Telematics and Informatics Reports 16 (2024) 100181

Contents lists available at ScienceDirect

Telematics and Informatics Reports

journal homepage: www.elsevier.com/locate/teler

Leveraging AI in ayurvedic agriculture: A RAG chatbot for comprehensive medicinal plant insights using hybrid deep learning approaches

Biplov Paneru [a,*], Bipul Thapa [b], Bishwash Paneru [c]

[a] Department of Electronics and Communication Engineering, Nepal Engineering College, Affiliated to Pokhara University, Bhaktapur, Nepal
[b] Department of Computer and Information Sciences, University of Delaware, Newark, USA
[c] Department of Applied Sciences and Chemical Engineering, Institute of Engineering Pulchowk Campus, Affiliated to Tribhuvan University, Lalitpur, Nepal

ARTICLE INFO

Keywords:
Ayurveda science
DeiT
Transfer learning
RAG (retrieval augmented generation), Deep learning

ABSTRACT

Medicinal plants are offering a lot of potential for treatment of various chronic diseases as well as healing wounds, enhancing healthy living for consumers. The Nepalese and Indian agriculture systems are one of the main areas focusing on medicinal plant cultivation, and the abundant availability of these plants in these regions is driving growth in ayurvedic research. Traditional methods for detecting plants as well as generating insights on them are often inefficient and time-consuming due to the manual research need and expertise required in plant and biological lives. In this paper, we develop an advanced LLM (Large Language Model)-powered approach to reliably identify the available medicinal plants and their profitable insights for farmers. We compare multiple deep learning and transfer learning techniques, employing models such as deep convolutional neural networks and advanced transformer models. By training and testing on the dataset that includes varieties of plant types, we select the efficient model after a detailed analysis of a variety of models, dataset split variations, and hyperparameter tuning. The selected model integrates into a retrieval augmented generation (RAG) application capable of providing various insights on the plant identified. The app supports both Nepali and English languages and integrates explainable AI for explaining medicinal plants, their health benefits, and remedies. Results show that the DeiT model achieves 95.97 % accuracy, VGG16 achieves 90.26 %, and a novel hybridized concept with DeiT + VGG16 achieves an accuracy of 96.75 % on a multi-class dataset. The integrated application explains the beneficial insights to users in English as well as local Nepali language.

1. Introduction

India has a long history of growing and harvesting medicinal herbs. The main source of many beneficial medicinal herbs is the Indian Forest. Medicinal plants have long been the focus of extensive study and thought due to their vital function in sustaining human existence [1]. Critical worldwide health requirements are met by medicinal plants, but accurate identification presents challenges [2]. Since it is regarded as a significant problem for medicine manufacturing, authenticity, and conservation, further research is needed into the automatic classification of medicinal plants. Medicinal plants are typically categorized according on their leaf characteristics, which are also a key factor in evaluating their nutrition, contentions, soil-water relationships, preservation strategies, agricultural ecosystems, respiration rate, transpiration rate, and photosynthesis. One of the most important and fundamental steps in plant conservation is species classification [3]. Many of the plants that have historically been used as medicines have bioactive compounds that scientists have found over time. However, it can be challenging to identify plant species based solely on their outward traits, and misidentification can have serious repercussions, including using the wrong plant as medication [4]. Nowadays, people employ experimental approaches to identify medicinal herbs, which are inaccurate and prone to many mistakes. Additionally, using laboratory procedures necessitates professionals and is very expensive and time-consuming. Therefore, it is crucial to use new, non-destructive, quick, and precise techniques like computer vision and artificial intelligence to discern between hazardous and medicinal plants [5].

Machine learning has transformed computer vision. Over the last 8–10 years, we've seen the increasing power of machine learning models supported by high-performance cloud computing platforms like Amazon Web Services (AWS), Google Cloud, IBM Cloud, and Microsoft Azure. Deep learning models can now recognize pictures with accuracy close to,

* Corresponding author.
E-mail address: biplovp019402@nec.edu.np (B. Paneru).

https://doi.org/10.1016/j.teler.2024.100181
Received 6 August 2024; Received in revised form 19 November 2024; Accepted 29 November 2024
Available online 8 December 2024

if not exceeding, human-level skill in some cases. It can do rather well compared to prior mathematically based methods, especially in supervised learning, if correct labels are supplied with the features. These models can successfully manage the increased complexity of exogenous factors such as image background interference and illumination variations caused by capture settings. With the rise of artificial intelligence in this technologically advanced era, there are numerous smartphone apps with various functions that are accessible for the detection, diagnosis, and treatment of plant diseases. However, to ensure the quality of these apps, they must be assessed and categorized according to the correct framework [6].

- In agricultural regions like Nepal, agricultural productivity is heavily impacted by lack of availability of technologies, which can result in significant economic losses for farmers. Despite the potential of AI-driven solutions to aid in profitable medicinal plant identification and management, the availability of accessible, user-friendly applications tailored specifically to Nepali farmers as well as agricultural countries remains limited.
- There is huge research going on in the field of agriculture automation as well as Ayurveda as a remedy for thousands of diseases. Many research claims that ayurvedic plants show beneficial improvements for the treatment of cancer and other life-threatening diseases too. Many existing applications lack localization features, such as language support for Nepali, and are often dependent on stable internet connections, which can be a barrier in rural areas.
- Furthermore, current tools rarely provide offline functionality, which is critical in Nepal's infrastructure-limited agricultural settings. This research aims to bridge this gap by developing a chatbot application that not only provides accurate plant identification but also includes local language support, offline capabilities, and an intuitive interface designed with Nepalese farmers in mind.

1.1. Significance of the research

The Internet provides a vast array of information resources and services that are now available to most of the people, with 90.56 percent of Nepal's population having access to the Internet. Farmers can now use AI-based mobile applications to assist them in various ways, such as detecting medicinal plants and providing possible remedies. The medicinal plants are sources of income for Indian as well as Nepalese farmers. A revolution is required in the field and era in which the Ayurvedic research is growing throughout the world. In addition to the offline classification ability, the app will have features such as the option to save predictions and the ability to switch the app's language between Nepali and English. As people employ experimental approaches to find out medicinal herbs, which is a very inaccurate step, it is also prone to many mistakes [5], and using laboratory procedures necessitates professionals also consumes time and investments. As India and Nepal share similar agricultural practices as well as a good neighborhood relationship, the agricultural production and farming practices are nearly the same in these countries, so the Indian Medicinal plants dataset available from Mendely would have the potential to be part of the development of a model to be implemented in both Nepal's and India's agricultural assistive systems and applications. The dataset featured classes of plants that are abundantly available in the many regions of both countries [1–3].

The solution in the form of a RAG chatbot application is a tool for gaining knowledge on various plant-related beneficial insights, their identification, and their usage for medicinal insights. The solution itself is an intelligent approach utilizing the Langchain OpenAI API, finally utilizing a trained machine learning model and putting the benefits and power of artificial intelligence in the hands of the general public, where such a solution will have the highest impact.

2. Related works

S. Kavitha and Satish Kumar (2024) [1] utilized a deep learning model, and presented a vision-based method for the rapid and accurate identification of medicinal plants. The collection, which focuses on six distinct plants—Indian beech, Curry, Tulsi, Mint, Neem, and Betel—contains 500 photos of each plant that have been resized and enhanced. Following validation, the MobileNet model is uploaded to the cloud and incorporated into a mobile application for real-time recognition, achieving 98.3 % accuracy in detecting these leaves. This automated approach has a lot of potential for botany and machine vision, helping the general public and scientists identify plants.

Trien & Fareed (2024) [2] performed review of various 30 papers in order to provide practical insights, and this systematic review examines on deep learning for automated classification of medicinal plant species. On plant organs like leaves and flowers, convolutional neural networks (CNNs) exhibit testing accuracy of over 90 %, allowing for accurate recognition models. Accuracy losses may be lessened by optimization techniques including data augmentation and ensemble models, even though using crowdsourced data and increasing species variety may provide performance issues.

Sapana & Sheshappa (2022) [3] examined computer learning models that use the texture and form of leaves to categorize medicinal plants. The problem of high redundancy in extracted features is discussed, along with how feature selection approaches might be used to lessen this redundancy. Cross-validation and confusion matrix are used to test and validate various supervised and unsupervised classification techniques. A suggested framework for categorizing medicinal plants is presented in the paper, with a focus on the significance of thorough categorization models for both medicinal and plant conservation.

In order to overcome issues like data privacy and geographic restrictions in deep learning, this study investigates the application of federated learning for image-based classification of medicinal plants. Without centralizing data, federated learning enables model training across several devices. In order to classify medicinal plants, the study uses four deep learning models and analyzes two federated learning strategies, FedAvg and FedProx. When compared to the baseline, the accuracy of the system increases by 5.65 % on IID data and 14.84 % on non-IID data. The increasing difficulty of categorizing medicinal plants using non-IID training data is also highlighted in the study [4].

A deep learning model for plant classification utilizing spatial attention (SA) and channel attention (CA) modules is presented in this work. By concentrating on pertinent input features, the model, which was trained on 900 photos of three different plant classes—weed, toxic, and oregano—improves its efficiency. Global average pooling, mixed pooling, gated pooling, and tree pooling are among the pooling strategies that are included in the CA module. With remarkable accuracy, precision, recall, specificity, and F1-score values of 99.63 %, 99.38 %, 99.52 %, 99.74 %, and 99.42 %, respectively, the Tree-CA method outperformed the others. The study emphasizes how AI-based systems might be used to automatically distinguish between toxic and therapeutic plants [5].

Several studies have employed different methodology on a dataset that does not accurately reflect the range of scenarios that may be experienced in life. This explains why the bulk of results in the literature are nearly perfect, with minimal variation between research. Furthermore, the studies discussed above are confined to the creation of deep learning classification models and do not address deployment. The deep learning models employed in the preceding articles are often bigger in terms of the number of parameters and the file size of the trained model, making them challenging to deploy on a mobile device. Moreover, the datasets examined and employed in the preceding publications are limited in terms of the number of classification classes and pictures accessible to train deep learning models. However, it is clearly known from the preceding studies that deep learning models may be utilized for classification tasks such as recognizing pests from disease pictures, and techniques such as transfer learning and finetuning can be employed to do this.

