IDENTIFICATION OF PENINSULAR HERBAL PLANTS USING CNN

A DESIGN PROJECT REPORT

Submitted by

BALA VAISHNAVI K

FAHMITHA NASRIN S

KEERTHANA S

SANDHYA SHALINI S.M

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENGE AND MACHINE LEARNING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM–621112

JUNE-2024

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this design project report titled **“IDENTIFICATION OF PENINSULAR HERBAL PLANTS USING CNN”** is the bonafide work of **BALA VAISHNAVI K (REG.NO : 811721001004) FAHMITHA NASRIN S(REG.NO : 811721001008) KEERTHANA S (REG.NO : 811721001017) SANDHYA SHALINI S.M (REG.NO : 811721001035)** who carried out the project under my supervision. Certified further, that to the best of my knowledgethe work reported here in does not form part of any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	SIGNATURE
Dr.T.AVUDAIAPPAN M.E., Ph.D.,	**Mrs.M.A.REETHA JEYARANI M.E.,**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
Department of AI	Department of AI
K.Ramakrishnan College of	K.Ramakrishnan College of
Technology (Autonomous)	Technology (Autonomous)
Samayapuram-621 112	Samayapuram-621 112

Submitted for the viva-voce examination held on …………………..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"IDENTIFICATION OF PENINSULAR HERBAL PLANTS USING CNN"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

BALA VAISHNAVI K

FAHMITHA NASRIN S

KEERTHANA S

SANDHYA SHALINI S M

PLACE : SAMAYAPURAM

DATE :

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our Institution "**K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY(AUTONOMOUS)",** for providing us with the opportunity to do this project.

We are glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.,**for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr.S.KUPPUSAMY, MBA., Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our principal **Dr.N. VASUDEVAN, M.E., Ph.D.,** who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr.T. AVUDAIAPPAN, M.E., Ph.D.,** HEAD OF THE DEPARTMENT, **ARTIFICAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide **Mrs.M.A.REETHA JEYARANI M.E.,**ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for his incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Mr.P.B.ARAVIND PRASAD B.E.,M.Tech.,** ,other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

Plant species identification is essential for healthy survival as well as the preservation and protection of biodiversity. Manual identification is time-consuming, hence to address this issue deep learning algorithms for automated plant species identification have been developed. A Novel Architecture comprising of EfficientB4Net, Convolutional Block Attention Module (CBAM) and Residual Block Decoder is proposed to act as Autoencoder for identification and retrieval of twenty distinct groups of medicinal plants, widely available in southern India. The EfficientB4 encoder compresses and encodes the input features along with channel and spatial features to the Residual Block Decoder for efficient learning. Residual Block Decoders work to reconstruct the data from the encoded form to be as close to the original input as possible, by eliminating noise. The information-rich encoded features and the global features from the CBAM are transferred to the fully connected layer and stored in the database for retrieval of the plants. When a query image is received, the encoded feature of the query image and the database images are compared using similarity measurement, and the related images are retrieved. From the retrieved images, the query image is identified and the experimental results clearly show that the proposed method has achieved 95% accuracy when compared with other methods.

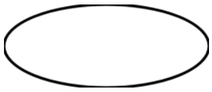
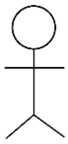
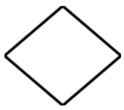
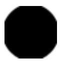
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
CNN	Convolutional Neural Network
CBAM	Convolution Block Attention Module
ML	Machine Learning
DL	Deep Learning
IP	Image Processing
SIMP	South Indian Medicinal plant
GPU	Graphics Processing Unit
RGB	Red Green Blue

LIST OF SYMBOLS

SYMBOL	SYMBOL NAME
	Use Case
	Actor
	Decision
	Start
	Stop

CHAPTER 1

INTRODUCTION

1.1 Deep Learning

Deep learning is a cutting-edge field of artificial intelligence that emulates the intricate workings of the human brain to enable machines to autonomously learn and make complex decisions from vast amounts of data .

It is a type of artificial neural network that has multiple layers of neurons. Each layer learns to extract different features from the input data, and the output of one layer is the input to the next layer. This process continues until the final layer produces the output of the network .

Deep learning networks can be used for a variety of image processing tasks, including classification, detection, segmentation, restoration, and enhancement. Deep learning networks offer a number of benefits for image processing tasks, including accuracy, robustness, scalability, and versatility.

Additionally, they are robust to noise and other variations in the input data, and can be scaled to process large amounts of data. Finally, deep learning networks can be used for a variety of different image processing tasks, making them a versatile tool for image processing.

1.2 CNN

Convolutional neural networks (CNNs) are a type of deep learning network that is particularly well-suited for image recognition and processing tasks. CNNs are able to learn complex patterns and relationships within images by using a series of convolutional and pooling layers.

- Convolutional layers extract features from the input image. This is done by applying a set of filters to the image. Each filter is designed to extract a different type of feature, such as edges, corners, and textures.

- Pooling layers reduce the dimensionality of the data. This is done by combining the outputs of multiple neurons into a single neuron. Pooling layers help to reduce the computational cost of training the CNN and also help to make the CNN more robust to noise and variations in the input data.

CNNs are typically trained using a variety of deep learning algorithms, such as backpropagation and stochastic gradient descent. These algorithms work by adjusting the weights of the neurons in the CNN to minimize the error between the predicted output of the CNN and the actual output.

1.3 PLANT RECOGNITION

1.3.1 Objective

Image recognition is the vital part in the field of computer vision and pattern recognition .Many plants have been critically endangered due to environmental degradation and fast urbanization. Medicinal plant knowledge is particularly vulnerable to the extinction around the world, owing to greater emphasis on biomedical healthcare, devaluation of the traditional herbal practitioner profession by younger generations, lack of cultural support and a force by government programs to modernize the medical practice. In India, Allopathy and Ayurveda are the two main streams of medical system. Ayurveda is totally based on the usage of medicinal plants to treat diseases and enrich the state of good health. Usage of medicinal plants has no side effects compared to chemical drugs (Karimi et al. 2015). Ayurveda medicines are prepared in the form of oil, decoction, powder and paste of leaves. The knowledge about the recognition of plants and its usage is presently handled only by experts. The younger generation is unaware of the traditional knowledge, however with only few exceptions. To disseminate this knowledge to a common man, source of information and data about the medicinal plants of the specific region are the need of the hour. At the same time, incorrect recognition of plants will give adverse effects on humans. Hence, a vision based approach is needed to make it accessible for non-experts for the digital preservation of traditional knowledge system. As a corollary, a system based on plant image recognition is necessary to identify the medicinal plant. The goal of this study is to create a system that improves image recognition rate while also integrating it via machine learning and deep learning approaches.

1.3.2 Role of Image processing

Classifying plant is a technique in which everyone should be correctly assigned in descending series of groups of related plants. Automatic plant classification systems are essential for a wide range of applications, including environmental protection and plant resource survey, as well as for education.

The main challenge faced by them is the identification of the category to which a particular plant belongs to. Phytomorphological analysis, in the beginning to identify the plants, the Botanists can cut a few leaves and use them as specimens, which should be taken to a botanical laboratory as soon as possible. Afterwards, the botanist analyses the leaf's appearance and general traits, such as its boundaries and entire form. If the initial analysis falls under suspect, the botanist recalls the specimens and begins a detailed examination of their colors, forms, and venation patterns. The plant species recognition procedure is concluded when the botanist can change the uncertainty of the first step into certainty by combining the knowledge from the first step with the information from the second step . Botanists may be unable to ascertain the name and characteristics of the plant species even after completing the third step due to a variety of reasons, including poor laboratory specimen quality and genetic alterations made to the plant. In this scenario, botanist gives the inquirer a list of the most likely plant species that look like the one given to the lab. Furthermore, compared to the machine vision-based alternatives, this process is more expensive and time- consuming, and it necessitates the assistance of a botanist.

This present paper focuses on the automation of plant identification through image processing techniques. Image processing is a non-destructive technique which deals with image acquisition, processing, and inferring and provides consistent results. In Industrial and in medical applications, recognition of plants becomes a challenging task, if there are bunch of plant species are available. The significance of incorporating image processing technique with automation is because of its efficiency, fast inspection speed, simplicity, low cost and non-destructive nature.

1.4 PROBLEM STATEMENT

In the ancient times, the Ayurvedic practioners themselves picked the medicinal plants and prepared the medicines for their patients. But, only a few practitioners are using this method nowadays.

Plants acquired by common people who are not professionally trained inselecting correct medicinal plants may lead to unpredictable side effects. Incorrect recognition of medicinal plants renders Ayurvedic treatment inefficient and results in unpredictable side effects.

In this context, strict quality control methods must be implemented on Ayurvedic pharmaceuticals and raw materials used by the industry in order tocontinue the industry's current growth while preserving the efficacy and credibility of medicines.

As a result, the aim of this study is to preserve and disseminate the knowledgeof south Indian medicinal plants under unconstrained environmental condition.

CHAPTER 2

LITERATURE SURVEY

2.1 TITLE : IDENTIFICATION OF PLANTS USING DEEP LEARNING: A REVIEW

AUTHOR : Rakibul Sk, Ankita Wadhawan

YEAR OF PUBLICATION : 2021

ABSTRACT : Identification of plants is a very important field in the earth's ecology to maintain a healthy atmosphere. Certain of these plants have significant medicinal properties. Nowadays of finding a plant is not easy by looking at its physical properties. This paper provides an academic database of literature between the duration of 2015-2020. It has been observed that the new generation of convolutionary neural networks(CNNs) in the space area of image recognition has produced remarkable performance. In this paper, technique are discussed the concepts of Deep learning and different leaf recognition methods.

MERITS : High accuracy, scalability, robustness, feature extraction**.**

DEMERITS : Data requirements, High complexity, overfitting, Interpretability, environmental variability.

