The Potential of Transfer Learning: Fine-Tuning Pre-Trained Models for Medicinal Herb Identification

Shreela Pareek
Department of Computer Science,
KIET Group of Institutions,
Delhi-NCR, Ghaziabad,
India
shreela.pareek@kiet.edu

Mohammad Ausaf
Department of Computer Science,
KIET Group of Institutions,
Delhi-NCR, Ghaziabad,
India
mohammad.2024cs1056@kiet.edu

Harsh Agrahari
Department of Computer Science,
KIET Group of Institutions,
Delhi-NCR, Ghaziabad,
India
harsh.2024cs1106@kiet.edu

Rahul Yadav
Department of Computer Science,
KIET Group of Institutions,
Delhi-NCR, Ghaziabad,
India
rahul.2024cs1166@kiet.edu

***Abstract*— In this research paper, we study the domain of medicinal plant identification, focusing on the application of machine learning and deep learning techniques. With the rich diversity of such materials in the world, accurate identification is essential, as misidentification can lead to confusion, adulteration, and a lack of trust in the traditional Ayurvedic or medical system(s). Our primary objective is to explore and harness the potential of transfer learning as an effective solution to this long-standing challenge. Transfer learning leverages knowledge acquired by a model in one task to enhance its performance in another, closely related task. To accomplish this, we have extensively examined the capabilities of renowned deep learning models such as ResNet-50 and Vgg16.**

Our study delves into the fundamental principles of transfer learning, considering its impact on the accuracy and reliability of medicinal plant classification. By adopting transfer learning, we utilize pre-trained models, specifically the ResNet-50 and Vgg16 architectures, which are well-recognized for their image-recognition abilities. These models have already been trained on vast and diverse datasets, enabling them to capture and extract invaluable features from images, a crucial aspect of plant classification which is then leveraged while customizing these models for our particular use case. We used transfer learning, combining both fine-tuning and feature-extraction techniques to achieve accuracy of 98%.

I. Introduction

Medicinal plant leaves have long been utilized in traditional medicine. Identification of medicinal leave is a very challenging task. This task is often fraught with difficulties when undertaken without external resources or assistance. Accurate identification of the specific medicinal plants employed in the creation of herbal remedies is a critical aspect of the medicinal industry. The utilization of traditional plant-based medicines alongside pharmaceutical drugs is a growing trend in many countries, signifying a wealth of opportunities in this field. To tackle the challenge of medicinal plant leaf identification, our research explores major image recognition algorithms of Image analysis, a fundamental method in this context, plays a pivotal role. The integration of image analysis, deep learning, and AI techniques enhances the precision and reliability of medicinal leaf identification, thereby contributing to the broader landscape of traditional medicine and herbal remedy preparation. Through this research, we aim to shed light on the potential and significance of these technological advancements in the realm of traditional medicine.

II. Literature Review

In the field of plant species identification based on leaves, a substantial body of research has been conducted. However, it is important to note that the majority of these studies have focused on general plant identification, rather than the specific identification of medicinal plants based on their leaves. This presents a significant gap in the existing literature, as the accurate identification of medicinal plants is important in fields such as healthcare and pharmacology. In this literature review, we will explore the works related to plant species identification and medicinal plant identification and introduce our research, which aims to address this notable gap.

A. Morphology-based classification

Thomkaew et al. [**1**] This study presents a novel approach to plant classification utilizing leaf edge features combined with Morphological Transformations and Scale-Invariant Feature Transform (SIFT). The methodology involves three key steps: image preprocessing, feature extraction employing SIFT, and image classification using random forest. Experiments conducted on the PlantVillage dataset, which includes 10 classes, yielded an accuracy of 95.62%. Xu et al. [**2**] Introduced a novel technique utilizing the inner distance shape context (IDSC) descriptor, which resulted in enhancements in shape-based recognition. Mahajan et al. [**3**] method emphasized morphological attributes, specifically focusing on the center of mass, which led to an impressive recognition accuracy of 95.40% on the Flavia dataset through the utilization of adaptive boosting. Yang et al. [**4**] introduced the triangular distance representation (TDR) for a more comprehensive description of plant leaf shapes, showing superior results compared to conventional shape-based methods. Wu et al. [**5**] achieved a commendable 90% recognition rate using a probabilistic neural network, leveraging features like aspect ratio. Hu et al. [**6**] surpassed the IDSC descriptor's performance by proposing a fast tree-leaf recognition model based on a multi-scale distance matrix.