Out of 668 papers that were obtained from nine reputable electronic databases, 56 primary studies were meticulously chosen. Seven research issues were addressed by means of their analysis and categorization. The findings indicate that 41 % of original research used machine learning methods to identify illness, 32 % employed image sensors to recognize symptoms associated with plant illnesses, and 30 % concentrated on developing novel machine learning models to identify illness. 71 % of the investigations were published in scientific publications, whereas 34 % were assessment studies. Research on the relationship between neural networks and computer vision seems to be a promising area for illness detection. Finally, by offering insights from a methodical mapping of the literature, this paper can provide as a springboard for future research [7].

The samples of diseased tomato leaves are taken into consideration in this study. Based on the early signs, the farmers would be able to identify the infections with ease using these disorder samples of tomato leaves. To enhance the quality of tomato samples, first the leaf samples are scaled to 256 × 256 pixels and then Histogram Equalization is applied. The dataspace is divided into Voronoi cells using the K-means clustering algorithm. Lastly, machine learning techniques like Support Vector Machine (SVM), CNN, and K-Nearest Neighbor (K-NN) are used to classify the extracted features. SVM (88 %), K-NN (97 %), and CNN (99.6 %) are used to test the proposed model's accuracy on tomato disordered samples [8].

In order to identify crop infection, the authors of the experiment looked at CNN, VGG-16, VGG-19, and ResNet-50 models using a dataset of 10,000 images from plants in villages. They found that the accuracy rates for these models were, respectively, 98.60 %, 92.39 %, 96.15 %, and 98.98 %. With an accuracy of 98.98 %, the study shows that ResNet-50 performs better than the other models. In order to anticipate crop diseases in real time, a smart web application was decided to be constructed using the ResNet50 model. By examining images of plant leaves, the suggested web tool seeks to help farmers diagnose plant illnesses. The suggested application classifies the current illness kind and distinguishes between healthy and infected leaves using the ResNet50 transfer learning model [9].

By training a deep CNN using a public dataset of 54,306 photos of healthy and diseased plant leaves that were collected under controlled conditions, we are able to recognize 26 illnesses and 14 crop species (or lack thereof). The trained model shows that this strategy is feasible, achieving 99.35 % accuracy on a held-out test set. All things considered, the method of using publicly accessible, progressively larger image datasets to train deep learning models offers a direct route to widespread, smartphone-assisted crop disease detection [10].

To identify the disease in plant species, a controlled environment dataset was used to train and evaluate all of the deep learning models. Furthermore, various cutting-edge deep learning optimizers were tried to increase the mean average precision of the best-obtained deep learning architecture. using a mean average precision (mAP) of 73.07 %, the SSD model trained using an Adam optimizer demonstrated the highest performance. The uniqueness of the technique was demonstrated by the ability to successfully identify 26 distinct types of defective and 12 types of healthy leaves in a single framework. The suggested detection technology may be used for further agricultural applications in the future. Furthermore, in a controlled or uncontrolled setting, the produced weights can be utilized again for real-time plant disease identification in the future [11].

To guarantee the model's dependability, it was trained using labeled data and assessed using the proper metrics. To determine which CNN architecture was best, a number of them were compared and assessed. MobileNetV2 was utilized in BotaniCare to construct and evaluate the model, and with the application of transfer learning, a training accuracy of 98.7 % and a validation accuracy of 96.4 % was attained. Through extensive evaluation utilizing real-world photos and validation against professional diagnosis, BotaniCare's high accuracy and dependability in disease prediction were demonstrated. It is expected that farmers in Mauritius would make extensive use of the smartphone application to recognize prevalent plant diseases and implement suitable preventative measures [12].

A dataset containing three maize crop diseases—Leaf Spot, Sugarcane Mosaic Virus, and Blight—was gathered at the University Research Farm Koont, PMAS-AAUR during various growth phases and under various meteorological circumstances. YOLOv3-tiny, YOLOv4, YOLOv5s, YOLOv7s, and YOLOv8n were among the prediction models trained on this data; the corresponding reported prediction accuracy was 69.40 %, 97.50 %, 88.23 %, 93.30 %, and 99.04 %. Findings show that compared to previous applicable models, the YOLOv8n model has a greater prediction accuracy. With a higher confidence score and accurate localization of the leaf's damaged area, this model has demonstrated good outcomes. YOLOv8n is the most recent model utilized for disease diagnosis when compared to other methods found in the literature [13].

On the testing dataset, the MVGG16 DTL method yielded results of 98.79 %, 97.93 %, 98.44 %, 98.95, and 98.63 % for accuracy, precision, recall, f1-score, and AUC, respectively. Then, a smartphone app that allows users to input photos, acquire a diagnosis, and review a history of previous diagnoses was integrated with the MVGG16 DTL model. We put the software to the test on a set of images of habanero plant leaves, and found that it was very good at identifying diseased plants. The program for smartphones can improve the early detection and management of habanero plant diseases, leading to increased crop yield and superior harvests [14].

With the localization of CNNs and the worldwide connection of vision transformers, the authors of this work provide a CNN-Transformer hybrid model that is both incredibly reliable and effective. We build our attention mechanism using an effective convolution operation to jointly attend to information in different representation subspaces while reducing the high quadratic cost of the self-attention mechanism. Furthermore, authors try to learn smoother decision boundaries to mitigate the vulnerability of our Transformer model to adversarial attacks. To do this, they permute the feature mean and variance within mini-batches in order to enhance the shape information of an image in the high-level feature space. On a vast collection of standardized MedMNIST-2D datasets, this proposed hybrid model shows its great robustness and generalization ability with less computational cost than the state-of-the-art studies [15].

This study assesses the effectiveness of four cutting-edge pre-trained deep learning models on the VNPlant-200 dataset, a complex dataset that includes different species of medicinal plants photographed in their natural environments: EfficientNetBO, EfficientNetV2-S, Vision Transformer (ViT), and Bidirectional Encoder Image Transformer (BEiT). Our findings demonstrate that BEiT outperformed the other models assessed on this benchmark, with the maximum accuracy of 99.14 %. These results demonstrate how well these models perform in plant recognition tasks, especially when it comes to medicinal plants [16].

Several studies are employing the same methodology on a dataset that does not accurately reflect the range of scenarios that may be experienced in life. This explains why the bulk of results in the literature are nearly perfect, with minimal variation between research. Furthermore, the studies discussed above are confined to the creation of deep learning classification models and do not address deployment. The deep learning models employed in the preceding articles are often bigger in terms of the number of parameters and the file size of the trained model, making them challenging to deploy on a mobile device. Moreover, the datasets examined and employed in the preceding publications are limited in terms of the number of classification classes and pictures accessible to train deep learning models. However, it is clearly known from the preceding studies that deep learning models may be utilized for classification tasks such as recognizing significant plants, and techniques such as transfer learning and finetuning can be employed to do this. Moreover, previous study uses CNN for local feature detection. However, we use both the local and global feature detection using the VGG16 and transformer-based DeiT model for more holistic approach.

3. Methodology

As, medicinal crops and plants have a great beneficial potential in the field of medicinal science but lack of AI-based research in this field shows a great need of research for development of assistive tools for their production. Due to the manual selection of relevant feature sets, traditional agricultural plants identification approaches are inefficient, and time-consuming. This study works on developing an advanced medicinal plant recognition approach using DeiT and VGG16 transfer learning model (CNN) to reliably detect various medicinal plants by image classification technique. We aim to compare the properties of three different deep learning and transfer learning techniques by training and testing on a extensive datasets and the final efficient model is selected after detailed analysis of a variety of models and the impact of dataset split variations, hyperparameter tuning. The model integrates with the RAG-based application chatbot which is capable of classifying the plant by uploading image and then, generating insights based on retrieval document and then, utilizing OpenAI API for generation of various insights.

3.1. DataSet collections

The Indian Medicinal Plant dataset [17] is utilized for the purpose of developing a deep learning model for predictions through system backend and RAG is utilized for generating insights. With the help of the RAG application the model operates from the backend of the system to make prediction and based on prediction and RAG retrieval document file the generation of insights are continuously updated. The dataset distribution phenomenon is shown in the Table 1.

For training: 4161 images belonging to 40 classes were utilized and for testing: 893 images belonging to 40 classes are utilized. The training include 70 % dataset and 15–15 % of total of images for the model development. The model details and division phenomenon are shown in Table 1 and various classes of plants utilized along with the number of images used are shown in the Table 2.

3.2. Data pre-processing and features extraction

Data pre-processing and data pipelining have several stages, such as the dataset preparation stage, the image loading and label generation stage, and the data preprocessing stage. The dataset was obtained and preprocessed with steps as seen in Table 3; then, the features (images) were extracted with the proper distinction for need of training testing and validation. The overall workflow is given in Fig. 1.

3.3. Model training

The images and labels now loaded into proper format are converted into tensors. But to use those tensors for the training process, we need to divide the training set into batches. While training any machine learning model, if we pass our dataset as it is, and if the size of the dataset is large as in our case, the memory will be quickly filled up and the training could be successfully completed. Therefore, the training set and validation set are divided into proper batches. The value of the batch is given as per the training case scenario. To create the training data pipeline, data splitting at ratio of 70:15:15 is carried out where 70 % is for training, and 15 % is for validation and test respectively.

Table 1
Data distribution.

Details	Value
Number of training images (70 %)	4161
Number of images in validation set (15 %)	891
Number of images in test set (15 %)	893

Table 2
Number of images in dataset.

