will call the CNN Model which is already generated and take the image

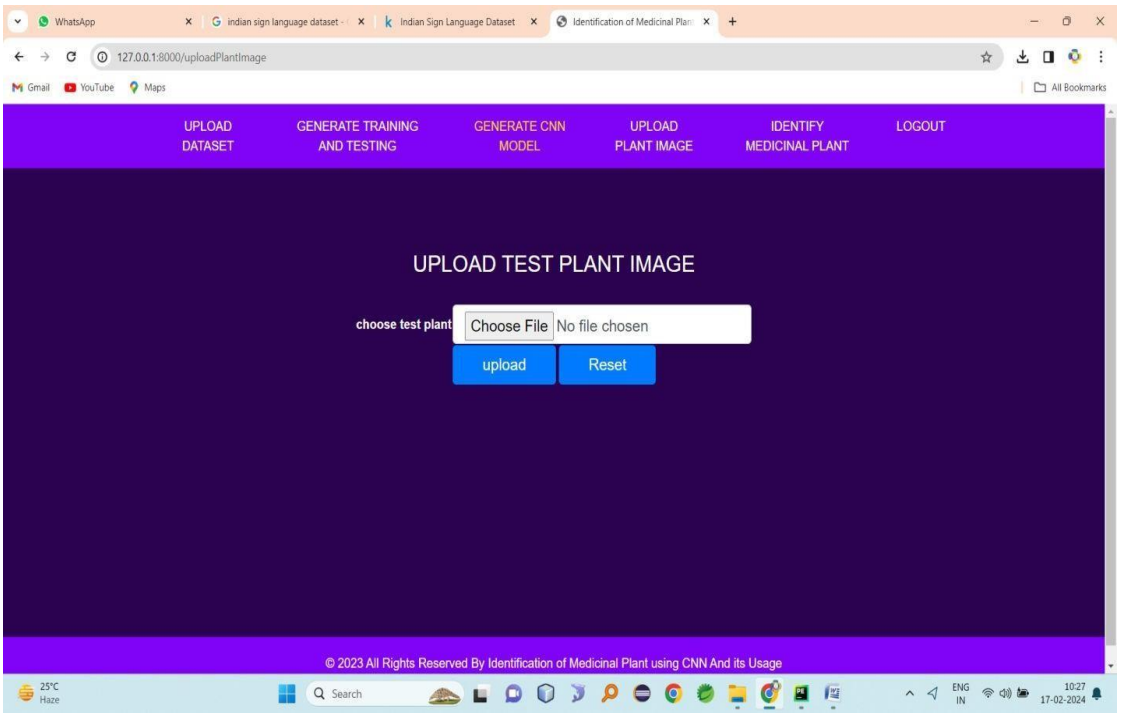
from the $4^{th}$ step and pass to model.Then the model will identify the medicinal plant.