B. Texture-based Classification

Several methods have been applied to plant leaf classification, yielding impressive results. Zhang et al. [**7**] method attains retrieval accuracies of 77.6%, 85.7%, and 67.5% on the Flavia, Swedish, and MEW2012 leaf datasets, respectively. The corresponding classification accuracies are 99.1%, 98.4%, and 95.6%. Turkoglu et al [**8**] proposed Region Mean- LBP and Overall Mean-LBP methods, attaining recognition accuracies of 98.94% and 99.46%. Chaki et al. combined Gabor filters and co-occurrence matrices, resulting in a re-markable 97.6% accuracy. Naresh et al. [**9**] employed MLBP for texture, achieving accuracies of 97.55% and 96.83%. Muthevi et al. [**10**] utilized various CLBP methods, reaching 84.78% and 88.89% accuracy. Herdiyeni et al. [**11**] used FLBP for texture and FCH for colour, obtaining a recognition accuracy of 74.51%. Le et al. [**12**] developed a plant classification system with a combination of LBP and SVM, achieving a classification accuracy of 91.85% on the bccr-segset dataset.

C. Multivariate classification approach

Researchers have explored various methods for plant leaf identification. Turkoglu et al. divided plant leaves into blocks and achieved 99.10% accuracy using the ELM classifier. Lv et al. [**13**] introducing a multi-feature fusion technique for plant leaf recognition employing an enhanced LBP descriptor, HOG, and color features. Attained 99.30% accuracy on the Flavia dataset and 99.52% on the Swedish dataset. Goyal et al. [**15**] introduced a dual support vector machine classification method with 98.11% accuracy. Su et al. [**16**] investigated multiscale triangle representations, reaching 99.35% and 99.43% accuracy on Swedish and Flavian datasets. Aakif et al. [**17**] developed a three-step plant identification algorithm, obtaining

96.25% accuracy through shape, texture, and color features. Saleem et al. [**18**] proposed a five-step algorithm with 97.6% recognition accuracy. Herdiyeni et al. [**19**] proposed a system incorporating texture, shape, and color features for medicinal plant identification. Goyal et al. [**20**] employed a 183-dimensional feature vector and achieved competitive results.

D. Transfer-Learning Approaches

Transfer Learning (TL) is a ML technique in which a data-driven model previously trained (general training) for a given task (source domain) is used as the base to build a model for a new task (target domain), with less data being required for the new training stage (task-specific training) [**20**]. Simply put, Transfer Learning (TL) is a technique in machine learning where we take a model that has learned a lot from one task and use that knowledge as a starting point to learn something new, related but slightly different. It's like leveraging your expertise in one field to help you become skilled in another field more quickly, saving time and resources. There are relatively less work done on leveraging transfer learning models for medicinal plant identification, we came across the following in the literature.

Sachar et al. [**21**] developed and evaluated three distinct models for this purpose. They used an ensemble approach known as stacking classifiers, where predictions from multiple classifiers were used to train a meta-classifier. This approach yielded remarkable results, achieving an accuracy of 99.16% on the unseen samples of the Swedish dataset and 98.13% on the Flavia dataset. Ali et al. [**22**] proposed a logistic regression classifier for leaf classification, and used the model for evaluating limited datasets in plant classification. The results were on two public datasets Flavia with 32 classes and leaf- snaps with 184 classes, achieving an accuracy of 99.6% and 90.54%, respectively. Ananda et al. [**23**] used data augmentation to increase leaf disease dataset volume and achieved better model performance, also improving performance metrics by setting hyper-parameters tuning with VGG model achieving CNN multilayer accuracy of 98.40% on grapes and 95.71% on tomatoes.

III. Methodology

We have employed the following methodology in our process:

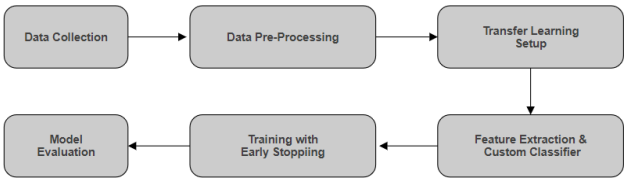

Fig. 1. Flow of the Process

we will repeat steps from Transfer Learning Setup up to Model Evaluation for each of our Models while the Data Collection and Data Pre-processing steps will remain constant.

A. Data Collection

We utilized publicly available data published on Mendeley, the "Medicinal Leaf Dataset" created by S. Roopashree and J. Anitha. [23].

This dataset encompasses 1,500 high-quality images of thirty diverse species of healthy medicinal herbs, including prominent plants like Santalum album and Muntingia calabura.

Each species is well-represented with an average of 60 to 100 images, capturing leaves in various orientations to enhance diversity. Out of these 1,500 images, we have used 70% for Training, 15% for validation and rest of 15% for Testing.