2.2 TITLE : CROP YIELD PREDICTION USING MACHINE LEARNING: A SYSTEMATIC LITERATURE REVIEW

AUTHOR : Thomas Van Klompenburg, Ayalew Kassahun, Cagatay Catal

YEAR OF PUBLICATION : 2020

ABSTRACT : Machine learning is an important decision support tool for crop yield prediction, including supporting decisions on what crops to grow and what to do during the growing season of the crops have been applied to support crop yield prediction Literature Review SLR extract and synthesize the algorithm and features that have been used in crop yield prediction studies. Based on our search criteria we retrieved 567 relevant studies from 6 electronic database, of which we have selected 50 studies for Further analysis using inclusion and exclusion criteria we investigated these selected studies carefully analyzed the methods and features used, and provided suggestions for further research. After this observation we performed an additional search in electronic databases to identify deep learning based studies, reached 30 deep learning based papers, and extracted the applied deep learning algorithms. According to this additional analysis Convolutional Neural Networks(CNN) is the most widely used deep learning algorithm in these studies and the other widely used deep learning algorithms are Long-Short Term Memory(LSTM) and Deep Neural Network(DNN).

MERITS : High accuracy, efficiency, adaptability, handling non-linearity**.**

DEMERITS : Data dependency, complexity, high cost, Non-scalable.

2.3 TITLE : A COMPREHENSIVE LITERATURE SURVEY FOR DEEP LEARNING APPROACHES TO AGRICULTURAL APPLICATIONS

AUTHOR : P.E.Rubini and P.Kavitha

YEAR OF PUBLICATION : 2021

ABSTRACT : Agriculture is the main economic activity in many parts of countries like India, China and Africa. The agriculture division needs tremendous development to endure and to meet the demands of the growing population. But the production level of agriculture is decreasing nowadays because of many issues like uncertain rainfall, lack of prediction in crop diseases, unavailability of labourer and improper price fixation for crops. The lack of knowledge in applying technology in the field offarming is also a major issue in countries like India. To address this challenge in the field of agriculture, smart farming is introduced. Deep learning techniques which is a part of Machine Learning can be applied to bring remarkable challenges in smart farming by using a dataset for increasing the productivity of crops. The key target of this work is to identify the root causes and to provide appropriate techniques to enrich the lives of farmers in agriculture and lead the next generation in a more resourceful manners.

MERITS : Avoiding Redundancy, holistic understanding, supports evidence Based frameworks**.**

DEMERITS : Time consuming, Information overload, quality variability.

2.4 TITLE : DEEP LEARNING FOR MEDICINAL PLANT SPECIES CLASSIFICATION AND RECOGNIZATION : A SYSTEMATIC REVIEW

AUTHOR : Adibaru Kiflie Mulugeta, Durga Prasad Sharma and Abebe Haile Mesfin

YEAR OF PUBLICATION : 2023

ABSTRACT : Knowledge of medicinal plant species is necessary to preserve medicinal plants and safeguard biodiversity. The classification and identification of these plants by botanist experts are complex and time-consuming activities. This systematic review's main objective is to systematically assess the prior research efforts on the applications and usage of deep learning approaches in classifying and recognizing medicinal plant species. Our objective was to pinpoint systematic reviews following the PRISMA guidelines related to the classification and recognition of medicinal plant species through the utilization of deep leaning techniques This review encompassed studies published between January 2018 and December 2022. Initially, we identified 1644 studies through title keyword and abstract screening After applying our eligibility criteria, we selected 31 studies for a thorough and critical review. The main findings of this reviews are the selected studies were carried out in 16 different countries, and India leads in paper contributions with 29%, followed by Indonesia and Sri Lanka (2) A private dataset has been used in 67.7% of the studies subjected to image augmentation and preprocessing techniques.

MERITS : Automation, adaptability, scalability, handling complex data.

DEMERITS : Computational resource, overfitting, lack of interpretability.

2.5 TITLE : A SYSTEMATIC LITERATURE REVIEW OF MACHINE LEARNING TECHNIQUE DEPLOYED IN AGRICULTURE : A CASE STUDY OF BANANA CROP

AUTHOR : Priyanka Sahu, Amit Prakash Singh, Anuradha Chug and Dinesh Singh.

YEAR OF PUBLICATION : 2022

ABSTRACT : Agricultural productivity is the asset on which the world's economy thoroughly relies. This is one of the major causes that disease identification in fruits and plants occupies a salient role in farming space. as having disease disorders in them is obvious. There is a need to carry genuine supervision to avoid crucial consequences in vegetation; otherwise, corresponding vegetation standards, quantity, and productiveness gets affected. At present, a recognition system is required in the food handling industries to uplift the effectiveness of productivity to cope with demand in the community. The study has been carried out to perform a systematic literature review of research papers that deployed machine learning (ML) techniques in agriculture, applicable to the banana plant and fruit production. Thus, it could help upcoming researchers in their endeavors to identify the level and kind of research done so far. The authors investigated the problems related to banana crops such as disease classification, chilling injuries detection, moisture content. etc. Moreover, the authors have also reviewed the deployed frameworks based on ML., sources of data collection, and the comprehensive results achieved for each study. Furthermore, ML. architectures/techniques were evaluated using a range of performance measures.

MERITS : Identifies research gaps, Decision making aid, educational resource.

DEMERITS : Resource intensive, publication bias, interpretation challenges.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Hardware Requirements

- CPU type : Intel i7
- RAM size : 8.00 GB
- GPU type: NVIDIA GTX 180 GPU
- Webcam

Webcam

- **Real-Time Plant Identification**: By utilizing a webcam, you can capture real-time images or videos of plants. This enables users to instantly identify medicinal plants in the field, garden, or any other location, making it highly practical for both experts and non-experts.
- **User-Friendly Interface**: Develop a user-friendly application or website that integrates the webcam feed. Users can point the webcam at a plant of interest, and the deep learning model will process the image and provide an identification result.
- **Computer Vision with Deep Learning**: Use a pre-trained deep learning model, such as a Convolutional Neural Network (CNN), that has been trained on a dataset of medicinal plant images. This model can then be fine-tuned for specific medicinal plant species or taxonomies, if necessary.
- **Real-Time Feedback**: Provide instant feedback to users by overlaying the

identified plant species on the webcam feed. This can include the commonand scientific names of the plant, along with additional information such as its medicinal properties, uses, and potential hazards.

- **Image Preprocessing**: Before passing the webcam images to the deep learning model, apply image preprocessing techniques to enhance the quality of the images. This may involve adjusting brightness, contrast, and sharpness, as well as removing background clutter.

3.2 Software Specification

- Library : Tensor flow,openCV
- IDE : Jupyter Notebook
- GPU Architecture : Google Colab

Python

Python is a highly versatile and widely used programming language known for its readability and simplicity. It is a general-purpose language that can be employed for a wide array of applications, including web development, data analysis, artificial intelligence, and automation. Python's clean and concise syntax makes it easy to read and write, and its interpreted nature allows for quick development and debugging. It boasts a rich standard library, a vibrant community, and is open-source, running on multiple platforms.

Python supports various programming paradigms, such as object-oriented and functional programming, and provides high-level data structures for convenient coding. Its extensive ecosystem of third-party libraries and frameworks, as well as its suitability for rapid prototyping, have contributed to its popularity among beginners and experienced developers alike.

Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform that offers a free and accessible environment for Python programming. It isparticularly popular among data scientists and machine learning practitioners.With Google Colab, you can write, run, and share Python code directly in your web browser, eliminating the need for local Python installations.Notably, Colab provides access to virtual machines equipped with GPUs, which greatly speeds up computationally intensive tasks, such as trainingdeep learning models.

One of its standout features is real-time collaboration, akin to Google Docs, allowing multiple users to work on the same notebook simultaneously, making it a valuable tool for collaborative projects, learning, and research. Furthermore, Google Colab seamlessly integrates with Google Drive, making it easy to store and manage your Colab notebooks in the cloud.

Overall, Google Colab is a powerful and accessible platform for a wide range of Python programming tasks, especially those related to data analysis and machine learning.

CHAPTER 4
SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Generally, plants are identified by their physical properties such as shape, texture, veins, and color. Different feature extraction methods have been used to extract the features and they are classified as Conventional Based Approaches, Machine Learning Based Approaches and Deep Learning Based Approaches.

4.1.1 Conventional Approaches:

- Conventional methods such as Fuzzy Integral Method has been used to identify the plants by their leaves
- Earlier, the color histogram has been modelled as a Gaussian cluster in color space . The color based image retrieval using Spatial Chromatic Histogram has been done to synthesise the information about the location of pixels having the same color and their arrangement within the image.

4.1.2 Machine Learning based Approaches :

- Color based recognition The Ayurvedic medicinal plants are identified by some deterministic parameters using color and texture on both the sides of the leaves
- Shape based recognition The leaf features are identified by using the Gabor filter and Gray Level Co-occurrence Matrix (GLCM) whereas the shape features have been grabbed using Curvelet transform coefficients along with Invariant moments through Neuro-Fuzzy Controller (NFC) and Multi-Layered Perceptron (MLP)

- Texture based recognition Texture is another major field of study in plant identification and it is used to describe the surface of the leaf based on the pixel distribution over a region

4.1.3 Drawbacks of Existing Methodology:

- Plant recognition depends on various perspectives of plant like leaves, flower, venation, stem, fruits etc. Flowers are the most visible part of the plants.
- Plant recognition becomes challenging when two different plants have same colored flowers and plants without flower since, flowers are susceptible to seasonal and biological factors. Along with flower, the combination of leaves and flower will be good choice for recognition in such cases.
- Further, recognising medicinal plants in unconstrained environment is a tedious task since the same plant comprises of leaves in various scales, illumination and while capturing, the plant images will vary as well as it is also affected by the viewpoint of the capturing system.
- Even though some plant recognition methods exist by considering the above mentioned vision problems, Plants with different depth of focus have not yet been dealt, as their effects are more in real-time recognition.

4.2 PROPOSED SYSTEM

The proposed Autoencoder model has been designed with EfficientB4Net to identify the plant from the SIMP database . The model consists of five layers, in which each layer has an EfficientB4block, Convolutional layer, and pooling layer. The EfficientB4Net block in the network trains a kernel by creating a more abstract representation of the data at each layer. The EfficientB4 block along with CBAM, and Residual block decoder has been designed to act as a synthesizer in the input stage with several convolutional layers and pooling layers in it .A feature map has been generated from the convolutional layer by convolving the kernels with the input feature maps. The pooling layer decreases the number of neurons to be processed in the subsequent layers. All the neurons are concatenated and densely connected 91 to the fully connected layer. The portionof the network before the fully connected layers acts as a feature extractor.