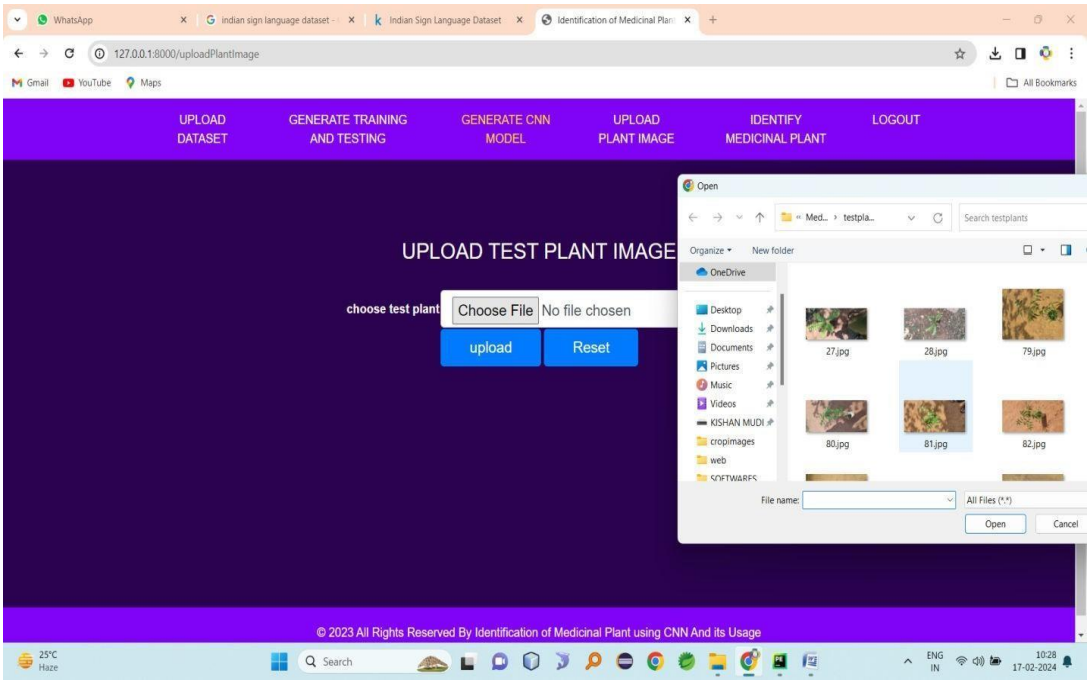

**CNN model generate successfully with training and testing images**



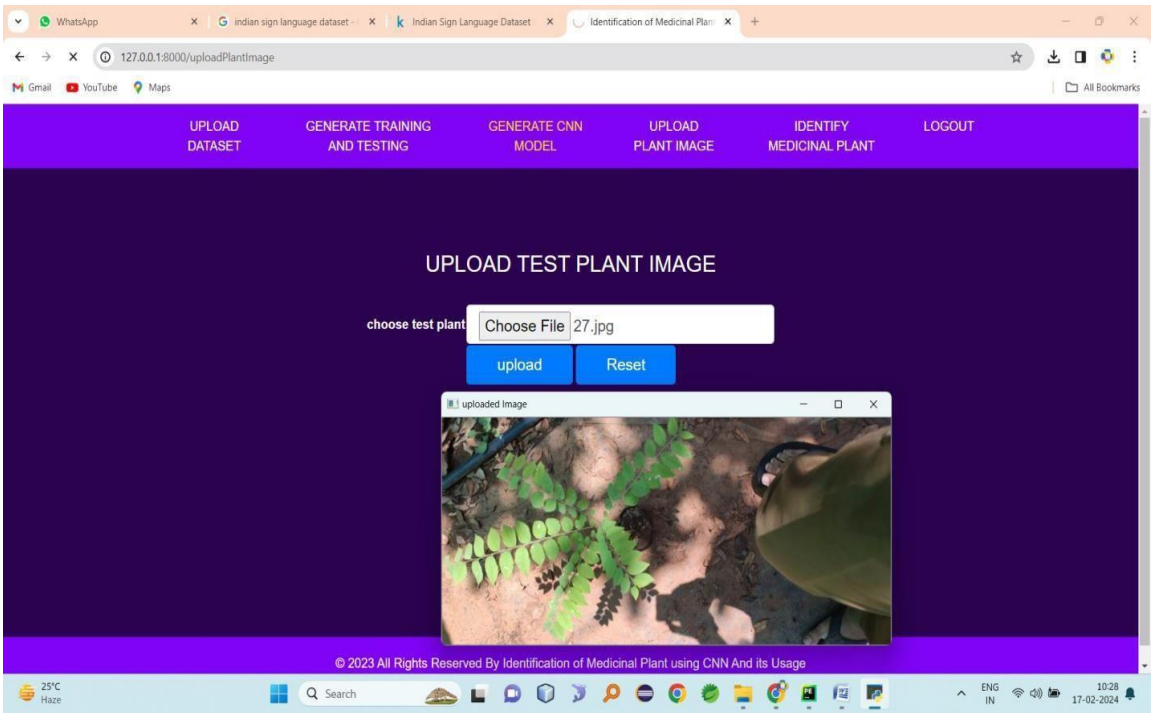