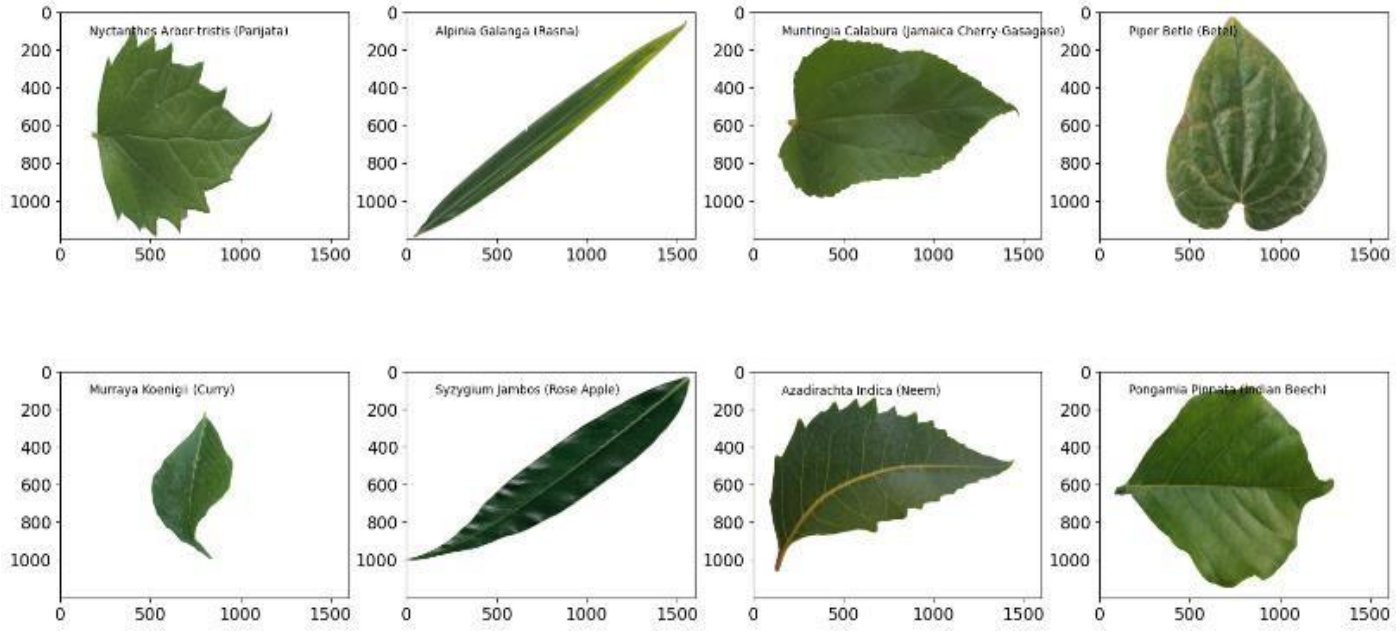

Fig 2. Sample of Original Dataset

B. Data Preprocessing

We have applied a sequence of data preprocessing steps, viz. Data Augmentation and Data Transform on Images.

a. Data Transforms and Data Augmentation

We employ data augmentation and transformation techniques to enhance the diversity, quantity and quality of our image dataset. Data augmentation is a crucial step in deep learning, as it introduces variations to the training data, making the model more robust and preventing overfitting. Data Augmentation is applied to train and validation datasets.

The following transform and augmentation techniques are used:

Technique	Description/Values
Random Resized Crop	256 X 256
Random Rotation	15 degrees
Color Jitter	Colour adjustments
Random Horizontal Flip	Flipped Horizontally

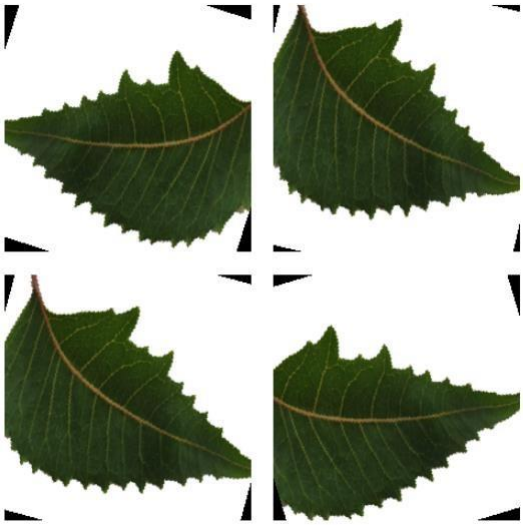

Fig. 3 Sample of Augmented Dataset

b. Data Standardization and Normalization

After these transforms, we perform a Center Crop to standardize the image size to 224x224 pixels. This size is a common standard for deep learning models trained on the ImageNet dataset.

Furthermore, we apply data normalization to ensure our images match the mean and standard deviation of the ImageNet dataset. The normalization values used are [0.485, 0.456, 0.406] for mean and [0.229, 0.224, 0.225] for standard deviation. This step is crucial as many pre-trained models, such as ResNet, Vgg16 and AlexNet, were trained on ImageNet data. Data normalization helps our model converge efficiently during training.

C. Transfer Learning Set-Up

The dataset is organized into training, validation, and testing sets using the defined transformations. Data loaders are created for each dataset. Data loaders help in efficiently loading and iterating over batches of data during training. We used a batch size of 512 while training the model.