Name	Number of Images
Aloevera	164
Amla	146
Amruta Balli	146
Arali	146
Ashoka	146
Ashwagandha	146
Avocado	146
Bamboo	146
Basale	146
Betel	151
Betel Nut	146
Brahmi	146
Castor	160
Curry Leaf	146
DoddaPatre	146
Ekka	146
Ganike	115
Guava	146
Geranium	146
Henna	150
Hibiscus	165
Honge	146
Insulin	146
Jasmine	187
Lemon	146
Lemon Grass	146
Mango	146
Mint	153
Nagadali	152
Neem	146
Nithyapushpa	146
Nooni	146
Pappaya	146
Pepper	146
Pomegranate	146
Raktachandini	146
Rose	168
Sapota	146
Tulasi	146
Wood Sorel	146

A novel architecture utilizing DeiT and VGG16 is proposed as a hybridized model for the development of the RAG application chatbot. By adding a retrieval component that retrieves pertinent data from outside sources, RAG chatbots increase the capability of language models and are especially useful for providing accurate insights. With the help of the Google Translator Python module, a RAG chatbot can offer bilingual assistance in both Nepali and English. Users can ask inquiries in either language and get answers in the language of their choice thanks to this configuration, which facilitates seamless language transfers. Based on user input, the RAG chatbot can utilize a trained model to forecast pertinent information about medicinal plants, including possible applications or therapeutic qualities. This chatbot combines retrieval and generation to retrieve scientific data on plants and describe it based on the model's predictions. It provides expert-backed insights in an intuitive interface that is customizable in terms of language with help of OpenAI API as seen in Fig. 2.

3.4. Models utilized

i. **VGG16**

A deep CNN architecture known as the VGG16 model became well-liked due to its ease of use and efficiency in image categorization applications. Convolutional layers with tiny 3 × 3 filters make up the majority of its 16 layers, which are followed by completely connected

Table 3
preprocessing steps used in the provided script.

Step	Action	Values/Details
Device Configuration	Checks if CUDA is available and sets the device to either CPU or GPU.	Device: cuda (if available) or cpu
Random Seed Initialization	Sets a seed for reproducibility using torch.manual_seed(), np.random.seed(), and other related functions.	Seed value: 42
Data Transformation (Train)	Applies random transformations like resized crop, horizontal flip, rotation, normalization, and tensor conversion.	- RandomResizedCrop: Image size 224 × 224 - RandomHorizontalFlip: 50 % chance - RandomRotation: Max rotation of 10 ° - ToTensor() - Normalize: Mean [0.485, 0.456, 0.406], Std [0.229, 0.224, 0.225]
Data Transformation (Test)	Applies resizing, normalization, and tensor conversion to test data.	- Resize: Image size 224 × 224 - ToTensor() - Normalize: Mean [0.485, 0.456, 0.406], Std [0.229, 0.224, 0.225]
Dataset Loading	Loads the dataset using datasets.ImageFolder() and applies train transformations.	Dataset Directory: ../input/indian-medicinal-leaves-dataset/Indian Medicinal Leaves Image Datasets/Medicinal plant dataset
Dataset Split	Splits the dataset into training, validation, and test sets using random_split().	- Train size: 70 % of dataset - Validation size: 15 % of dataset - Test size: Remaining 15 % of dataset
Data Loader Creation	Creates DataLoader instances for training, validation, and test sets with specified batch size and parallel loading.	- Batch size: 32 - Number of workers: 2 - pin_memory: True
Model Initialization	Initializes a modified Vision Transformer (DeiT) model with dropout and a custom classification head.	Pre-trained model: deit_base_patch16_224 - Dropout: 0.3 - Number of output classes: Equal to the number of classes in dataset
Model Freezing	Freezes all parameters except the classification head and dropout layer during training.	All layers except fc and dropout are frozen.
Loss Function and Optimizer	Defines the cross-entropy loss function and AdamW optimizer for training the model.	Loss function: CrossEntropyLoss() Optimizer: AdamW Learning rate: 1e-4 (for fc parameters only)

layers. It was created by Oxford's Visual Geometry Group (VGG). The 3 × 3 filters guarantee a fine-grained feature extraction, and the model stacks these layers to capture complex patterns and hierarchical features in images. The depth and homogeneity of the VGG16 architecture are well-known, which makes it simple to use and incredibly efficient in a range of computer vision applications.

ii. **DeiT**

In contrast to conventional transformers, the DeiT (Data-efficient Image Transformer) model is a vision transformer (ViT) architecture designed to function well with minimal training data. DeiT, created by Facebook AI, does not rely on convolutional layers but rather employs self-attention techniques to capture associations across visual patches. DeiT incorporates a teacher-student distillation technique, which trains the model utilizing soft labels from a neural network (such as ResNet) as a guide, in contrast to standard ViTs. Even in situations with insufficient data, DeiT is efficient and effective for visual tasks thanks to this method, which allows it to achieve high accuracy on picture classification tasks. The DeiT architecture is given in Fig. 3.

iii. **VGG16+DeiT**

The VGG16 + DeiT (Data-efficient Image Transformer) hybrid model improves both spatial and contextual understanding of pictures by fusing the CNN-based feature extraction of the VGG16 with the Transformer-based methodology of the DeiT. In this hybrid, low-level features are extracted from photos using VGG16 as a front-end. The images are also fed into DeiT, a transformer designed for visual tasks requiring less data. DeiT layers capture global dependencies and fine-grained contextual interactions inside images by further analyzing the inputs. The features obtained from both VGG16 and DeiT are concatenated along with a dropout and fully connected layer for the classification. Improved accuracy and robustness are made possible by the combination of transformer and convolutional architectures, especially in challenging picture classification and identification applications.

3.5. *Performance evaluation metrics for model*

i. **Accuracy**

Accuracy:
The number of right predictions among all the forecasts made is the measure of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{(FP + FN + TP + TP)}. \tag{3}$$

ii. **Precision**

Precision is the ratio of true positives to all positive predictions, and it is sometimes also referred to as positive predictive value. It depicts how well the model predicted the good results.

$$\text{Precision} = \frac{TP}{(FP + TP)} \tag{4}$$

iii. **Recall**

Recall quantifies the percentage of real positive cases that the model correctly classified; it is sometimes referred to as Sensitivity and also called as True Positive Rate that final shows ability of model to capture good examples.

$$\text{Recall} = \frac{TP}{(FN + TP)} \tag{5}$$

iv. **F1-Score**

Precision and recall are harmonic means, and the F1 score performs a balance between both. The harmonic mean of precision and recall gives the F1-score.

$$\text{F1 Score} = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

Where,
TP = True Positives
TN = True Negatives
FP= False Positives
FN = False Negatives

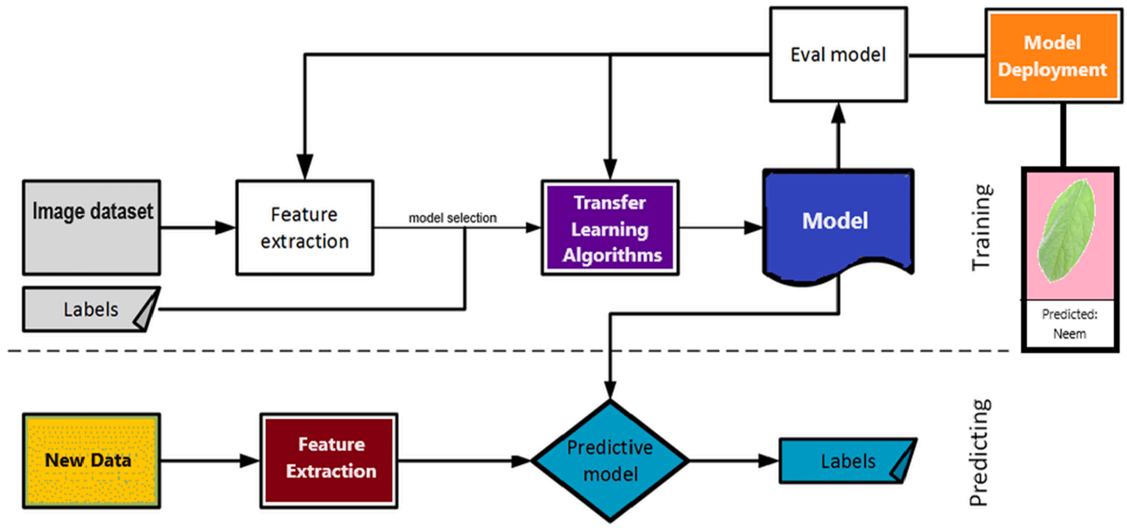

Fig. 1. Proposed workflow.

Fig. 2. OpenAI integration for Chatbot.

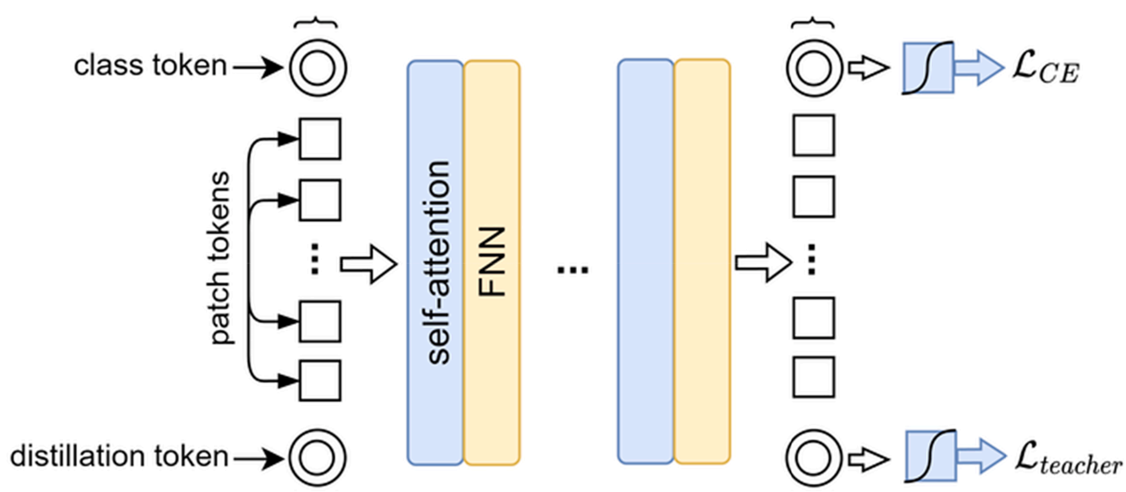

Fig. 3. DeiT model architecture [18].