**Choose test image and upload**




**Uploading test plant**



**Uploaed test plant**




**Upload status**

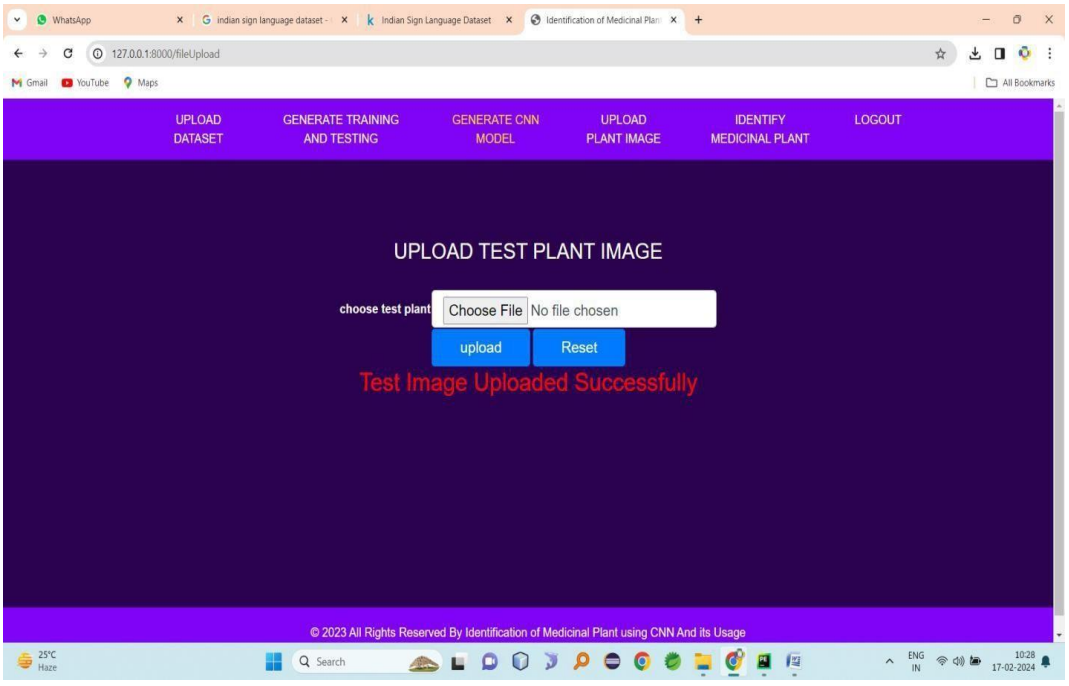
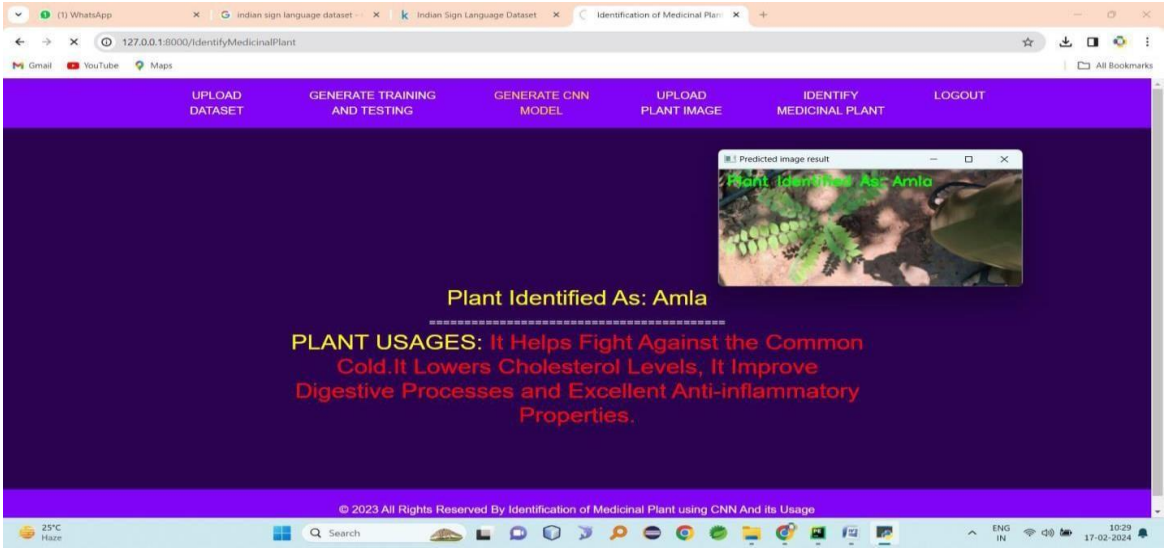