After Defining data loaders, we load pre-trained ResNet50 and Vgg16 models. This model has already been trained on a large dataset (e.g., ImageNet) and has learned useful features for image classification.

D. Feature Extraction and Custom Classifier

After Loading our pre-trained models, we freeze the parameters of the model, essentially resulting in the retention of the default weights and knowledge.

The classification layers (the final fully connected layer in ResNet and 3rd Convolutional layer in Vgg16) of the model, which are responsible for mapping features to class labels, are replaced with a custom classification layer. The new classification layer is specific to the number of classes in your dataset. Each output unit in this layer corresponds to a particular class. This custom layer allows the model to adapt to our specific image classification task.

Following are the custom classifier layers replacing default classifier layers:

```
Sequential(
  (0): Linear(in_features=2048, out_features=256, bias=True)
  (1): ReLU()
  (2): Dropout(p=0.4, inplace=False)
  (3): Linear(in_features=256, out_features=30, bias=True)
  (4): LogSoftmax(dim=1)
)
```

Fig. 2 . Custom Classifier

1. Fully Connected Layer:
   - Input Features: 2048
   - Output Features: 256
   - Activation Function: None
   - Parameters: 2048 x 256 (Weights), 256 (Biases)

2. Rectified Linear Unit (ReLU):
   - Activation Function: ReLU
   - Applies element-wise non-linearity

3. Dropout Layer:
   - Probability: 0.4
   - Regularization Technique: Dropout
   - Reduces overfitting by randomly setting input units to zero during training

4. Fully Connected Layer (Output Layer):
   - Input Features: 256
   - Output Features: 30
   - Activation Function: None
   - Parameters: 256 x 30 (Weights), 30 (Biases)

5. Log SoftMax Activation:
- Activation Function: Log SoftMax
- Converts output scores to log probabilities for multiclass classification.

E. Model Training with Early Stopping

In this study, the deep learning models VGG16 and ResNet50 were trained using a 15GB NVIDIA Tesla K80 GPU. To develop algorithms and implement the models, TensorFlow version 2.14.0 with the Keras API was employed as the deep learning framework.

Following Negative Log Likelihood Loss Function and Adam Optimizer were used while training the model :

$$\text{NLLLoss}(\theta) = -\sum_{i=1} log(p_{yi})$$

Where:

Θ represents the model's parameters.
Yi represents the true class labels.
P_{yi} is the predicted probability of the true class.

Early Stopping was also as a regularization technique used during training to prevent overfitting. It involves monitoring the model's performance on a validation dataset and halting training when the performance begins to degrade.

IV. Results

With this setup, we achieved significant results on both models. For ResNet50, we achieved a training accuracy of 99.0% and validation accuracy of 96.46% training over 52 epochs. While for Vgg16, the results were a training accuracy of 82.29% and a validation accuracy of 90.46%. Following is a detailed discussion of results of each model and a comparison of the results.

a. Result for ResNet50

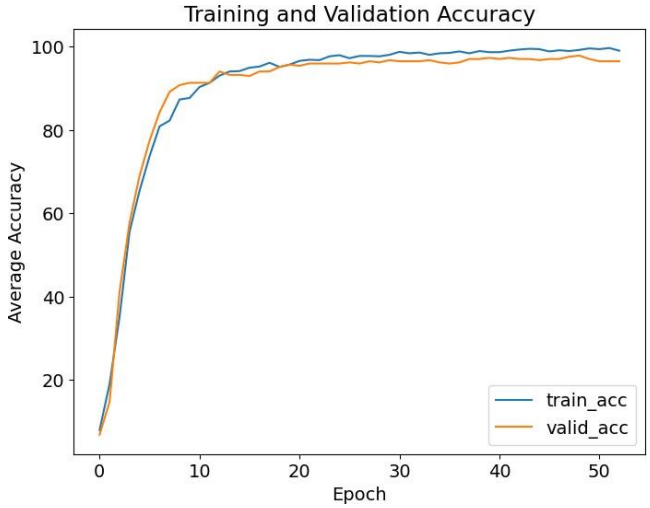

Fig. 4 Training vs Validation Accuracy of ResNet50

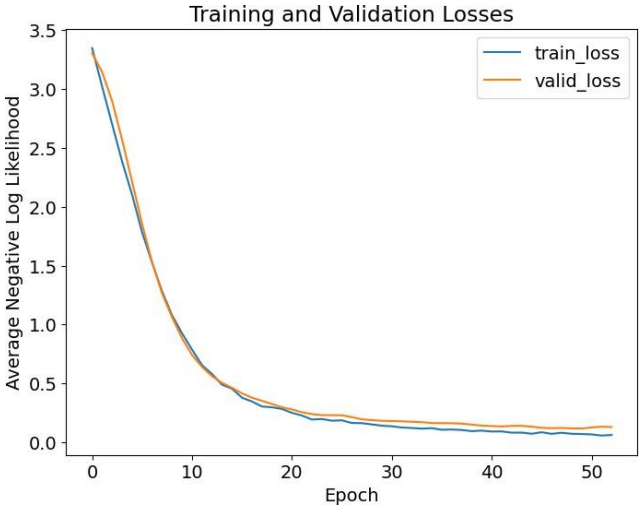

Fig. 5 Training vs Validation Loss for ResNet50

For the final Epoch, the model achieved a validation loss of 0.12. A lower loss indicates that the model was making predictions that were closer to the true values. This suggests that the model's predictions were quite accurate at this point.