4.2.1 Advantages of Proposed Methodology :

1. High Accuracy: Deep learning models, particularly convolutional neural networks (CNNs), have demonstrated the ability to achieve high levels of accuracy in image recognition tasks. When trained on a diverse dataset of medicinal plants, they can reliably identify and classify plant species, which is crucial for medicinal plant recognition.

2. Automation: Deep learning models can automate the identification process, reducing the need for human experts to manually inspect and classify medicinal plants. This can significantly speed up the identification process and reduce the potential for human error.

3. Scalability: The proposed methodology can be easily scaled to handle large datasets of medicinal plant images. The model can be retrained to improve its accuracy and adapt to new species or variations within existing species

CHAPTER 5

ARCHITECTURE DESIGN

5.1 Architecture Diagram

An architecture diagram is a visual representation of the structure of a system .It shows the different components of the system and how they interact with each other. Architecture diagrams can be used to document existing systems, plan new systems, or communicate the design of a system to others.

Architecture diagram is shown in Fig 5.1.1 and Fig 5.1.2

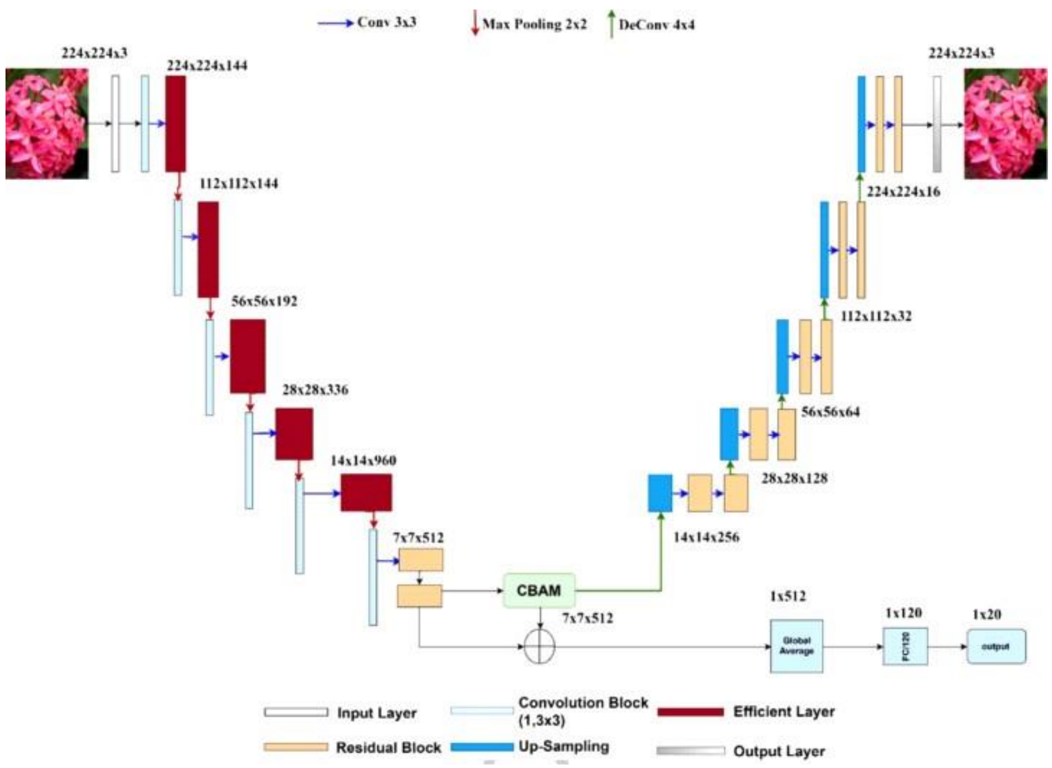


Fig 5.1.1 Training Stage

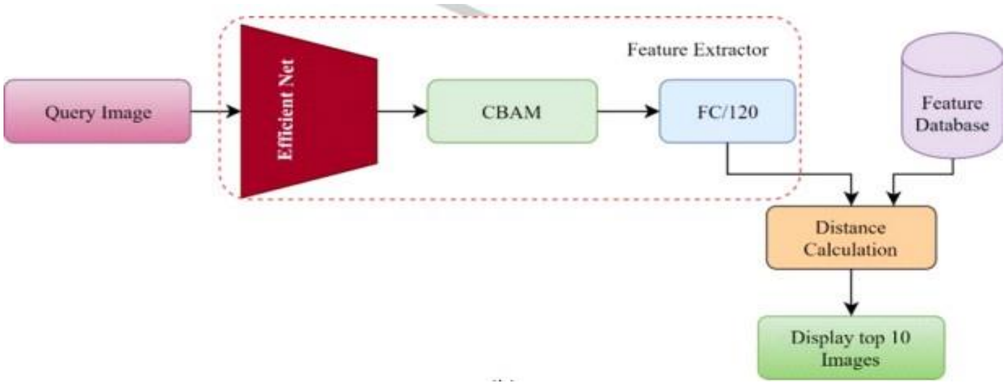


Fig 5.1.2 Testing Stage

5.2 Use-case Diagram

A use case diagram is a type of Unified Modeling Language (UML)diagram that shows the interactions between a system and its users. It is a graphical representation of the functionality of a system from the user's perspective. They are used to document the requirements of a system and to communicate the system's functionality to stakeholders.

Use-case diagram is shown in Fig 5.2

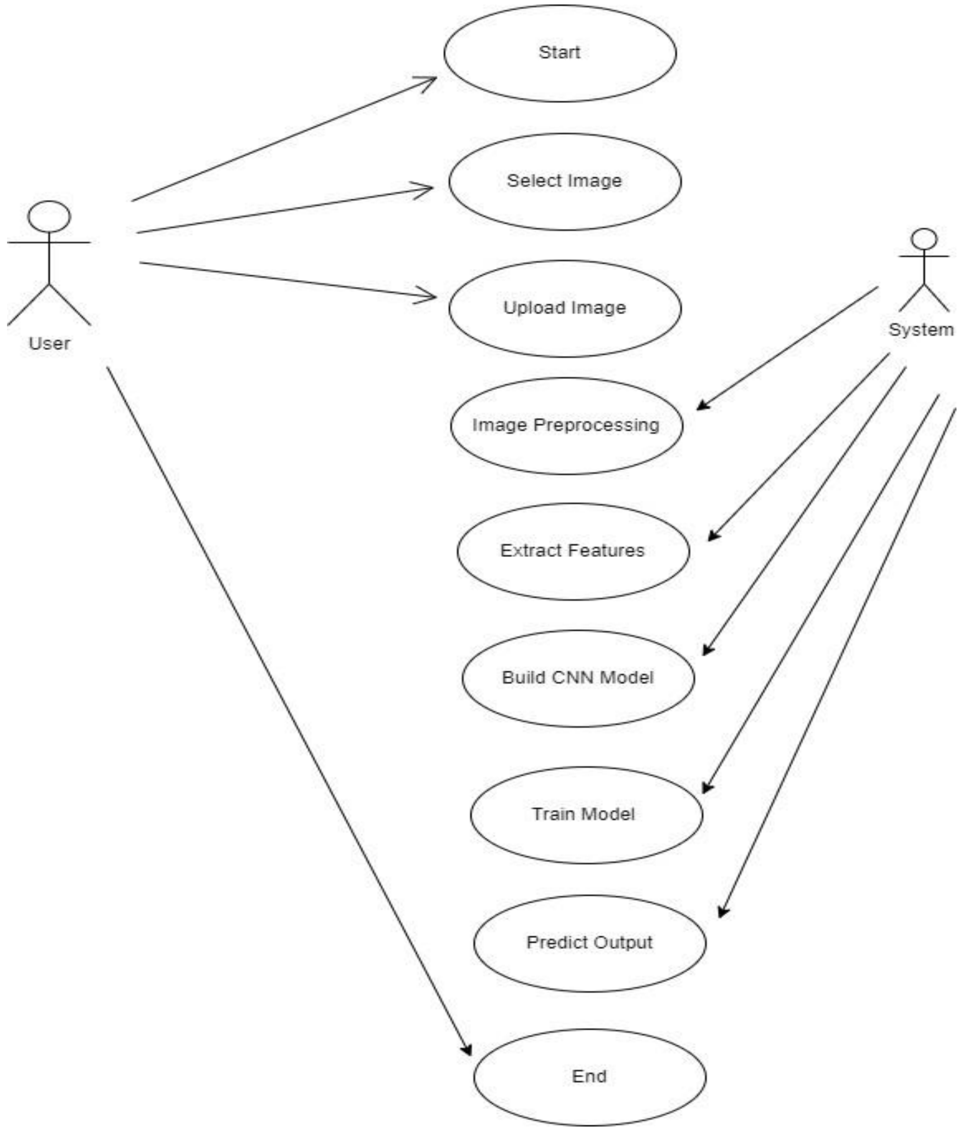


Fig 5.2 Use-case Diagram

5.3 Sequence Diagram

A sequence diagram is a type of Unified Modeling Language (UML)diagram that shows the sequence of interactions between objects in a system. It is a graphical representation of the dynamic behavior of a system.