3.6. *Googletrans language conversion*

Using Google Translate's API, the googletrans Python module is an effective tool for translating text between different languages. It is well-liked for applications requiring rapid, automated translations and supports more than 100 languages. The library's Translator class can be used to translate an English discussion into Nepali. This allows you to choose 'ne' (Nepali) as the target language for every English sentence or phrase. This makes it possible to translate discussions into Nepali in real time or in batches, which is helpful for developing multilingual user interfaces or apps that adapt to different languages.

4. Results and discussions

The various models are tested for the purpose of yielding a good performance on model and the models are fine tuned with various methods. The model successfully obtains accuracy beyond 90 % on testing sets. The model classifies the medicinal plant and recommends their usages with the help of predicted classes, using the RAG that provides economical insights for the farmers. These results show that model can be used for the purpose of deployment in the application. The Table 4 depicts the performance of each models utilized. The chatbot conversation can be switched to Nepali too utilizing the googletrans library in python.

As seen in Table 4, with a high training accuracy of 99.30 % and a validation accuracy of 95.97 %, the DeiT model demonstrates that it effectively learn the training data and make fair generalizations to the validation data. With a validation accuracy of 96.75 % and a training accuracy of 99.57 %, the hybridized DeiT + VGG16 model outperforms DeiT, demonstrating even more generalization.

4.1. DeiT model performance

For both training and validation datasets, the DeiT model's performance graphs show accuracy and loss across dataset over 57 epochs, with early stopping applied using a patience of 7 epochs. As shown in Fig. 4, the loss plot for training and validation decreases sharply in the initial epochs, showing rapid improvements as the model learns. Around epoch 10, the loss values for both training and validation stabilizes and continue to decrease at slower rates. By the end of training, both training and validation loss remains low, with minimal divergence between the two curves, indicating that the model generalizes well without significant overfitting

The accuracy plot shows the training and validation accuracy for the training, displaying steep increase in the first 10 epochs. After this, the accuracy for both datasets continues to rise gradually, eventually stabilizing close to 1.0 as the training progresses. The training and validation accuracy curves remain close to each other throughout, suggesting that the model maintains a good balance between fitting the training data and generalizing to the validation data.

The DeiT model's confusion matrix for the test dataset is shown in Fig. 5. The figure illustrates DeiT model's classification performance across multiple classes. Each row represents the true labels, while each column corresponds to the predicted labels. The diagonal entries contain the correctly classified instances for each class, highlighted by the values in blue, which show a high level of accuracy across most classes. For instance, classes like "Aloevera," "Ashwagandha," and "Pepper" demonstrate perfect or near-perfect accuracy with all or nearly all test samples correctly classified. Off-diagonal entries, which represent misclassifications, are minimal, indicating the model's strong ability to differentiate between these classes. However, a few misclassifications are observed, which may indicate confusion between visually similar or semantically related classes, suggesting poor collected images. Overall, the matrix highlights a well-performing model with strong predictive accuracy, effectively capturing the distinctions across diverse categories in the test set. The sparse distribution of misclassifications outside the diagonal suggests that the DeiT model has generalized well to this dataset, with limited overfitting and a robust understanding of class-specific features.

4.2. DeiT + VGG16 model

The DeiT + VGG16 model's training history curves is shown in Fig. 6, which span 51 epochs with an early-stopping criterion set at a patience of 7 epoch. The figure illustrates the evolution of accuracy and loss for training and validation datasets. The accuracy plot shows that over the first 10 epochs, both training and validation accuracy rise quickly, reaching roughly 95 % for training and 93 % for validation. After this point, the accuracies gradually converge and reach plateau with both curves showing consistently high value. This shows the model effectively learns key features of the data without significant overfitting.

The loss plot shows a similar pattern, with both the training and validation losses decreasing in the initial epochs. By epoch 10, the loss stabilizes around 0.6 and in later epoch the training loss almost falling to zero. The close alignment between training and validation loss curves, particularly in later epochs, indicates good generalization, as the validation loss does not increase significantly. Overall, the graph exhibits strong learning performance with rapid convergence and minimal overfitting.

The efficacy of a DeiT +VGG16 model as confusion matrix in forecasting forty classes of various medicinal plants displayed in Fig. 7. The anticipated class is represented by each column, and the actual class is represented by each row. The correctly identified examples are shown by the diagonal elements, which show great accuracy for classes such as Aloevera, Ashwagandha, and Guava. Few errors such as misclassifying Heena and Nagadali, show minimal misclassification. Each cell's color intensity represents the number of occurrences; greater values are displayed by deeper hues. The sparse off-diagonal errors suggest that the model effectively generalizes across categories, with only minor confusion in visually similar classes, highlighting its robustness and accuracy as compared to the DeiT and VGG16 model individually.

Table 4
Performance comparison of different models for classification.

Model	Training Accuracy	Test Accuracy
DeiT	99.30 %	95.97 %
VGG16	99.71 %	90.26 %
DeiT + VGG16 hybridized	99.57 %	96.75 %

4.3. VGG16 model

The plots in Fig. 8 show the training and validation loss and accuracy curves for a model over 22 epochs. In the loss plot, the training loss decreases sharply in the initial epochs, reaching close to zero by around epoch 5, indicating rapid learning and optimization. The validation loss also drops quickly initially but then fluctuates and begins to increase slightly after around epoch 10. This divergence between training and validation loss suggests that the model may be starting to overfit the training data after this point, as it continues to improve on the training set but not on the validation set.

In the accuracy plot, training accuracy climbs steeply, approaching 1.0 by around epoch 5, showing that the model learns the training data very well. The validation accuracy also increases initially, reaching a stable level around 90 %. However, it exhibits slight fluctuations in the later epochs, which aligns with the increase in validation loss. This further supports the likelihood of overfitting, as the model performs almost perfectly on training data while achieving slightly less stable performance on validation data. Overall, the graphs indicate that while the model trains effectively, achieving high accuracy and low loss on the training set, it may be starting to overfit after a certain point, as seen in the increasing validation loss and fluctuating validation accuracy.

The confusion matrix for the test dataset using VGG16 model is shown in Fig. 9. as confusion matrix in forecasting forty classes of medicinal plants is displayed in Fig. 6. Most predictions fall along the diagonal, indicating correct classifications, which shows the model's high accuracy. Classes such as Aloevera, Ashwagandha, Betel Nut, and Guava show high correct predictions, with minimal errors, highlighting strong model performance for these categories. However, a few off-diagonal entries indicate occasional misclassifications, particularly in classes like Henna, Hibiscus, and Neem, where the model misclassifies a small number of instances into other categories. These errors are minimal but suggest some overlap in feature representation or visual similarity between certain classes. The model classes are compared as seen in table 5.

Comparison of classification report

The model architecture for DeiT+VGG16 is shown in Fig. 10. The model consisted up of input layers along with the concatenation layer. Dropout was added as 0.3 for the model performance improvement and the fully connected layer was utilized before output layer for developing final model.

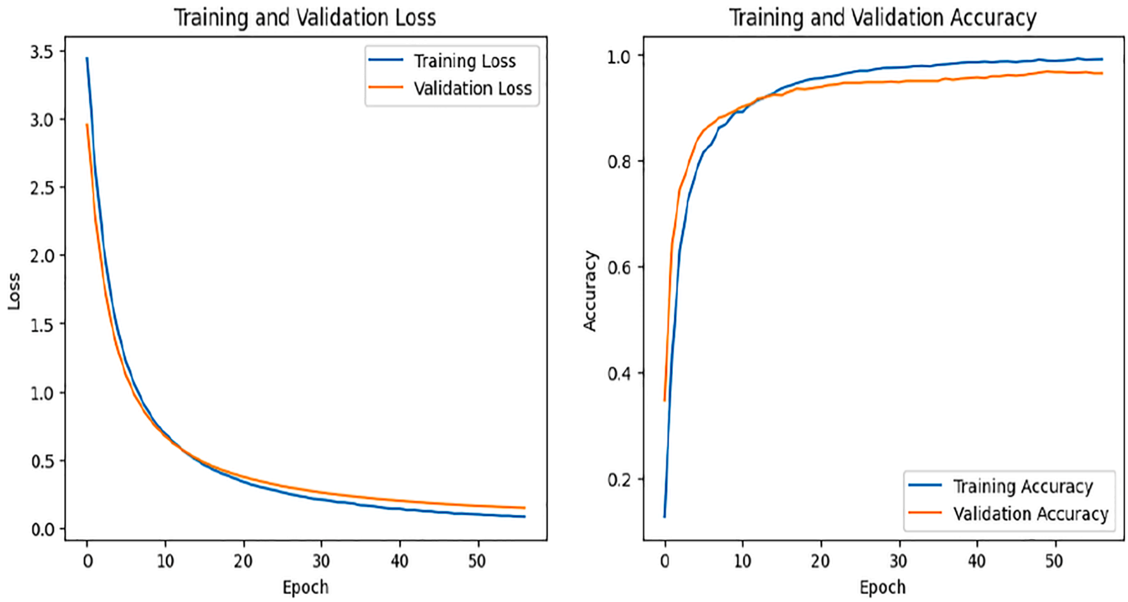

Fig. 4. Model history plot.