At the same epoch, the model had an accuracy of 96.46%. This means that nearly 96.46% of the predictions matched the true labels in the validation dataset, indicating a high level of accuracy.

Top-1 Metrics:

1. *Top-1 Accuracy (98.37%):* The high top-1 accuracy indicates that the model correctly predicts the most likely class label for the majority of the samples. This tells us that nearly 98.37% of the predictions match the true class labels suggesting that our model performs very well in making the single most probable prediction.

2. *Precision (72.20%):* The precision is slightly lower 72.20%. This tells us that when the model predicts a class label as positive (e.g., class A), it is accurate approximately 72.20% of the time.

3. *Recall (0.073 or 7.3%):* The low recall value indicates that our model misses a significant number of positive samples. In this case, only 7.3% of the actual positive samples are correctly identified by the model. This suggests us that there's room for improvement in the model's ability to capture all positive instances.

4. *F1 Score (0.145):* The F1 score is a harmonic mean of precision and recall. In our case, the low F1 score of 0.145 indicates that there is an imbalance between precision and recall. The model's predictions are more skewed toward precision, but it comes at the cost of lower recall.

Class	top1	top3	top5	loss	Train Count	Validation Count	Test Count
Alpinia Galanga (Rasna)	100.000	100.000	100.000	0.135	30	10	10
Amaranthus Viridis (Arive-Dantu)	68.000	92.000	100.000	0.789	73	24	25
Artocarpus Heterophyllus (Jackfruit)	100.000	100.000	100.000	0.188	34	11	11
Azadirachta Indica (Neem)	75.000	91.667	100.000	0.838	36	12	12
Basella Alba (Basale)	95.238	100.000	100.000	0.334	62	20	21
Brassica Juncea (Indian Mustard)	100.000	100.000	100.000	0.132	20	7	7
Carissa Carandas (Karanda)	100.000	100.000	100.000	0.227	44	15	15
Citrus Limon (Lemon)	63.636	72.727	90.909	1.357	34	12	11
Ficus Auriculata (Roxburgh fig)	20.000	70.000	80.000	1.903	30	10	10
Ficus Religiosa (Peepal Tree)	100.000	100.000	100.000	0.122	38	12	13
Hibiscus Rosa-sinensis	50.000	100.000	100.000	0.802	26	9	8
Jasminum (Jasmine)	92.857	100.000	100.000	0.334	43	14	14
Mangifera Indica (Mango)	100.000	100.000	100.000	0.104	37	12	13
Mentha (Mint)	85.000	95.000	100.000	0.665	58	19	20
Moringa Oleifera (Drumstick)	93.333	100.000	100.000	0.234	46	16	15
Muntingia Calabura (Jamaica Cherry-Gasagase)	100.000	100.000	100.000	0.233	33	12	11
Murraya Koenigii (Curry)	75.000	83.333	100.000	0.978	36	12	12
Nerium Oleander (Oleander)	100.000	100.000	100.000	0.029	37	12	13
Nyctanthes Arbor-tristis (Parijata)	100.000	100.000	100.000	0.082	24	8	8
Ocimum Tenuiflorum (Tulsi)	70.000	90.000	100.000	0.799	31	11	10
Piper Betle (Betel)	90.000	100.000	100.000	0.704	29	9	10
Plectranthus Amboinicus (Mexican Mint)	88.889	100.000	100.000	0.436	29	10	9
Pongamia Pinnata (Indian Beech)	100.000	100.000	100.000	0.156	37	12	12
Psidium Guajava (Guava)	100.000	100.000	100.000	0.094	39	13	13
Punica Granatum (Pomegranate)	93.750	100.000	100.000	0.275	47	16	16
Santalum Album (Sandalwood)	81.818	90.909	100.000	0.883	35	12	11
Syzygium Cumini (Jamun)	100.000	100.000	100.000	0.071	23	8	8
Syzygium Jambos (Rose Apple)	100.000	100.000	100.000	0.057	34	11	11
Tabernaemontana Divaricata (Crape Jasmine)	90.909	100.000	100.000	0.304	34	11	11
Trigonella Foenum-graecum (Fenugreek)	57.143	85.714	85.714	1.522	22	7	7

Figure 5. Evaluation Metrics across Classes

Top-5 Metrics:

1. *Top-5 Accuracy (100%):* The top-5 accuracy is high, indicating that for each sample, this tells us that the true label is within the top 5 predicted labels meaning that our model, considering a broader range of possible predictions, correctly identifies the true class label.