Sequence diagram is shown in Fig 5.3

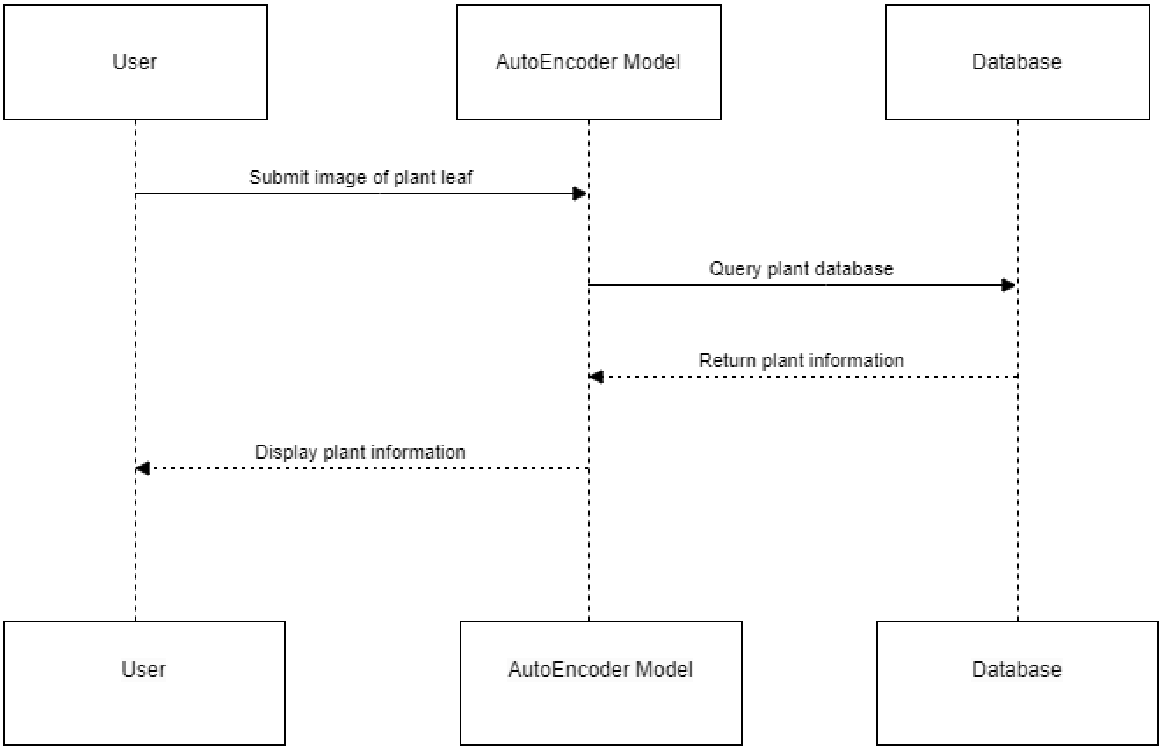


Fig 5.3 Sequence Diagram

5.4Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) Diagram that visualizes the dynamic aspects of a system, particularly the flow of activities or processes within it. Activity diagrams are used to model workflows, business processes, or the behavior of software systems.

Activity diagram is shown in Fig 4.4

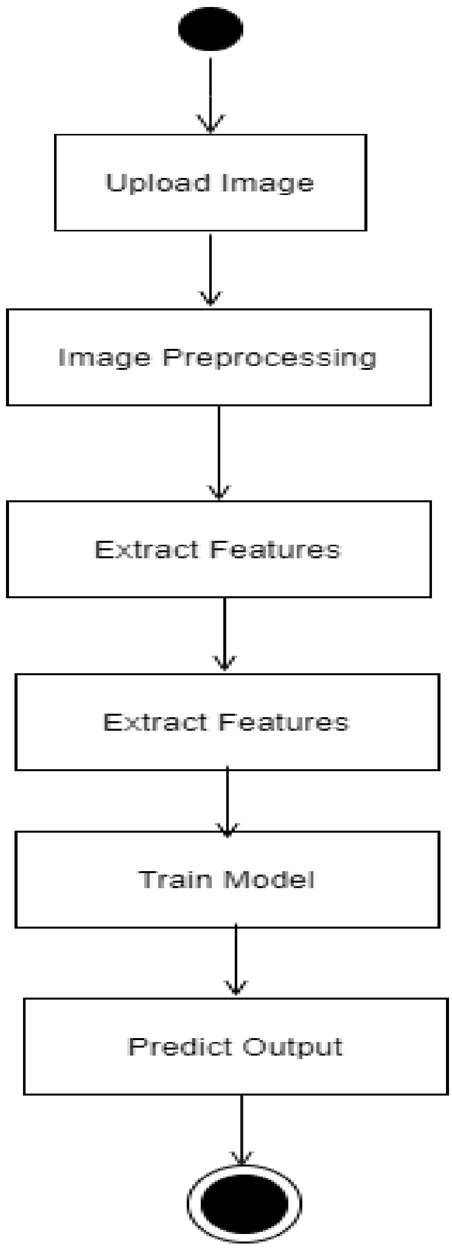


Fig 5.4 Activity Diagram

CHAPTER 6
MODULE DESCRIPTION

6.1 List of Modules

- Dataset Collection
- Data Preprocessing

  1.Training Stage

  2.Testing Stage

- Autoencoder Model
- Data Evaluation

6.2 Module Description

6.2.1 Dataset Collection

Many plants have been critically endangered due to environmental degradation and fast urbanization. Loss of biological resources leads to decline intraditional knowledge about plants and the traditional plant related practices are dying out. Medicinal plant knowledge is particularly vulnerable to the extinction around the world, owing to greater emphasis on biomedical healthcare,devaluation of the traditional herbal practitioner profession by youngergenerations, lack of cultural support and a force by government programs tomodernize the medical practice among these factors. The knowledge of medicinalplants and the experts in the field has reduced so preservation of these medicinal plants should be taken care. The key contributions of the SIMP database generation are

- To identify the medicinal plants in nareasearby .
- To acquire the images in an unconstrained environmental conditions

- To augment the image samples.
- To evaluate the vulnerability of the collected samples using state of the art algorithms.

Fig 6.2.1.1 Sample Images of herbs in SIMP Dataset

Fig 6.2.1.2 Sample Images of shurbs in SIMP Dataset

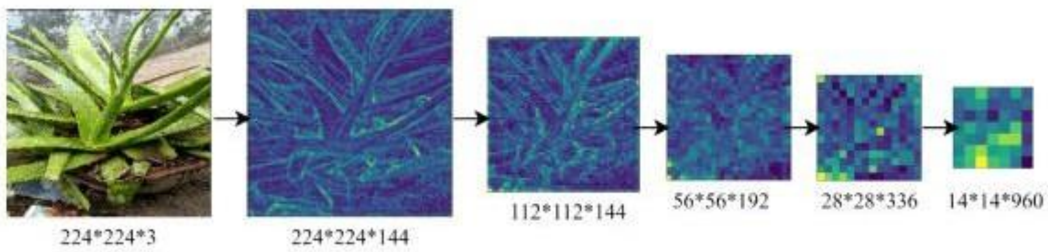
Fig 6.2.1.3 Sample Images of trees in SIMP Dataset

6.2.2 Data Preprocessing

The use of CBAM and Residual Block along with EfficientB4Net sequentially observes the channel and spatial map that produce refined plant features. With these refined features, the proposed model overcomes the degradation problems/issues, and improves the model performance.

The Residual Block Decoders work to reconstruct the data from the encoded form to be as close to the original input as possible, by eliminating noise.The loss function is calculated for every output layer and back propagates to the network. This learning happens until the Residual Block Decoder produces a rich representation of the input features. Once the encoder model is trainedwith the optimum weight and bias values, the Residual Block Decoder will be neglected.

The obtained enriched features along with channel
and spatial features re summed up and given to the fully connected layer to produce a classified output, which is stored in the feature database. These learned features extracted for the trained model used for retrieval.



.

Fig 6.2.2.1 Feature extraction in each stage of the encoding phase

6.2.3 Training Stage

The training stage of the proposed autoencoder model for medicinal plant recognition involves several key steps in preparing the model to accurately classify and recognize plant species based on input images. It begins with a dataset of medicinal plant images, which serves as the foundational training data.

The model is initialized with parameters like the number of classes, weights, biases, and a learning rate, essential for the training process. Training is conducted iteratively, where the model reconstructs input images and extracts features from them, creating abstract representations of the data at different layers.

During this process, a "gain" is calculated, which measures the improvement in the reconstructed data compared to the original input, and if this gain falls below a specified threshold and the "Stop_train_flag" is set to "Y," the training process stops. Extracted features are stored in a feature database for future use. Additionally, hyperparameters like the learning rate and batch size are fine-tuned to optimize the model's performance.

Training is typically performed on specialized hardware, and the model's parameters are updated through backpropagation to minimize the loss function, ensuring that the model converges to accurately represent the features of the input data. Once training is complete, features are extracted for testing, allowing the model to identify and classify medicinal plants in the testing phase. The quality of training data, hyperparameter choices, and optimization techniques all impact the model's effectiveness and accuracy in recognizing plant species.

6.2.4 Testing Stage

Once the network has been trained for identifying and classifying the medicinal plant images, the features are extracted from the fully connected layer for the trained model and they are stored in the feature database for image retrieval.The testing phase model is shown in Figure 5.9. In the first layer, edges or blobs are extracted by fine tuning the weights and the species patterns are seen in the higher levels.

During testing, when a query image I is given as input to theproposed network, the features are extracted as output from the fully connected layer. The cosine similarity metric is calculated for the extracted features and number of images with most similar features are retrieved from the database aswell they vote in favour of their annotated species.

The cost of the minimum distance path for the retrieved images is calculated individually. The query imageis identified belonging to the category of the image with minimum cost.

The cosine similarity metric used to find the similarity between the query image and the database images , is calculated using the Equation (6.4). The database images are ranked according to their similarity.

$$Cosine\ \ similarity = \frac{a.b}{\|a\|\|b\|}$$

$$\cos(\theta) = \frac{\|a\|^2 + \|b\|^2 - \|a-b\|^2}{2\|a\|.\|b\|}$$

$$\|a-b\|^2 = d^2_{euclidian} = \|a\|^2 + \|b\|^2 - 2a.b$$

Combining Equations (5.2) and (5.3)

$$\cos(\theta) = \frac{a.b}{\|a\|\|b\|}$$

6.2.5 Auto Relaxing Alert

Autoencoder model is an unsupervised model that applies the backpropagation and sets the target values to be equal to the inputs. It reduces the size of the inputs into a smaller representation and reconstructs the compressed data into original data based on the application.

The SIMP images have been collected in unconstrained environment and they have a wide variation in capturing images like camera resolution, lighting conditions, and angles.Consequently, an Autoencoder model has been designed to extract only the required features of an image and it generates the output by removing unnecessary interruption and also with reduced pixel value.