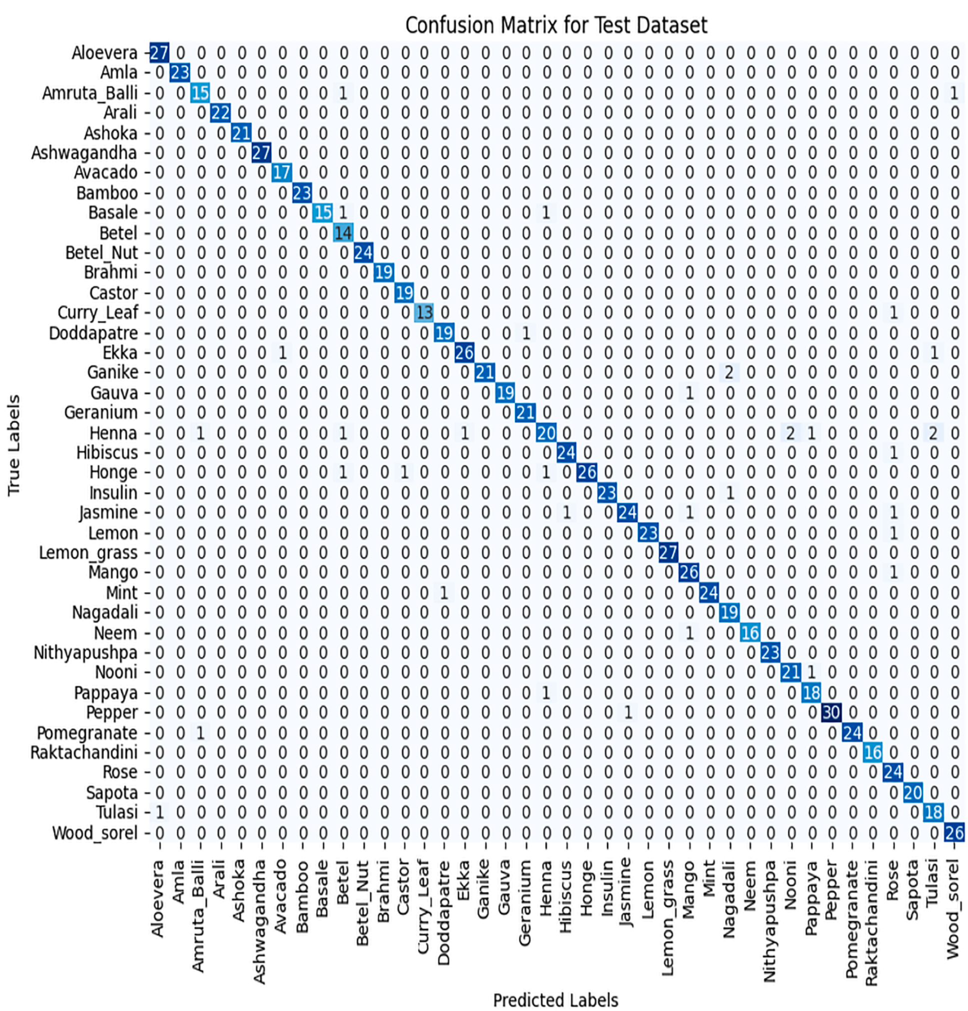

Fig. 5. Confusion matrix plot for DeiT.

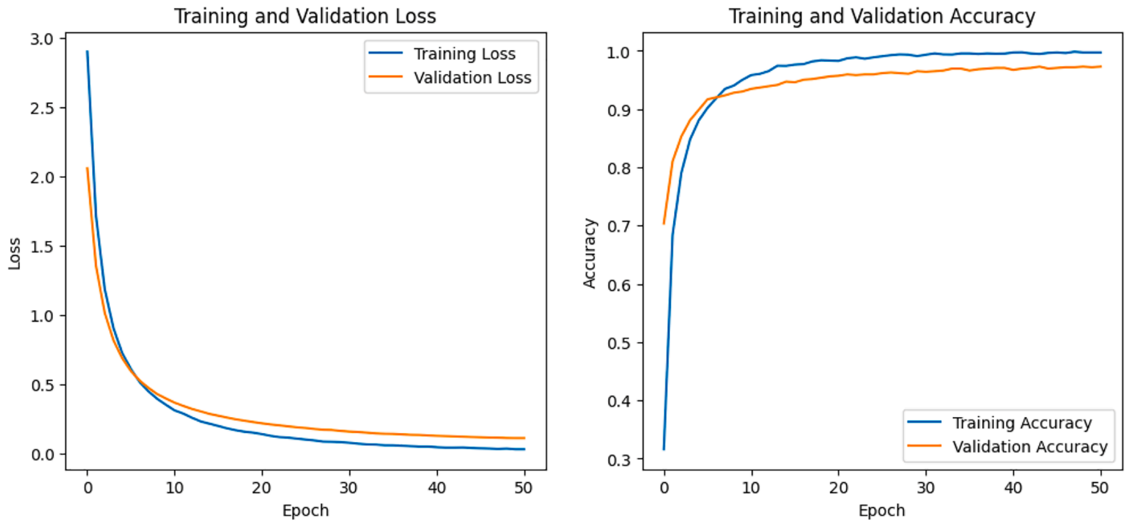

Fig. 6. Model history plot.

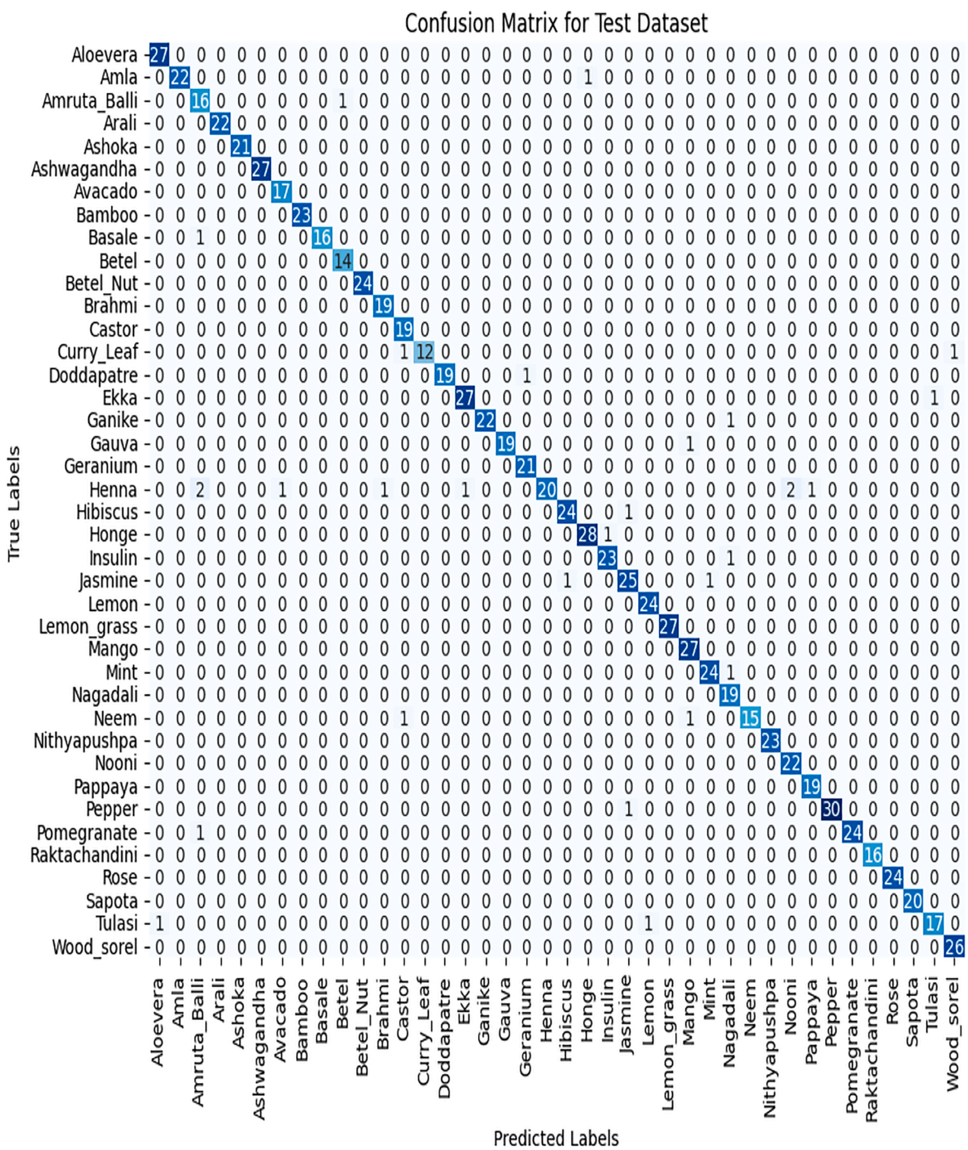

Fig. 7. Confusion matrix plot for Deit +VGG16 hybridized model.

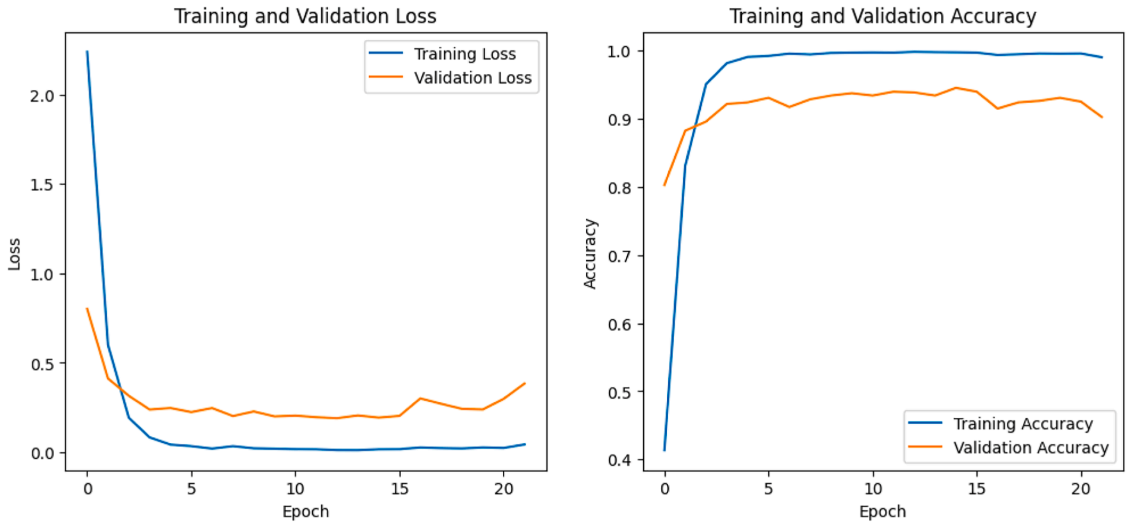

Fig. 8. Model history plot for VGG16.