2. *Precision (73.40%):* The precision for the top-5 predictions is relatively high at 73.40%. It tells us that when the model predicts a positive label within the top 5 predictions, it gets it right approximately 73.40% of the time.

3. *Recall (0.074 or 7.4%):* Like the top-1 recall, the recall for the top-5 predictions is relatively low, indicating that only 7.4% of the actual true class predictions are among the top-5 predictions.

4. *F1 Score (0.147):* The F1 score for the top-5 predictions remains relatively low, indicating an imbalance between precision and recall, similar to the top-1 context.

Overall, the above metrics suggest that while the model performs exceptionally well in terms of accuracy and precision, it has a significant challenge in terms of recall. Our model tends to make conservative predictions, which results in missing a portion of positive samples. Improving recall while maintaining high precision could be a key objective for enhancing the model's performance in future work. This may involve adjusting the model's threshold or exploring other techniques to increase its ability to identify true positives, which can be undertaken for future work.

b. Results for Vgg16

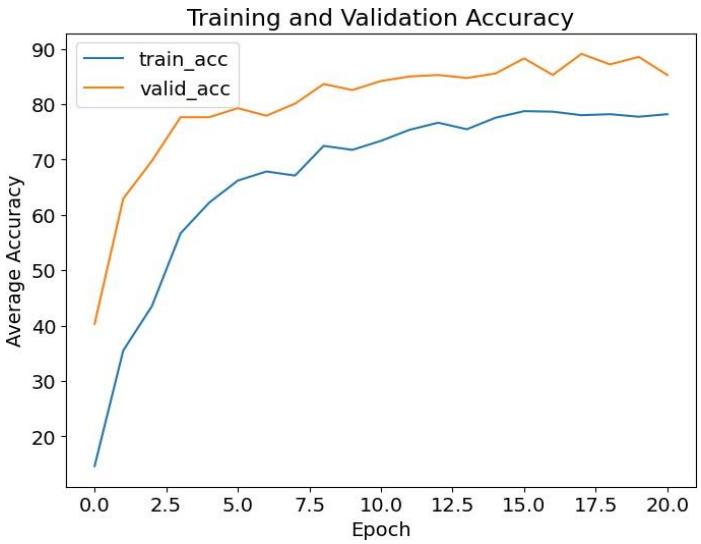

Fig 6. Training Loss vs Validation Loss

For the last Epoch, the model achieved a validation loss of 0.40, this lower loss indicates that the model was making predictions that were closer to the true values. This suggests that the model's predictions were quite accurate at this point.

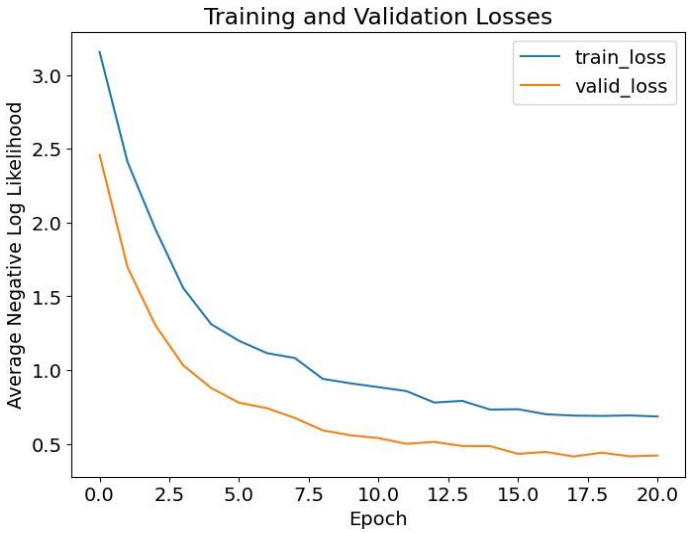

Fig 7. Training vs Validation Accuracy

At the same epoch, the model had a validation accuracy of 86.38%. This means that nearly 86.38% of the predictions matched the true labels in the validation dataset, indicating a relatively lower level of accuracy with respect to ResNet50.

Top-1 Metrics:

1. *Top-1 Accuracy (89.84%):* The top-1 accuracy is at 89.84%, indicating that the model correctly predicts the most likely class label for approximately 89.84% of the time. This suggests a reasonable level of performance in making the single most probable prediction.

2. *Precision (23.0%):* The precision value is relatively low at 23.0%. This means that when the model predicts a class label as positive (e.g., class A), it is accurate only 23.0% of the time.

3. *Recall (0.067 or 6.7%):* The low recall value (6.7%) indicates that the model misses a significant number of positive samples. In this case, only 6.7% of the actual positive samples are correctly identified by the model, suggesting that there's room for improvement in the model's ability to capture positive instances.