**Image identified**




## 5. CONCLUSION

This research has successfully implemented the Convolutional Neural Network method to extract features on medicinal plant and identify them into 40 classes of medicinal plant based on the closest value between the training data and test data.

## REFERENCES

[1] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and Joao V. B. Soares. Leafsnap: A ˜ Computer Vision System for Automatic Plant Species Identification. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, Computer Vision – ECCV 2012, volume 7573, pages 502–516. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[2] Alexis Joly, Herve Go ´ eau, Pierre Bonnet, Vera Baki ¨ c, Julien Barbe, ´ Souheil Selmi, Itheri Yahiaoui, Jennifer Carre, Elise Mouysset, Jean- ´ Franc¸ois Molino, Nozha Boujemaa, and Daniel Barthel´ emy. Interactive ´ plant identification based on social image data. Ecological Informatics, 23:22–34, September 2014.

[3] O. Soderkvist, Computer Vision Classification of Leaves from Swedish Trees, Dissertation, 2001.

[4] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. In 2007 IEEE International Symposium on Signal Processing and Information Technology, pages 11–16, Giza, Egypt, December 2007. IEEE.

[5] Trung Nguyen Quoc and Vinh Truong Hoang. Vnplant-200–a public and large-scale of vietnamese medicinal plant images dataset. In International Conference on Integrated Science, pages 406–411. Springer, 2020.

[6] Tuan Le-Viet and Vinh Truong Hoang. Local binary pattern based on image gradient for bark image classification. In Kezhi Mao and Xudong Jiang, editors, Tenth International Conference on Signal Processing Systems, page 39, Singapore, Singapore, April 2019. SPIE.

[7] Shervan Fekri-Ershad. Bark texture classification using improved local ternary patterns and multilayer neural network. Expert Systems with Applications, 158:113509, November 2020.

[8] Shanwen Zhang, Chuanlei Zhang, Zhen Wang, and Weiwei Kong. Combining sparse representation and singular value decomposition for plant recognition. Applied Soft Computing, 67:164–171, June 2018.

## AUTHOR'S PROFILE

**Ms.M.Anitha** Working as Assistant Professor & Head of Department of MCA ,in SRK Institute of technology in Vijayawada. She done with B .tech, MCA ,M. Tech in Computer Science .She has 14 years of Teaching experience in SRK Institute of technology, Enikepadu, Vijayawada, NTR District. Her area of interest includes Machine Learning with Python and DBMS.

**Ms.L.Sravani** is an MCA Student in the Department of Computer Application at SRK Institute Of Technology, Enikepadu, Vijayawada, NTR District. She has Completed Degree in B.Sc(compuputer Science) from Triveni Mahila Degree College, Patamata Centre, MG Road, Vijayawada, NTR District. Her area of interest are Machine Learning with Python and java