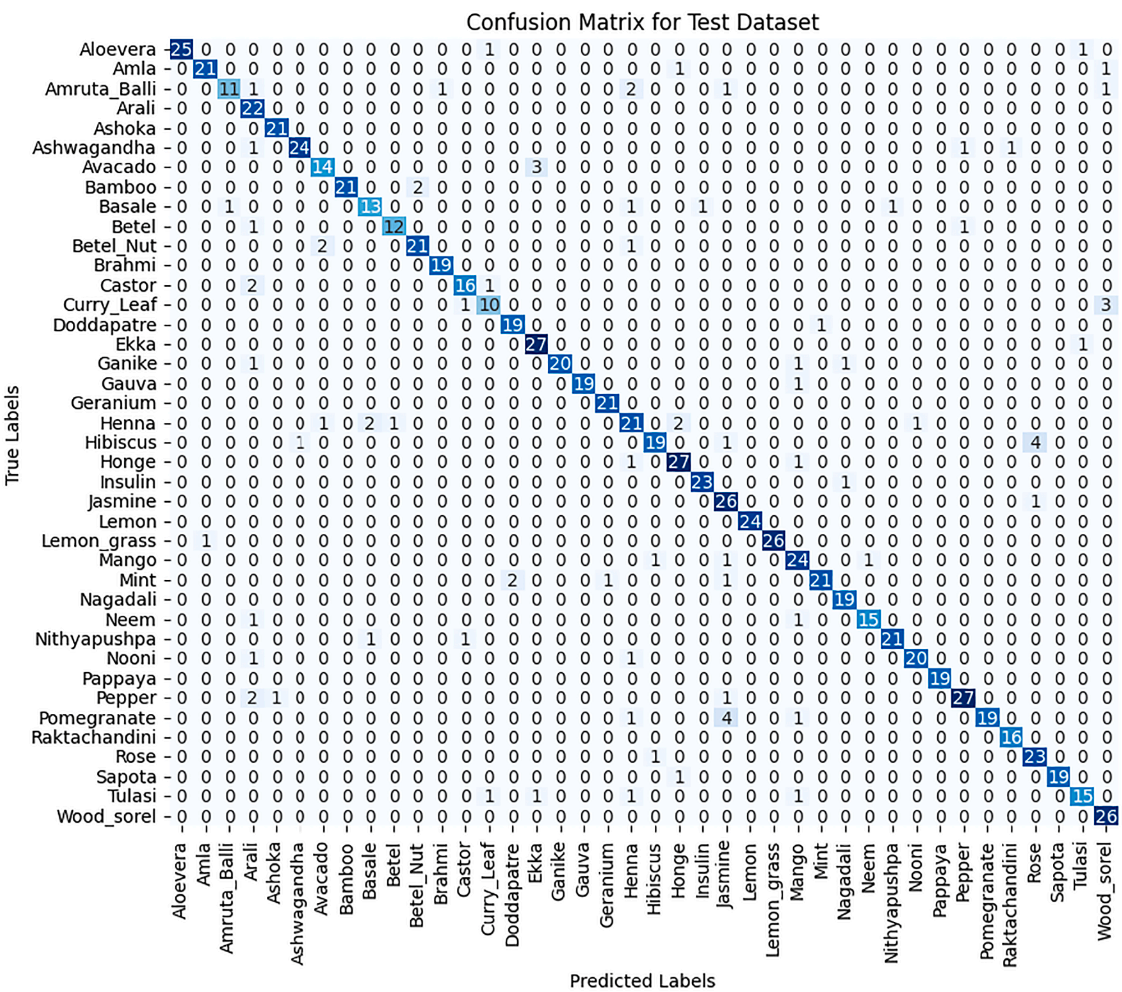

Fig. 9. Confusion matrix plot for VGG16 model.

4.4. Model cross validation performance for DeiT+VGG16

The model performance on various folds is shown in Table 6. The model showed great performance across each fold, as indicated by the accuracy, precision, recall, and F1-score values. The model's average accuracy, precision, recall, and F1-score of approximately 95 % shows how well the model's performance fits on every fold, and this is an important phenomenon to analyze the ability of the model for making overall predictions.

Table 5
Comparison table for performance between the DeiT + VGG16 and DeiT models.

Class	DeiT + VGG16				VGG16			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Aloevera	0.96	1.00	0.98	27	1.00	0.93	0.96	27
Amla	1.00	0.96	0.98	23	0.95	0.91	0.93	23
Amruta_Balli	0.80	0.94	0.86	17	0.92	0.65	0.76	17
Arali	1.00	1.00	1.00	22	0.69	1.00	0.81	22
Ashoka	1.00	1.00	1.00	21	0.95	1.00	0.98	21
Ashwagandha	1.00	1.00	1.00	27	0.96	0.89	0.92	27
Avacado	0.94	1.00	0.97	17	0.82	0.82	0.82	17
Bamboo	1.00	1.00	1.00	23	1.00	0.91	0.95	23
Basale	1.00	0.94	0.97	17	0.81	0.76	0.79	17
Betel	0.93	1.00	0.97	14	0.92	0.86	0.89	14
Betel_Nut	1.00	1.00	1.00	24	0.91	0.88	0.89	24
Brahmi	0.95	1.00	0.97	19	0.95	1.00	0.97	19
Castor	0.90	1.00	0.95	19	0.89	0.84	0.86	19
Curry_Leaf	1.00	0.86	0.92	14	0.77	0.71	0.74	14
Doddapatre	1.00	0.95	0.97	20	0.90	0.95	0.93	20
Ekka	0.96	0.96	0.96	28	0.87	0.96	0.92	28
Ganike	1.00	0.96	0.98	23	1.00	0.87	0.93	23
Gauva	1.00	0.95	0.97	20	1.00	0.95	0.97	20
Geranium	0.95	1.00	0.98	21	0.95	1.00	0.98	21
Henna	1.00	0.71	0.83	28	0.72	0.75	0.74	28
Hibiscus	0.96	0.96	0.96	25	0.90	0.76	0.83	25
Honge	0.97	0.97	0.97	29	0.87	0.93	0.90	29
Insulin	0.96	0.96	0.96	24	0.96	0.96	0.96	24
Jasmine	0.93	0.93	0.93	27	0.74	0.96	0.84	27
Lemon	0.96	1.00	0.98	24	1.00	1.00	1.00	24
Lemon_grass	1.00	1.00	1.00	27	1.00	0.96	0.98	27
Mango	0.93	1.00	0.96	27	0.80	0.89	0.84	27
Mint	0.96	0.96	0.96	25	0.95	0.84	0.89	25
Nagadali	0.86	1.00	0.93	19	0.90	1.00	0.95	19
Neem	1.00	0.88	0.94	17	0.94	0.88	0.91	17
Nithyapushpa	1.00	1.00	1.00	23	0.95	0.91	0.93	23
Nooni	0.92	1.00	0.96	22	0.95	0.91	0.93	22
Pappaya	0.95	1.00	0.97	19	1.00	1.00	1.00	19
Pepper	1.00	0.97	0.98	31	0.93	0.87	0.90	31
Pomegranate	1.00	0.96	0.98	25	1.00	0.76	0.86	25
Raktachandini	1.00	1.00	1.00	16	0.94	1.00	0.97	16
Rose	1.00	1.00	1.00	24	0.82	0.96	0.88	24
Sapota	1.00	1.00	1.00	20	1.00	0.95	0.97	20
Tulasi	0.94	0.89	0.92	19	0.88	0.79	0.83	19
Wood_sorel	0.96	1.00	0.98	26	0.84	1.00	0.91	26
Accuracy			**0.97**	**893**			**0.90**	893
Macro Avg	**0.97**	**0.97**	**0.97**	**893**	**0.91**	**0.90**	**0.90**	893
Weighted Avg	**0.97**	**0.97**	**0.97**	**893**	**0.91**	**0.90**	**0.90**	893

4.5. Models hyperparameters

The various hyperparameters utilized for model development is given in Table 7. These values were obtained using the random search. This shows the comparison too for various hyperparameters requirement for the model's performance tuning properly.

4.6. RAG implementation and model deployment

Utilizing RAG with Langchain and the OpenAI API, a RAG chatbot is successfully implemented for the classification of medicinal plants utilizing this model. The system, utilizing a PDF document with details of all plant classes, their medicinal insights, use cases, and economical value, is utilized in the form of a document that the RAG chatbot retrieves. The system architecture is given in Fig. 13.

The application supports both English and Nepali, making it easier for users to transition between the two languages and improving accessibility. With the addition of GPT-4 for RAG implementation with help of a document with all the medicinal plants details and their medicinal insights, economic insights, the application gains a special capability in which it can identify illnesses and offer comprehensive explanations and suggested fixes in Nepali as seen in Fig. 11. This makes the program both user-friendly and instructive by guaranteeing that users receive actionable insights in a language they are comfortable with.

As, seen in Fig. 12, the app chatbot can be able to predict the class of the medicinal plant by scanning. This allows the users to be able to know what kind of medicinal plant it is and the insights on it can be gained. The program incorporates the OpenAI API to produce recommendations and fixes in Nepali. This guarantees that the suggestions are comprehensible and available to the nearby farmers, who might not speak English well. A variety of strategies, including the application of particular fungicides, crop rotation techniques, and organic treatment alternatives, are included in the therapies.

The application uses RAG to offer comprehensive information about this illness, including treatment options. By examining the user-scanned photos, the application's embedded TensorFlow Lite model can forecast a range of plant ailments. When a plant is identified and anticipated, the app makes use of this data to provide users with exact recommendations and profitable insights. The main chatbot screen UI and its operation are depicted in Figs. 13(a) and (b). Utilizing the Google Translation Library, the conversation can be continued or viewed in Nepali for supporting illiterate farmers through an application featuring local language, as seen in Fig. 13 (c). With the library, the country code for a language can be changed for different other language translations, so this app offers a solid feature in order to support a vast number of languages too.