4. *F1 Score (0.134):* The F1 score is a harmonic means of precision and recall. In our case, the F1 score is relatively low at 0.134, which indicates that there is an imbalance between precision and recall. The model's predictions lean more towards precision, but this comes at the cost of lower recall.

Class	top1	top3	top5	loss	Train Count	Validation Count	Test Count
Alpinia Galanga (Rasna)	100	100	100	0.07	30	10	10
Amaranthus Viridis (Arive-Dantu)	100	100	100	0.122	73	25	24
Artocarpus Heterophyllus (Jackfruit)	100	100	100	0.148	34	11	11
Azadirachta Indica (Neem)	100	100	100	0.055	36	12	12
Basella Alba (Basale)	100	100	100	0.062	62	20	21
Brassica Juncea (Indian Mustard)	85.714	100	100	0.323	20	7	7
Carissa Carandas (Karanda)	93.333	100	100	0.173	44	15	15
Citrus Limon (Lemon)	91.667	100	100	0.342	34	11	12
Ficus Auriculata (Roxburgh fig)	90	100	100	0.209	30	10	10
Ficus Religiosa (Peepal Tree)	100	100	100	0.102	38	12	13
Hibiscus Rosa-sinensis	100	100	100	0.089	26	8	9
Jasminum (Jasmine)	100	100	100	0.061	43	14	14
Mangifera Indica (Mango)	69.231	100	100	0.462	37	12	13
Mentha (Mint)	100	100	100	0.018	58	20	19
Moringa Oleifera (Drumstick)	100	100	100	0.087	46	16	15
Muntingia Calabura (Jamaica Cherry-Gasagase)	100	100	100	0.094	33	12	11
Murraya Koenigii (Curry)	100	100	100	0.181	36	12	12
Nerium Oleander (Oleander)	100	100	100	0.036	37	12	13
Nyctanthes Arbor-tristis (Parijata)	100	100	100	0.049	24	8	8
Ocimum Tenuiflorum (Tulsi)	100	100	100	0.169	31	11	10
Piper Betle (Betel)	100	100	100	0.316	29	9	10
Plectranthus Amboinicus (Mexican Mint)	100	100	100	0.201	29	10	9
Pongamia Pinnata (Indian Beech)	100	100	100	0.064	37	12	12
Psidium Guajava (Guava)	100	100	100	0.017	39	13	13
Punica Granatum (Pomegranate)	100	100	100	0.161	47	16	16
Santalum Album (Sandalwood)	90.909	100	100	0.192	35	12	11
Syzygium Cumini (Jamun)	100	100	100	0.068	23	8	8
Syzygium Jambos (Rose Apple)	100	100	100	0.017	34	11	11
Tabernaemontana Divaricata (Crape Jasmine)	81.818	90.909	100	0.844	34	11	11
Trigonella Foenum-graecum (Fenugreek)	100	100	100	0.102	22	7	7

Figure 8. Evaluation Metrics across all classes

Top-5 Metrics:

1. *Top-5 Accuracy (100%):* The top-5 accuracy is high, meaning that for each sample, the true label is within the top 5 predicted labels. This suggests that when considering a broader range of possible predictions, the model can identify the true class labels.

2. *Precision (25.6%):* The precision for the top-5 predictions is relatively low at 25.6%. It indicates that when the model predicts a positive label within the top 5 predictions, it is accurate only 25.6% of the time.

3. *Recall (0.075 or 7.5%):* Similar to the top-1 recall, the recall for the top-5 predictions is low, indicating that only 7.5% of the actual positive samples are among the top 5 predictions.

4. *F1 Score (0.149):* The F1 score for the top-5 predictions remains relatively low, suggesting an imbalance between precision and recall, similar to the top-1 context.

In summary, these metrics indicate that the model's top-1 and top-5 accuracies are relatively high, but there are issues with precision and recall. The model tends to make conservative predictions, which results in low recall.

V. Conclusion and Scope of Future Work

Our comparative analysis of the ResNet50 and VGG16 models, considering both top-1 and top-5 performance metrics reveals differences in their performance. Both models exhibit a high top-1 accuracy, achieving approximately 98.37% and 86.38%, indicating their proficiency in correctly predicting the most likely class label for the majority of samples. However, ResNet50 emerges with a more balanced approach, achieving a higher precision of 72.20% compared to VGG16 of 23.0% in top-1 predictions. While both models struggle with low recall—ResNet50 at 7.4% and VGG16 at 6.7%—implying their inability to capture a proportion of positive samples, ResNet50 excels in this regard. The F1 scores also indicate these imbalances, yet ResNet50 performs better.

When we look at top-5 predictions, both models excel with perfect top-5 accuracy, suggesting their ability to identify the true class label or one of the top 5 predicted labels. However, ResNet50 shows stronger precision (73.40%) compared to VGG16 (25.6%), implying greater accuracy when predicting positive labels within the top 5. Despite this, both models face challenges with low recall in top-5 predictions (ResNet50 at 7.4% and VGG16 at 7.5%), suggesting that they struggle to capture positive instances within the top-5 predictions, thereby impacting the overall balance of their F1 scores.