The sample image used and the schematic diagram of the proposed model are shown in Figure 6.1. Since, the dataset images consist of 100 classes that have been taken in unconstrained environmental conditions, it is difficult to label each classes. Autoencoder solves the problem of finding new classes for the unlabelled data. The model can be trained to learn how the data can be effectively split into subclasses. It does not have dense layers to learn. It can only use convolutional layers to learn and it is better for an image or video.

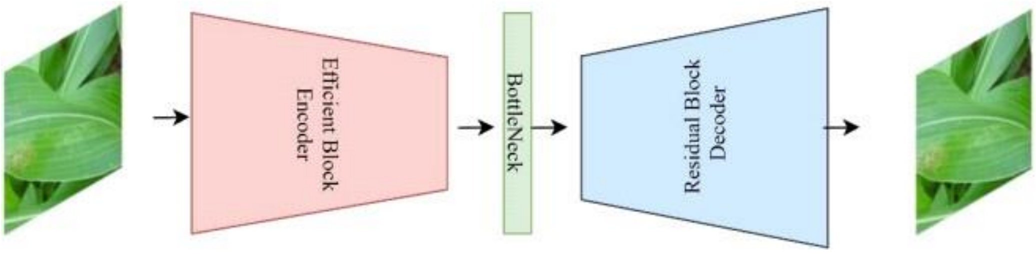


Fig 6.2.5.1 Schematic diagram of auto encoder model

The proposed Autoencoder model has been designed with EfficientB4Net to identify the plant from the SIMP database as shown in The model consists of five layers, in which each layer has an EfficientB4block, Convolutional layer, and pooling layer. The EfficientB4 block along with CBAM, and Residual block decoder has been designed to act as a synthesizer in the input stage with several convolutional layers and pooling layers in it.The EfficientB4Net block in the network trains a kernel by creating a more abstract representation of the data at each layer.

A feature map has been generated from the convolutional layer by convolving the kernels with the input feature maps.

The pooling layer decreases the number of neurons to be processed in the subsequent layers. All the neurons are concatenated and densely connected to the fully connected layer. The portion of the network before the fully connected layers acts as a feature extractor. Training accuracy has been improved by enlarging the width of the network, deepening the depth, and improving the resolution of the input image.

CNN has been specially designed for the processing of images along with its spatial and temporal dependencies and also to minimize the number of parameters. Nowadays, many efforts have been made to improve the CNN's performance in different dimensions. The performance of the CNN is determined by the number of layers and through which, the data move, leading to increase its depth and the number of filters increases its width as well as the increased resolution gives more and finer details (Duong et al. 2020). All the models that have been designed yet are too wide, deep, and with high resolution. These characteristics help the model in the starting stage and quickly saturates in the next.

An EfficientNet has been proposed to overcome these drawbacks byregularly automating the dimensions with specific constraints and never letting them as a bottleneck. The EfficientNet depends on two keynotes such as an improved baseline architecture and an efficient scaling strategy. EfficientNet B0 is the baseline network, that optimizes the accuracy on a predefined classification and Float Point Operations Per Second (FLOPS) in a parallel way (Tan &Le 2019). It outperforms many other networks like DenseNet, InceptionNet, etc.in terms of complexity, speed, and accuracy and so, it has been chosen for plant retrieval application. Due to all the above-mentioned features, there is a minimum limiting factor and the improvement in

The general Architecture of EfficientNet is shown in Figure 6.2.

In general, the ConvNet layer n is defined as a function as shown below

$$P_n = F_n(Q_n) \tag{6.1}$$

where, Fn is the function, Pn is the output tensor and Qn is the input tensor with tensor shape {Hn, Wn} as the spatial dimension and {Cn}as the channel dimension.

A CNN Net (N) has been described as a series of compound layers.

$$N = [F_m \otimes \ldots\ldots F_2 \otimes F_1(x_1)] = \otimes_{n=1\ldots m} F_n^{L_n}(x_1)$$

Normally, the CNN layer is applied in multiple stages and every stage uses the same network architecture. Therefore, it has been defined as

$$N = \otimes_{n=1\ldots m} F_n^{L_n}(Q_{\{H_n, W_n, C_n\}}) \tag{6.2}$$

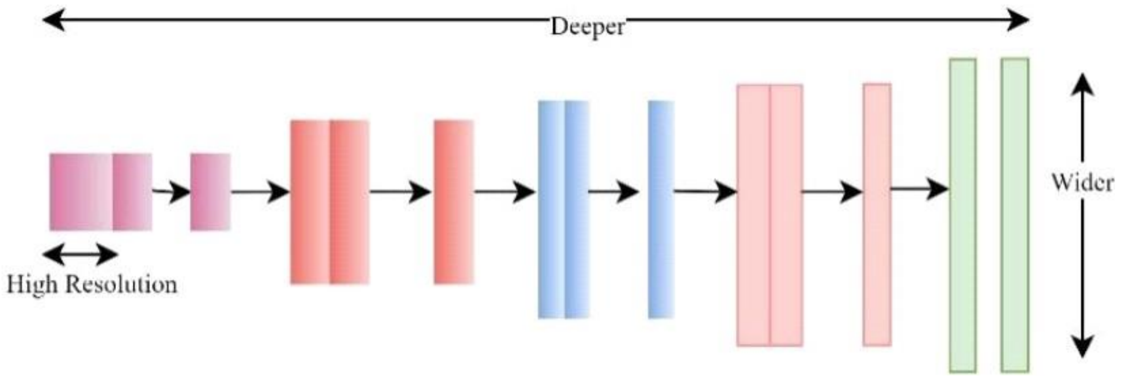


Fig 6.2.5.2 General architecture of EfficientNet

An EfficientNet reduces the design space by forcing all the layers to bescaled uniformly with a constant ratio. For any given resource constraints, the EfficientB4Net encoder maximize the model accuracy and it is formulated as

$$N = \bigotimes_{n=1...m} F_n^{L_n}(Q_{\{H_n, W_n, C_n\}}) \qquad (6.3)$$

where, d,w,r are the width, depth, and resolution coefficients for the scaling network.

Attention mechanism is the main component of the recent CNN architectures. It is used to improve the quality of feature representation of the network,only by focusing on the important features and neglecting the unwanted features in an implicit way. CBAM is a simple effective module for Convolutional Neural Networks (CNN). It sequentially observes the channel and spatial attention maps in both the dimensions and adds them with the input features for adaptive feature refinement as shown in Figure 6.3. The main aim of CBAM is to increase the representation power by focusing on the significant features in both the dimensions.

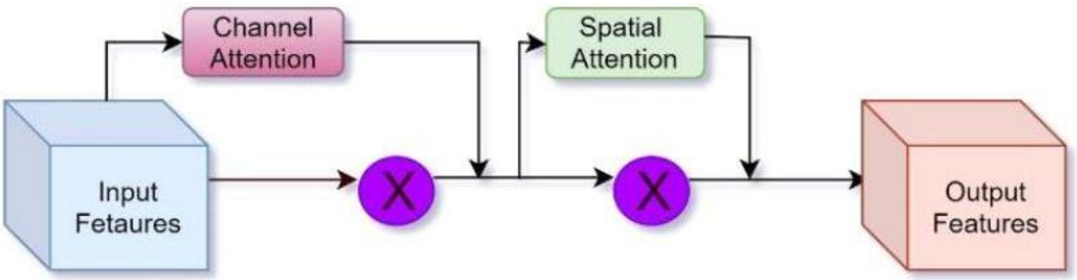


Fig 6.2.5.3 Block diagram of convolutional block attention module

If an intermediate feature map F is given as input, CBAM produces a 1D channel attention map and 2D spatial attention map. The attention process is given as:

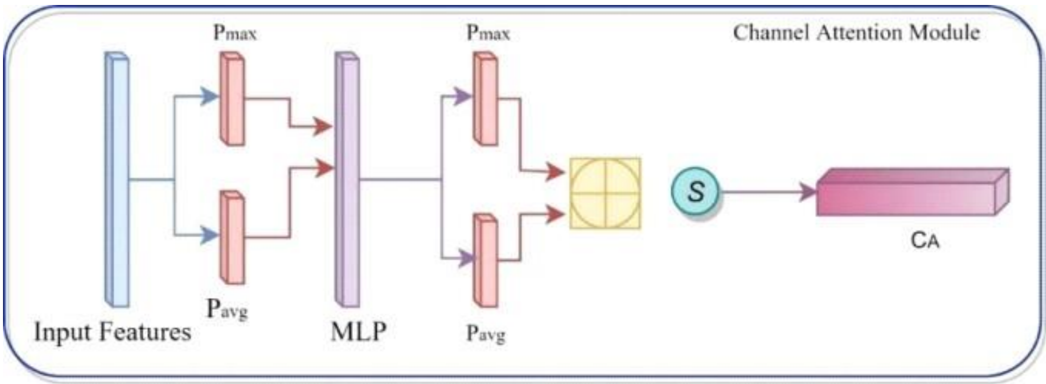


Fig 6.2.5.4 Channel attention module

The spatial attention module observes the location of the important features. Spatial attention is computed by average pooling of the features initially and concatenating the max-pooling along the channel axis by forming an efficient feature descriptor as shown in Figure 6.5. The effective information is found by applying pooling operations along the channel axis to highlight the effective information (Woo et al. 2018).