The application offers a great platform for implementing a NLP chatbot for gaining insights and details on medicinal plants and, finally,

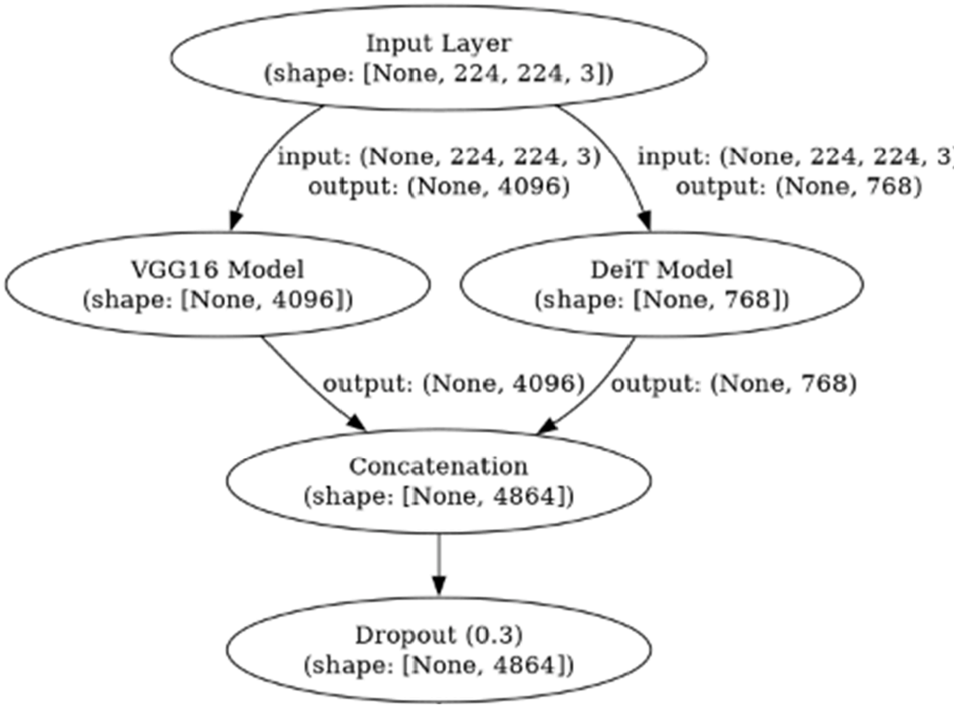

Fig. 10. DeiT + VGG 16 model architecture.

Table 6
Data for the stratified k-fold metrics.

Fold	Accuracy	Precision	Recall	F1 Score
Stratified Fold 1	0.9496	0.9525	0.9516	0.9508
Stratified Fold 2	0.9555	0.9572	0.9574	0.9552
Stratified Fold 3	0.9567	0.9582	0.9571	0.9564
Stratified Fold 4	0.9567	0.9577	0.9580	0.9570
Stratified Fold 5	0.9724	0.9715	0.9725	0.9712
Average	**0.9582**	**0.9594**	**0.9593**	**0.9581**

allowing users to know about their medicinal and economic benefits. This Chabot application not only offers the medicinal insights but also hands-on details of their economical profit-generating value in the market. By implementing a feature from the gpt-4 API token as well as the RAG retrieval document, the model from the backend makes predictions and gives the prediction of predicted medicinal plants to the RAG chatbot that finally generates various insights regarding the plants and their economical value, which can support the farmers and agriculturists livelihood.

There have been many works implemented utilizing vision transformers [16] and CNN-based approaches for medicinal plant image classification [1,2,4]. This work offers a solid boost to the identification models developed previously and also utilizes a transformer and hybrid deep learning method for classification. Mobile application for plant like aize disease detection [13] to various other agricultural applications are showcasing their applications in the agricultural development in many parts of world [6,14]. The proposed RAG chatbot language can be set to any kind of language utilizing country code as supported by the Google Translate library of Python. This can be a great approach for enhancing accessibility to knowledge and insight generation tools and enhancing the application of LLM in agricultural productions [5,10,19]. Comparing to previous work, this work presents a hybrid DeiT + VGG16 model with outstanding performance on cross-validation that shows the potential of the hybrid model in contributing to agricultural as well as ayurvedic

Table 7
Models utilized hyperparameters.

Parameter	DeiT	VGG16	DeiT + VGG16 Hybrid
Image Size	224 × 224	224 × 224	224 × 224
Batch Size	32	32	32
Learning Rate	1e-4	1e-4	1e-4
Optimizer	AdamW (final layer only)	AdamW (final layer only)	AdamW (final layer only)
Dropout Rate	0.3	0.3	0.3
Transformations	RandomResizedCrop, Horizontal Flip, RandomRotation	RandomResizedCrop, Horizontal Flip, RandomRotation	RandomResizedCrop, Horizontal Flip, RandomRotation
Normalization	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]
Scheduler	None	None	None
Epochs	150	150	150
Patience (Early Stopping)	7	7	7
Seed Value	42	42	42
Device	CUDA if available	CUDA if available	CUDA if available
Feature Extraction	Not applicable	Not applicable	VGG16 and DeiT features concatenated
Final Classifier	Not applicable	Not applicable	Fully connected layer with 4096 + 768 inputs and num_classes output

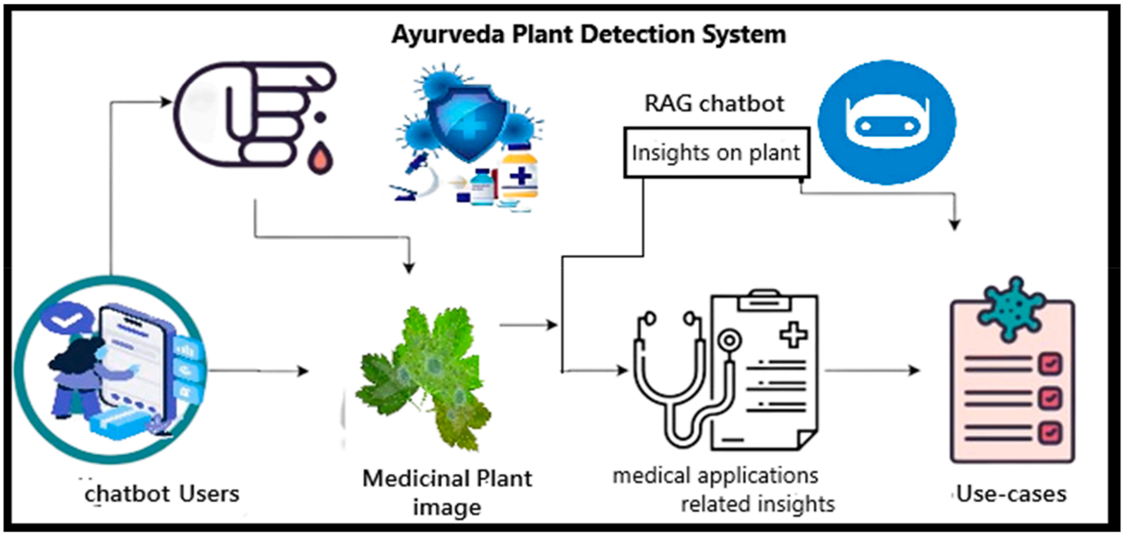

Fig. 11. Final system usage phenomenon.

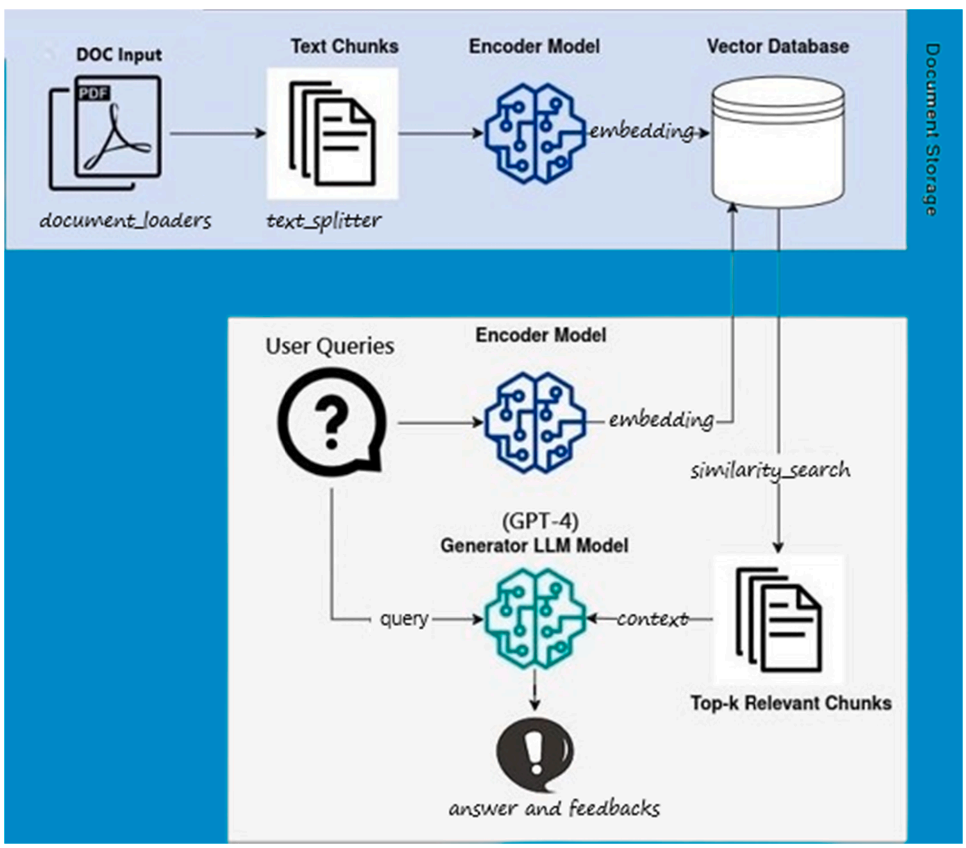

Fig. 12. Medicinal plant support RAG system.

medicinal plant identification. Finally, the proposed RAG application is a support for people involved in ayurvedic plant research and identification, including the small businesses depending upon such plants selling for livelihood.

4.7. Limitations and future works

The work involves the identification of medicinal plants by deep learning models for the purpose of clear identification of their usage as well as various insights in both Nepali and English. For this purpose, a RAG chatbot is developed by the authors. The work has several limitations too, as the model's performance isn't strong enough for identifying all medicinal plans, plus the RAG chatbot failed to generate insights on some plants and also sometimes provided misinformation regarding the plants and their usage and economic insights. Another limitation is the language conversion-related issues due to API-based dependency on GoogleTrans from Google, which wasn't heavily perfect.