In conclusion, our study reveals the ResNet50 model as more balanced in terms of precision and recall, particularly in top-1 predictions, while both models prioritize precision over recall. Future works should be directed towards enhancing recall while preserving high precision, potentially through parameter fine-tuning or novel techniques, data augmentation and potentially using Ensemble Model Development, to further optimize the practical utility of these models in a range of applications.

REFERENCES

[1] Thomkaew, Jiraporn, and Sarun Intakosum. "Plant Species Classification Using Leaf Edge [J]." *Journal of Image and Graphics* 11.1 (2023).

[2] Xu, Guoqing. "Effective scale-generation scheme-based shape descriptor for fast leaf recognition and retrieval." *The Journal of Engineering* 2023.2 (2023): e212241.

[3] Mahajan, S.; Raina, A.; Gao, X.-Z.; Kant Pandit, A. Plant Recognition Using Morphological Feature Extraction and Transfer Learning over SVM and AdaBoost. Symmetry 2021, 13, 356.

[4] Yang, Chengzhuan, and Hui Wei. "Plant species recognition using triangle-distance representation." *IEEE Access* 7 (2019): 178108-178120.

[5] Wu, Stephen Gang, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. "A leaf recognition algorithm for plant classification using probabilistic neural network." In *2007 IEEE international symposium on signal processing and information technology*, pp. 11-16. IEEE, 2007.

[6] Hu, Rongxiang, Wei Jia, Haibin Ling, and Deshuang Huang. "Multiscale distance matrix for fast plant leaf recognition." *IEEE transactions on image processing* 21, no. 11 (2012): 4667-4672.

[7] Yang, Chengzhuan. "Plant leaf recognition by integrating shape and texture features." *Pattern Recognition* 112 (2021): 107809.

[8] Turkoglu, Muammer, and Davut Hanbay. "Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine." *Physica A: Statistical Mechanics and its Applications* 527 (2019): 121297.

[9] Naresh, Y. G., and H. S. Nagendraswamy. "Classification of medicinal plants: an approach using modified LBP with symbolic representation." *Neurocomputing* 173 (2016): 1789-1797.

[10] Muthevi, Anilkumar, and Ravi Babu Uppu. "Leaf classification using completed local binary pattern of textures." In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pp. 870-874. IEEE, 2017.

[11] Herdiyeni, Yeni, and Mayanda Mega Santoni. "Combination of morphological, local binary pattern variance and color moments features for indonesian medicinal plants identification." In *2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 255-259. IEEE, 2012.

[12] Le, Vi Nguyen Thanh, Beniamin Apopei, and Kamal Alameh. "Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods." *Information processing in agriculture* 6, no. 1 (2019): 116-131.

[13] Lv, Zhimin, and Zhibin Zhang. "Research on plant leaf recognition method based on multi-feature fusion in different partition blocks." *Digital Signal Processing* 134 (2023): 103907.

[14] Kadir, Abdul, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. "Foliage plant retrieval using polar Fourier transform, color moments and vein features." *arXiv preprint arXiv:1110.1513* (2011).

[15] Goyal, Neha, Nitin Kumar, and Kapil. "On solving leaf classification using linear regression." *Multimedia Tools and Applications* 80 (2021): 4533-4551.

[16] Su, Jianyu, Meihua Wang, Zhenxin Wu, and Qingliang Chen. "Fast plant leaf recognition using improved multiscale triangle representation and KNN for optimization." *IEEE Access* 8 (2020): 208753-208766.

[17] Aakif, Aimen, and Muhammad Faisal Khan. "Automatic classification of plants based on their leaves." *Biosystems Engineering* 139 (2015): 66-75.

[18] Saleem, Gulshan, M. Akhtar, Nisar Ahmed, and Waqar Shahid Qureshi. "Automated analysis of visual leaf shape features for plant classification." *Computers and Electronics in Agriculture* 157 (2019): 270-280.

[19] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22, no. 10 (2009): 1345-1359.

[20] Herdiyeni, Yeni, and Ni Kadek Sri Wahyuni. "Mobile application for Indonesian medicinal plants identification using fuzzy local binary pattern and fuzzy color histogram." In *2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 301-306. IEEE, 2012.

[21] Sachar, Silky, and Anuj Kumar. "Survey of feature extraction and classification techniques to identify plant through leaves." *Expert Systems with Applications* 167 (2021): 114181.

[22] Beikmohammadi, Ali, and Karim Faez. "Leaf

classification for plant recognition with deep transfer learning." In *2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 21-26. IEEE, 2018.

[23] Paymode, Ananda S., and Vandana B. Malode. "Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG." *Artificial Intelligence in Agriculture* 6 (2022): 23-3