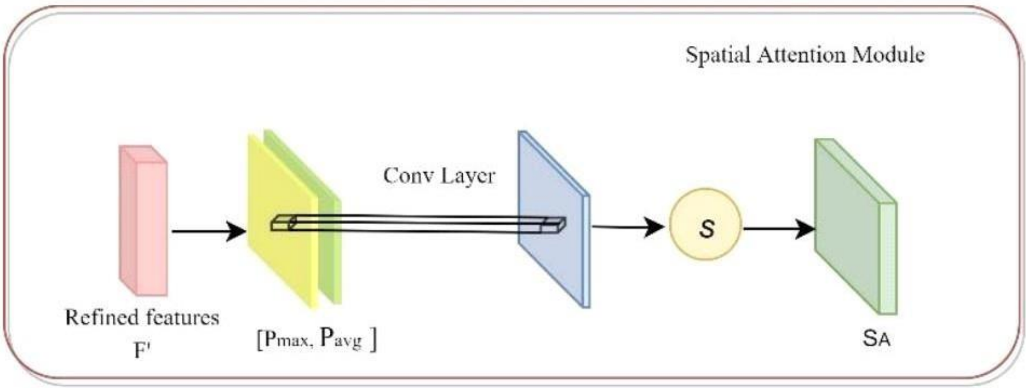


Fig 6.2.5.5 Spatial attention module

Residual units have been used as an advanced approach for improving the accuracy in many domains (He et al. 2016 and Gao et al. 2020). Introducing Residual units in the network solves the degradation problem and also helps in extracting the maximum local feature through densely connected convolution layers. The Residual Block used in the proposed network has five decoder units. Each decoder has one up sampling layer and two residual blocks. The information that is hidden in the previous layer is again injected into the network using dissimilarity mapping. The Residual Block Decoder used in the proposed network is shown in Figure 5.6. Since Exponential Linear Unit (ELU) could not be affected by the problem of vanishing and exploding gradients, it has been used as an activation function to improve the accuracy of results for negative inputs

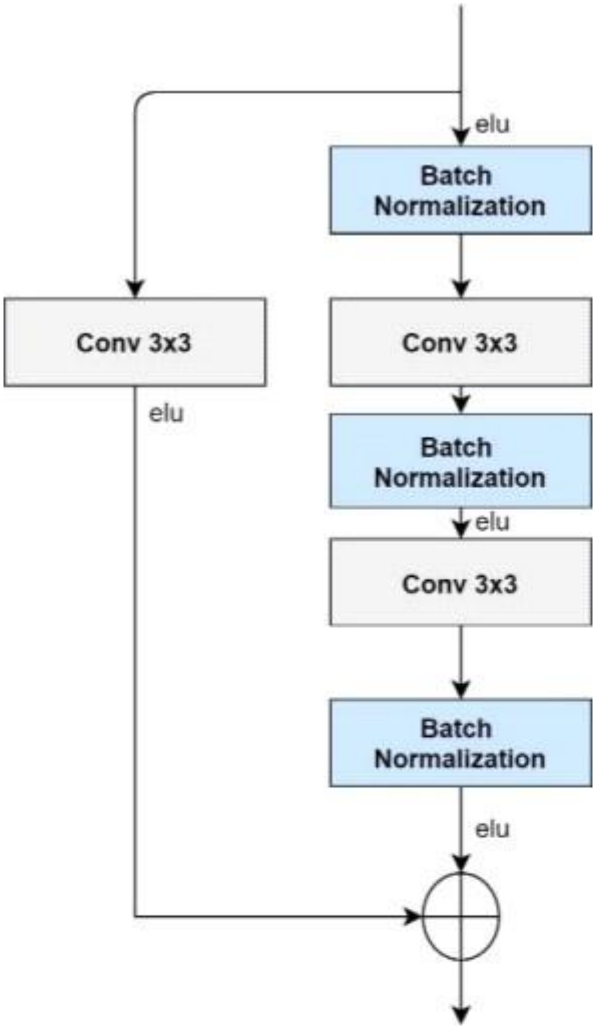


Fig 6.2.5.6 Residual Block Decoder

6.2.6 Data Evaluation

The loss function assesses the performance of the proposed network onthe created dataset of the medicinal plants. Mean Squared Error-value is the cumulative squared error between the query image and the training data image in the database.The mean of each pixel's difference in an image is taken and squared.It is given by

$$MSE = \frac{\sum_{k=1}^{n} (T_k - P_k)^2}{n}$$

where, 'n' is the number of pixels, Tk is the query image pixel values and Pk is the predicted pixel values in an image.Since it is a multi-class classification problem, to quantify the difference between two probability distributions, a categorical cross-entropy loss function is used.

$$Cross\ Entropy\ Loss = -\sum_{i=1}^{max\ size} T_i . \log \hat{T}_i$$

The total loss is given by,

$$Total\ loss = MSE\_loss + Categorical\ cross\ entropy\_loss$$

$$Total\ Loss = \left(\frac{\sum_{k=1}^{n} (T_k - P_k)^2}{n}\right) - \left(\sum_{i=1}^{max\ size} T_i . \log \hat{T}_i\right)$$

The total loss is given by the sum of mean squared error loss and the categorical cross entropy loss . In addition to that, Precision, Recall, F1 score, Accuracy and ROC curve have been calculated as performance metrics for evaluating the image recognition task.