In the future, we aim to enhance this work with a real-time mobile application, Chabot, with the features of identifying various medicinal plants and providing their insights and key parameters, and helping the

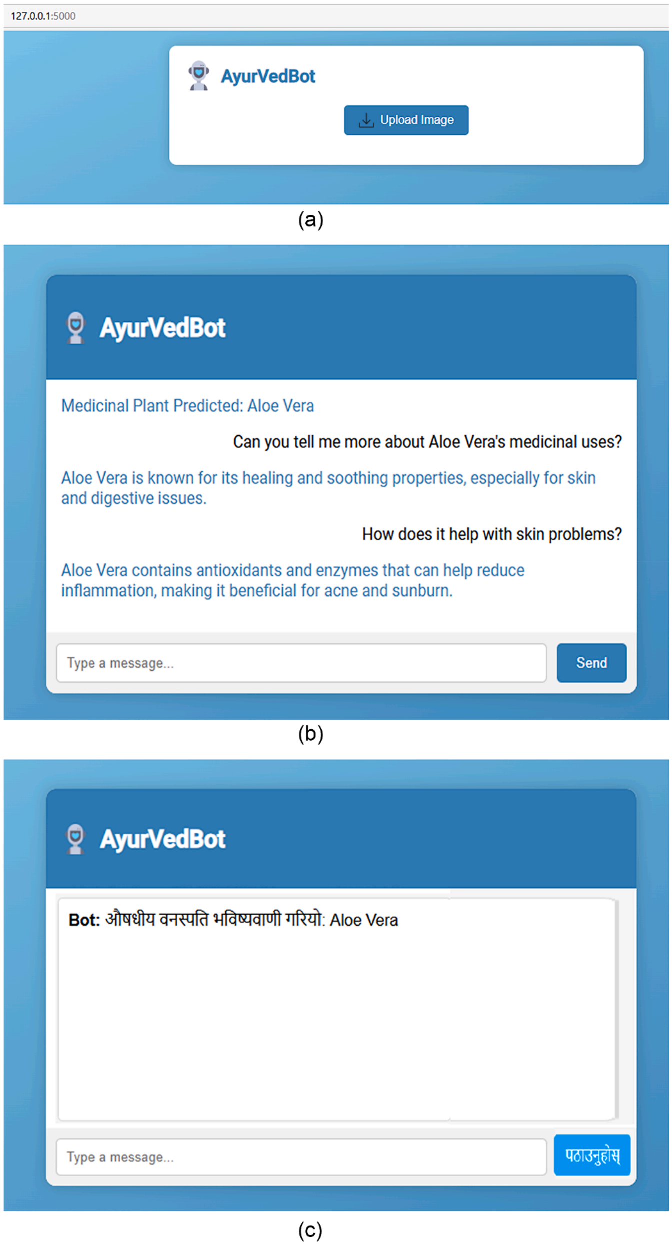

Fig. 13. (a). Chatbot image uploading section
(b). Chatbot application generating insights
(c). Nepali language-based chat on chatbot application.

users gain more medicinal knowledge. Further enhancement can be done with hardware IoT implementation for tracking plant location, identifying valuable medicinal plants, and generating detailed insights. The model performance and Chabot User Interface can be improved in the future with a lot more features and accessibility options. Moreover, we plan to include more medicinal plant species using advanced, resource-efficient models.

5. Conclusion

In this work, we developed an efficient ML-based solution for AI-based applications to assist farmers in many ways, such as identification of medicinal plants and providing their usage/significance with the help of a chat application. We leveraged the DeiT + VGG16 hybrid model to enhance the classification for efficient plant detection with a test accuracy of 96.75 %, a significant improvement to the VGG16 90.25 % accuracy. In addition to the offline classification, the app offers features such as saving predictions and switching Nepali and English languages. This handheld solution will be hand-held and remove dependence on any network connectivity, making users less dependent on infrastructure constraints, thus increasing their confidence and productivity. The app functions as an intelligent system, utilizing on-device machine learning to bring the power of ML directly to the public where such solutions have significant impact.

Declaration of use of LLM tools

The authors would like to declare that they used LLM tools for minimizing grammar errors and writing

CRediT authorship contribution statement

Biplov Paneru: Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bipul Thapa:** Methodology, Project administration, Resources, Software, Writing – review & editing. **Bishwash Paneru:** Writing – review & editing, Visualization, Validation, Software, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Authors would like to thank the Nepal Agricultural Research Council for providing useful insights on different plant diseases and current contextual issue related to crop diseases in Nepal.

Funding availability statement

The authors declare that no funding has been received for this research work

Data availability

Data will be made available on request.

References

[1] S. Kavitha, T.S. Kumar, E. Naresh, et al., Medicinal plant identification in real-time using deep learning model, SN Comput. Sci. 5 (2024) 73, https://doi.org/10.1007/s42979-023-02398-5.

[2] T.P. Tran, F.U. Din, L. Brankovic, C. Sanin, S.M. Hester, et al., A systematic review of medicinal plant identification using deep learning, in: N.T. Nguyen, et al. (Eds.), Intelligent Information and Database Systems, ACIIDS 2024, Lecture Notes in Computer Science, Springer, Singapore, 2024, https://doi.org/10.1007/978-981-97-4985-0_1 vol. 14796.

[3] R. Sapna, S.N. Sheshappa, An extensive study on machine learning paradigms towards medicinal plant classification on potential of medicinal properties, in: J.I. Z. Chen, J.M.R.S. Tavares, F. Shi (Eds.), Third International Conference On Image Processing and Capsule Networks, ICIPCN 2022, Lecture Notes in Networks and Systems, Springer, Cham, 2022, https://doi.org/10.1007/978-3-031-12413-6_43 vol. 514.

[4] K.L.D. Viet, K.L. Ha, T.N. Quoc, V.T. Hoang, Medicinal plants identification using federated deep learning, Procedia Comput. Sci. 234 (2024) 247–254, https://doi.org/10.1016/j.procs.2024.02.171.

[5] R. Azadnia, F. Noei-Khodabadi, A. Moloudzadeh, A. Jahanbakhshi, M. Omid, Medicinal and poisonous plants classification from visual characteristics of leaves using computer vision and deep neural networks, Ecol. Informatics 82 (2024) 102683, https://doi.org/10.1016/j.ecoinf.2024.102683.

[6] A. Siddiqua, M.A. Kabir, T. Ferdous, I.B. Ali, L.A. Weston, Evaluating plant disease detection mobile applications: Quality and limitations, Agronomy 12 (2022) 1869, https://doi.org/10.3390/agronomy12081869.

[7] V. Bischoff, K. Farias, J.P. Menzen, G. Pessin, Technological support for detection and prediction of plant diseases: A systematic mapping study, Comput. Electron. Agric. 181 (2021) 105922, https://doi.org/10.1016/j.compag.2020.105922.

[8] S.S. Harakannanavar, J.M. Rudagi, V.I. Puranikmath, A. Siddiqua, R. Pramodhini, Plant leaf disease detection using computer vision and machine learning algorithms, Glob. Trans. Proc 3 (2022) 305–310, https://doi.org/10.1016/j.gltp.2022.03.016.

[9] M.M. Islam, M.A.A. Adil, M.A. Talukder, M.K.U. Ahamed, M.A. Uddin, M.K. Hasan, S. Sharmin, M.M. Rahman, S.K. Debnath, DeepCrop: Deep learning-based crop disease prediction with web application, J. Agric. Food Res. 14 (2023) 100764, https://doi.org/10.1016/j.jafr.2023.100764.

[10] S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016), https://doi.org/10.3389/fpls.2016.01419.

[11] M.H. Saleem, S. Khanchi, J. Potgieter, K.M. Arif, Image-based plant disease identification by deep learning meta-architectures, Plants 9 (2020) 1451, https://doi.org/10.3390/plants9111451.

[12] W.F. Hulkury, L. Nagowah, Plant disease prediction using deep learning techniques, in: M.E. Auer, T. Tsiatsos (Eds.), Smart Mobile Communication & Artificial Intelligence, IMCL 2023, Lecture Notes in Networks and Systems, Springer, Cham, 2024, https://doi.org/10.1007/978-3-031-56075-0_24 vol. 937.

[13] F. Khan, N. Zafar, M.N. Tahir, M. Aqib, H. Waheed, Z. Haroon, A mobile-based system for maize plant leaf disease detection and classification using deep learning, Front. Plant Sci. 14 (2023), https://doi.org/10.3389/fpls.2023.1079366.

[14] R.S. Babatunde, A.N. Babatunde, R.O. Ogundokun, et al., A novel smartphone application for early detection of habanero disease, Sci. Rep. 14 (2024) 1423, https://doi.org/10.1038/s41598-024-52038-y.

[15] O.N. Manzari, H. Ahmadabadi, H. Kashiani, S.B. Shokouhi, A. Ayatollahi, MedViT: A robust vision transformer for generalized medical image classification, Comput. Biol. Med. 157 (2023) 106791, https://doi.org/10.1016/j.compbiomed.2023.106791.

[16] D.T.N. Nhut, T.D. Tan, T.N. Quoc, V.T. Hoang, Medicinal plant recognition based on Vision Transformer and BEiT, Procedia Computer Science 234 (2024) 188–195, https://doi.org/10.1016/j.procs.2024.02.165.

[17] BR Pushpa, Shobha Rani, Indian Medicinal Leaves Image Datasets, Mendeley Data 3 (2023), https://doi.org/10.17632/748f8jkphb.3.

[18] Teng, Chongyangzi & Yang, Pengwei & Guo, Mengshen. (2023). Multimodal in Multi-Label Classification: A Report. 10.13140/RG.2.2.29898.54722.

[19] L. Shuangfeng, Lite TensorFlow, On-device machine learning framework, J. Comput. Res. Dev. 57 (2020) 1839, https://doi.org/10.7544/ISSN1000-1239.2020.20200291.