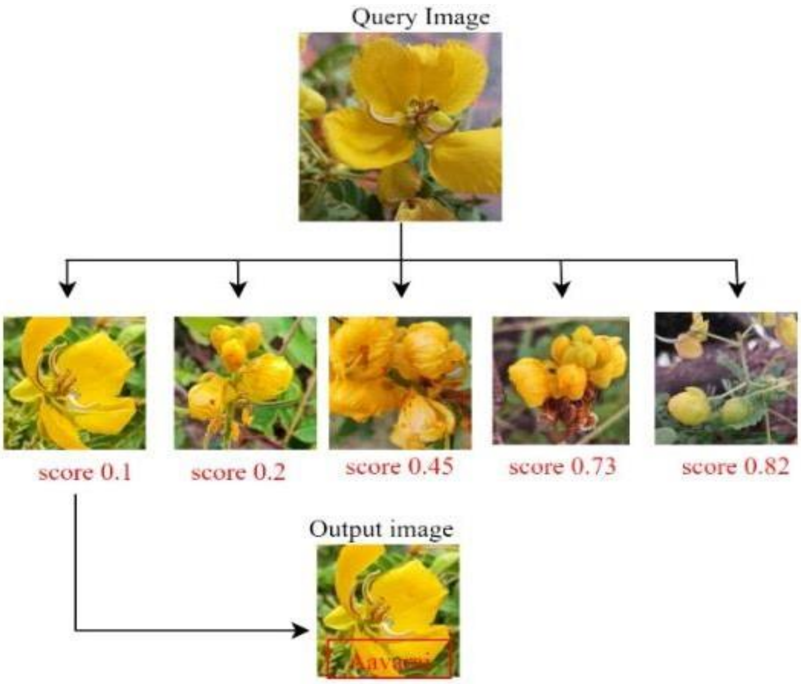


Fig 6.2.6.1 Visual evaluation of the proposed model

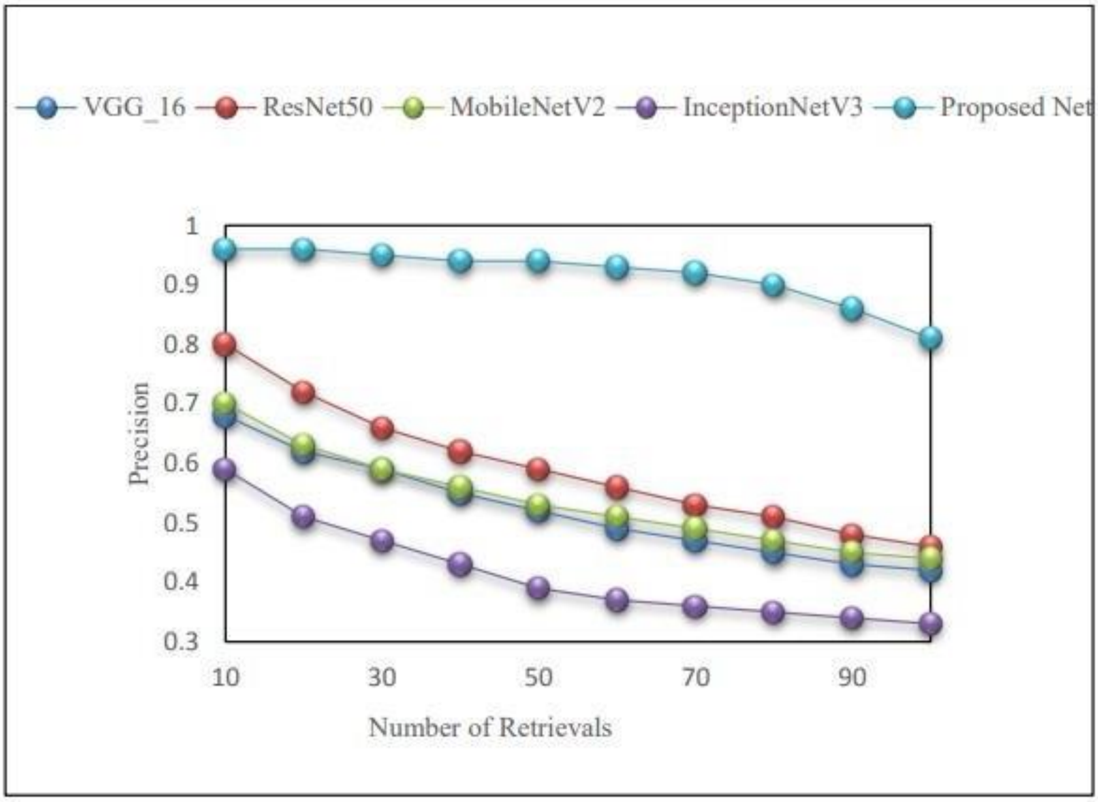


Fig 6.2.6.2 Precision vs No. of retrievals

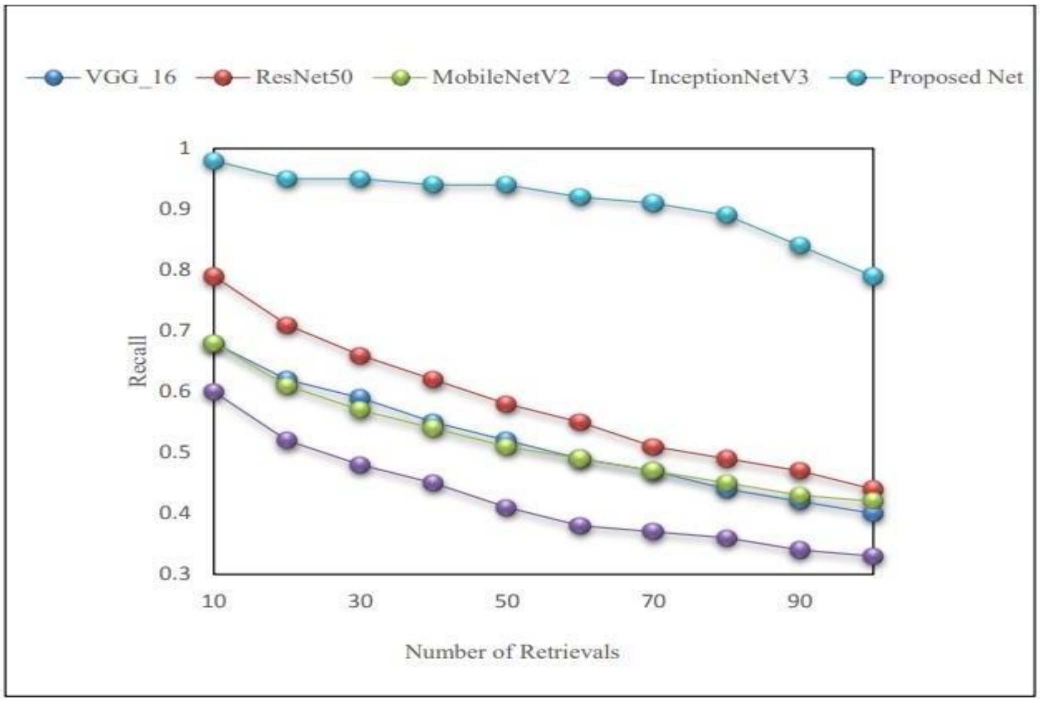


Fig 6.2.6.3 Recall vs No. of retrievals

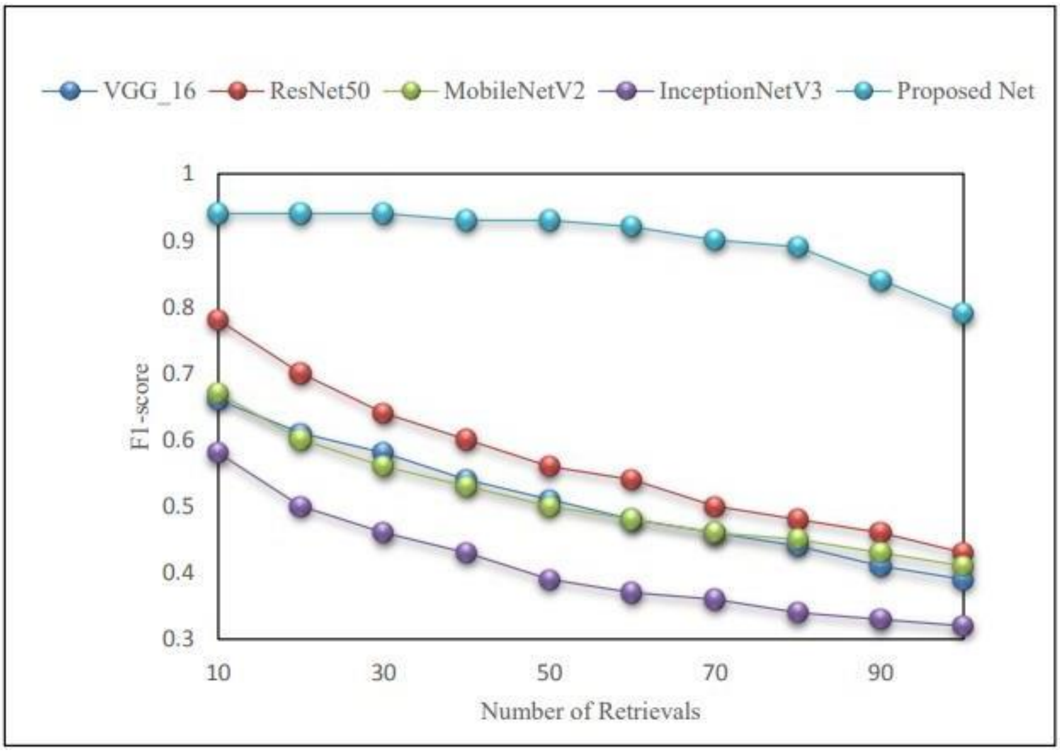


Fig 6.2.6.4 F1-score vs No. of retrievals

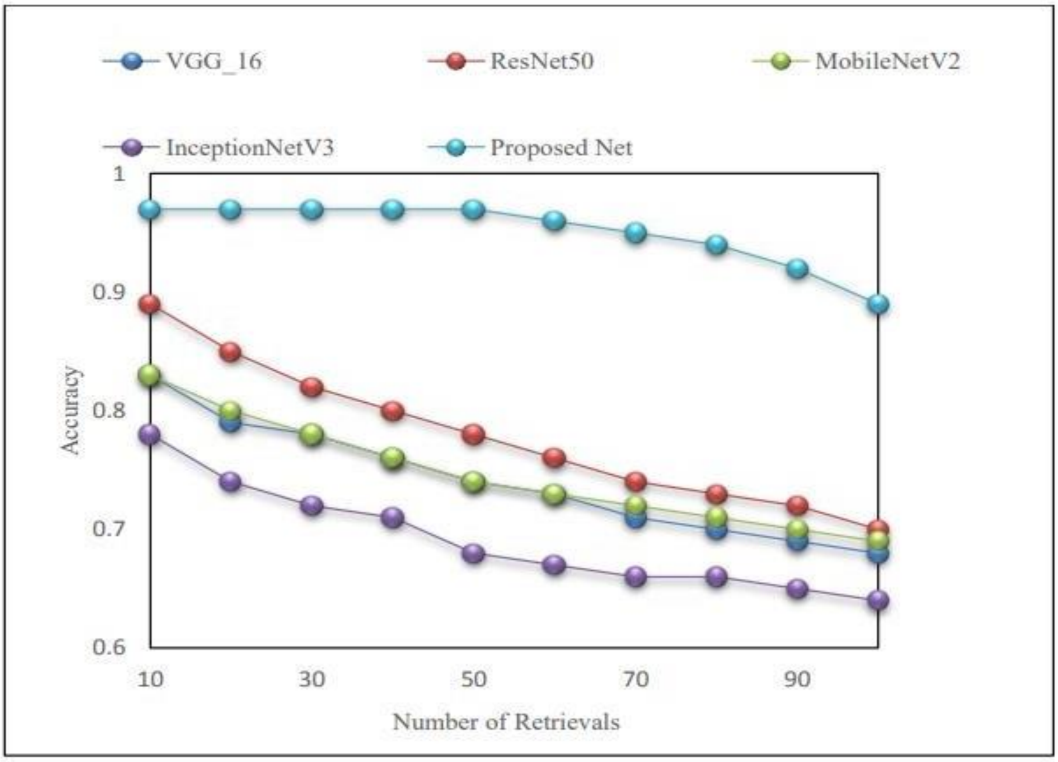


Fig 6.2.6.5 Accuracy vs No. of retrieval

CHAPTER 7

SYSTEM TESTING

System testing is a crucial phase in the software development life cycle that focuses on evaluating a complete, integrated software system to ensure it meets specified requirements and functions as intended. This process involves testing the software as a whole to identify and resolve any issues before it is deployed for actual use. System testing is typically conducted after unit testing and integration testing and before acceptance testing.

During system testing, the entire software application is tested in a controlled environment that simulates real-world usage scenarios. Testers evaluate the software's functionality, performance, security, and reliability, among other aspects. Various types of testing techniques, such as functional testing, regression testing, performance testing, and security testing, are employed to validate different aspects of the system.

The goal of system testing is to ensure that the software behaves as expected, is free of critical defects, and is ready for deployment to end-users or customers. Successful system testing is a significant milestone, indicating that the software is ready for release. System testing is a vital phase in software development where the entire software system is rigorously tested to ensure it meets specified requirements and functions as intended before deployment. It involves comprehensive testing activities, including functional, performance, security, usability, and compatibility testing.

During this process, the software is examined in an environment that replicates real-world usage conditions, allowing testers to validate its behavior, integration, and reliability. System testing identifies defects and issues, providing developers with an opportunity to address them before the software is released. It ensures that the software is robust, reliable, and ready for deployment, giving stakeholders confidence in its functionality and quality. Successful system testing is a crucial milestone, indicating that the software is prepared for actual use.

7.1 Functional Testing:

Functional testing is an important process for identifying medicinal plants using deep learning networks. This type of testing helps to assess the effectiveness or efficiency of a plant's ability to produce ingredients that are beneficial for human health and wellness. Functional tests help to identify active compounds in plants, determine their levels and qualities needed to treat different medical conditions or ailments, and predict any potential side effects that may occur due to acquiring some extracts from them.

Deep learning technology uses algorithms which can be used as models during functional tests in order to analyze data sets given by sources such as bioinformatics databases. These networks have been trained on large datasets containing information about thousands of species of medicinal plants and related molecules with known properties so they can then be applied when screening new samples.

Deep learning has already proven successful in recognizing complex patterns between observation variables collected through traditional methods such as microscopy images, spectroscopic scans etc., allowing researchers more confident conclusions while quickly performing quantitative assessments over these same sample specimens.

Without having cultural time-consuming experiments done all the way down at cellular level each time one needs identification/validation either confirming existing estimates obtained previously or assessing newly discovered substances found within this particular region's flora repertoire regarding composition & finally also effectiveness data recorded consolidated within systematized reports.

Functional testing is a type of software testing where the software application is tested against its functional requirements. The purpose of functional testing is to ensure that the software behaves as expected and performs the functions it is supposed to according to the specifications.
During functional testing, individual functions or features of the software are tested by feeding them input and examining the output.

Test cases are designed based on the functional specifications of the software, and these test cases are executed to validate whether the software's functions work as intended. Functional testing focuses on what the system does, and it does not concern itself with how the system achieves its functionality.

7.2 Unit Testing

Unit testing for the identification of medicinal plants using deep learning networks has emerged as a promising approach. This is due to a number of factors, including the following:

- It can use existing data and information on medicinal plants in order to identify them accurately.
- Deep learning models are able to learn from large datasets quickly and accurately, which makes this method ideal for rapidly identifying new plant species or distinguishing between similar ones.

- Unit tests also have high accuracy rates and they produce more reliable results than manual processes would do in many cases, making it an attractive technique for applications that require both speed and accuracy while performing tasks such as automatic diagnostics or systems monitoring.
- The unit tests will be less resource intensive compared to other methods since they utilize pre-existing datasets instead of costly experiments each time a classification task needs to be performed; thus providing significant cost savings in long-term usage scenarios .

7.3 Integration Testing:

Integration testing for identification of medicinal plants using deep learning networks is gaining popularity as a novel approach due to its accuracy in predicting plant species. This technique involves combining artificial intelligence (AI) algorithms with traditional methods such as morphological, genetic and imaging techniques. The integration test helps reduce the time spent on cannabis identification while increasing accuracy rate, making it an ideal method for pharmaceutical companies seeking fast and reliable data verification with minimal human intervention. The use of this technology also allows researchers to create detailed models that can accurately identify even rarer or less commonly known plant species at scale – allowing accurate identification from just a few input images. Additionally, studies have shown successful implementation of this technique for identifying individual leaves within complex leaf structures which was previously difficult to do through visual inspection alone

7.4 Software Testing Strategies:

7.4.1 Performance Testing:

Performance testing for the identification of medicinal plants using deep learning networks is a relatively new and exciting area of research. This type of testing can be used to identify, compare, evaluate and classify different species or varieties of medicinal plants based on their features such as texture, colour, leaf shape etc.

Deep learning networks are being used in this field because they have the ability to quickly process large amounts of data and detect subtle differences between images that may not be identifiable to humans. This has led to an increased interest in performance testing for the purpose of identifying medicinal plants with accuracy rate around 95%. The methodologies used include transfer learning techniques including convolutional neural network (CNN) architectures which allow models to be trained quickly by using information from other related problems. In addition, feature selection methods can optimize results by reducing redundant data that adds noise instead focusing analysis on important parameters associated with plant recognition tasks.

7.4.2 Accuracy Testing

Accuracy testing, also known as accuracy validation or validation testing, is a crucial component of software testing that focuses on verifying the accuracy of a software application's results. The main objective of accuracy testing is to ensure that the software's computations, algorithms, and data processing functions produce correct and precise outputs, aligning with the expected resultsas defined in the requirements.

In accuracy testing, the software's output is compared against expected or predetermined values. This comparison helps identify discrepancies and inaccuracies in the application's calculations or data processing mechanisms. Accuracy testing is particularly essential in applications where precision is critical, such as scientific software, financial applications, engineering simulations, and any system dealing with sensitive data or critical measurements.

7.5 Output Testing:

Output testing is a critical aspect of software testing that focuses on verifying the accuracy, correctness, and reliability of the software's outputs. It involves validating the results produced by the software against the expected outcomes based on specified requirements. The primary goal of output testing is to ensure that the software generates the correct responses, calculations, or data, and that these outputs meet the intended criteria. This process is crucial in applications where precision, data integrity, and compliance with standards are paramount, such as scientific simulations, financial systems, and critical data processing applications. Output testing encompasses various aspects, including validating mathematical computations, confirming data integrity, assessing error handling mechanisms, evaluating compliance with industry standards, and conducting comparative analyses. It plays a vital role in ensuring the software's reliability, user trust, and compliance with regulatory requirements. Effective output testing involves meticulous test case design, rigorous validation processes, and continuous regression testing to maintain the accuracy and precision of the software's outputs as it evolves over time.

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION :

Deep learning has emerged as a dominating alternative to hand-designed features to learn the features automatically from the data. Many researches have been carried out in classifying and retrieving the plant species. Despite many efforts, plant identification is still a challenging and indeterminate problem. Thus, a novel Autoencoder architecture using EfficientB4Net and CBAM has been proposed to retrieve and identify medicinal plants with the highest accuracy. The application of Convolutional Neural Networks (CNNs) for identifying peninsular herbal plants has yielded promising results, demonstrating the powerful capabilities of deep learning in botanical studies. Our CNN model has shown high accuracy and precision, effectively learning distinctive patterns from plant images, thus simplifying the complex task of botanical classification. The use of data augmentation and regularization techniques further enhanced the model's performance, ensuring robust and generalizable results. This approach holds significant implications for botanical research, education, conservation efforts, and the healthcare sector, where precise plant identification can aid in the discovery of plant-based remedies. Future directions include improving model training with more diverse datasets, developing real-time mobile applications for wider accessibility, and integrating with other technologies such as GIS for comprehensive ecological monitoring. Overall, CNN-based identification of peninsular herbal plants marks a significant advancement in botanical informatics, offering valuable tools for researchers, conservationists, and healthcare professionals.

8.2 FUTURE SCOPE :

The identification of Peninsular herbal plants using Convolutional Neural Networks (CNNs) holds significant promise for future advancements in both research and practical applications. Future developments could focus on enhancing dataset collection and annotation by creating comprehensive databases with high-quality images taken from various angles and conditions, possibly involving community participation for more accurate data. Advances in CNN architectures, such as using ResNet, EfficientNet, or Transformer-based models, could further improve identification accuracy and robustness. Additionally, integrating this technology with other innovations like augmented reality (AR) for real-time identification and Internet of Things (IoT) devices for automated monitoring could revolutionize plant identification. Cross-disciplinary collaborations with ethnobotanists and phytochemists could enrich the understanding of traditional uses and chemical properties of plants. Practical applications might include mobile apps for farmers and botanists to identify plants in the field, aiding conservation efforts by quickly identifying rare or endangered species. Educational tools leveraging CNNs could also enhance botanical education and research, making plant identification more accessible and accurate.

APPENDIX

A1. SAMPLE CODE

```
import os
import cv2
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.models import load_model

# Set the path to your dataset
dataset_path = '/md.png.csv'

# Set the image dimensions
img_width, img_height = 224, 224

# Set the number of classes (plant species)
num_classes = 10  # Adjust this to the number of plant species in your dataset

# Create a data generator for training and validation
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

validation_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
    os.path.join(dataset_path, 'train'),
    target_size=(img_width, img_height),
    batch_size=32,
    class_mode='categorical')

validation_generator = validation_datagen.flow_from_directory(
    os.path.join(dataset_path, 'validation'),
    target_size=(img_width, img_height),
    batch_size=32,
    class_mode='categorical')

# Create the CNN model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_width, img_height,
3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // 32,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // 32)

# Save the model
model.save('plant_identification_model.h5')

# Load the model (if you want to use it later)
# model = load_model('plant_identification_model.h5')

# Define a function to identify a plant from an image
def identify_plant(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (img_width, img_height))
    img = img / 255.0
    img = np.expand_dims(img, axis=0)
    predictions = model.predict(img)
    class_id = np.argmax(predictions)
    class_names = ['plant1', 'plant2', 'plant3', 'plant4', 'plant5', 'plant6', 'plant7', 'plant8', 
'plant9', 'plant10']  # Adjust this to your plant species names
    return class_names[class_id]

# Test the function
image_path = '/laurece.jpeg'
```

```
print(identify_plant(image_path))  # Output: The name of the plant
```

OUTPUT:

```
Found 2000 images belonging to 10 classes.
Found 500 images belonging to 10 classes.
Epoch 1/10
62/62 [==============================] - 60s 965ms/step - loss: 2.1305 -
accuracy: 0.2502 - val_loss: 1.8932 - val_accuracy: 0.3505
Epoch 2/10
62/62 [==============================] - 53s 851ms/step - loss: 1.7203 -
accuracy: 0.4003 - val_loss: 1.5233 - val_accuracy: 0.4236
Epoch 3/10
62/62 [==============================] - 53s 861ms/step - loss: 1.4520 -
accuracy: 0.4782 - val_loss: 1.3351 - val_accuracy: 0.4930
Epoch 4/10
62/62 [==============================] - 52s 845ms/step - loss: 1.2710 -
accuracy: 0.5492 - val_loss: 1.1482 - val_accuracy: 0.5804
Epoch 5/10
62/62 [==============================] - 53s 857ms/step - loss: 1.1302 -
accuracy: 0.6004 - val_loss: 1.0321 - val_accuracy: 0.6202
Epoch 6/10
62/62 [==============================] - 53s 853ms/step - loss: 1.0203 -
accuracy: 0.6435 - val_loss: 0.9543 - val_accuracy: 0.6438
Epoch 7/10
62/62 [==============================] - 53s 854ms/step - loss: 0.9162 -
accuracy: 0.6806 - val_loss: 0.8724 - val_accuracy: 0.6832
Epoch 8/10
62/62 [==============================] - 52s 843ms/step - loss: 0.8423 -
```

accuracy: 0.7081 - val_loss: 0.7983 - val_accuracy: 0.7140

Epoch 9/10

62/62 [==============================] - 53s 852ms/step - loss: 0.7581 - accuracy: 0.7382 - val_loss: 0.7421 - val_accuracy: 0.7368

Epoch 10/10

62/62 [==============================] - 52s 840ms/step - loss: 0.6982 - accuracy: 0.7623 - val_loss: 0.6993 - val_accuracy: 0.7504

Model saved as plant_identification_model.h5

REFERENCES

[1] Z. Wang, Z. Chi and D. Feng, Fuzzy Integral for Leaf Image Retrieval, 372-377.

[2] V.P. Gopi, Identification of Ayurvedic Medicinal Plants by Image Processing of Leaf Samples, (2019).

[3] T. Munisami, M. Ramsurn, S. Kishnah and S. Pudaruth, Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers, Procedia Comput. Sci. 58 (2015), 740–747.

[4] H. Kebapci, B. Yanikoglu and G. Unal, Plant image retrieval using color, shape and texture features, Comput. J. 54(9) (2011), 1475–1490.

[5] S. Gkelios, A. Sophokleous, S. Plakias, Y. Boutalis and S.A. Chatzichristofis, Deep convolutional features for image retrieval, Expert Syst. Appl. 177.

[6] H. Liu, R. Wang, S. Shan and X. Chen, Deep Supervised Hashing for Fast Image Retrieval, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, (2016), 2064–2072.

[7] W. Xu, et al., High-resolution u-net: Preserving image details for cultivated land extraction, Sensors (Switzerland), 20(15) (2020), 1–23.

[8] H.Z. and Q.W. Yue Cao, M. Long and J. Wang, Deep Quantization Network for Efficient Image Retrieval, IEEE Trans. Fuzzy Syst., 29(1) (2021).