A Comparative Study of Supervised and Semi-Supervised Learning for Mango Leaf Disease Classification

by

Shahriar Hasan
ID: 2010776115
Session:2019-20

Department of Computer Science and Engineering
University of Rajshahi

Introduction

- Mango is a vital crop in Bangladesh and globally
- Leaf diseases reduce yield and farmer income
- Manual diagnosis is slow, subjective, and expert-dependent
- Need automated, fast, and accurate disease recognition
- Early detection of leaf diseases can prevent spread and improve overall crop productivity.

Figure 1: Affected Mango leaf

Objective

- Compare supervised vs semi-supervised learning methods
- Evaluate multiple transfer learning models
- Develop a lightweight CNN for real-time usage
- Deploy a working mobile application for farmers

Literature Review

Rizvee et al., 2020 – LeafNet: 7 mango leaf diseases, 98.55% avgerage accuracy, lightweight CNN.

Rao et al., 2021 – Modified AlexNet: Mango & grape leaves, 89% & 99% accuracy, real-time app.

Zhang et al., 2020 – GoogLeNet/CIFAR10: Maize leaf diseases, 98.9% accuracy, faster training.

Bhujel et al., 2021 – CBAM-CNN: Tomato leaf diseases, 99.69% accuracy, low-resource devices.

Arivazhagan & Ligi, 2019 – CNN: 5 mango leaf diseases, 96.67% accuracy.

Rajbongshi et al., 2021 – Transfer Learning: DenseNet201 best, 98% accuracy, 1500 mango images.

Mia et al., 2020 – Neural Network Ensemble (NNE): Mango Leaf Disease Recognition, 80% average accuracy.

Dataset Details

Total images: 4,000
Classes: 8 (7 diseases + 1 healthy)
Images per class: 500
Image size: 240×320 pixels
Collection sites:

- Sher-e-Bangla Agricultural University orchard
- Jahangirnagar University orchard
- Udaypur village mango orchard
- Itakhola village mango orchard

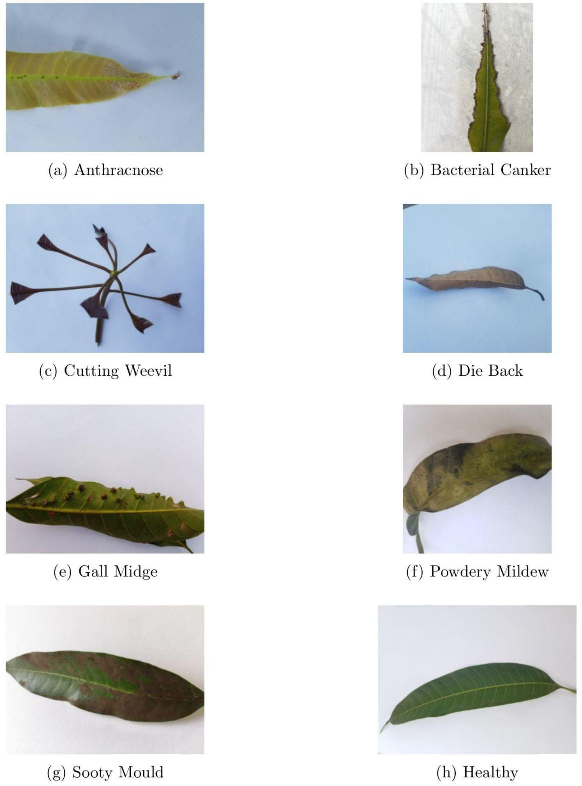


Figure 2: Representative images of each class

Methodology

Data Preparation

- Data split
  - Supervised: 80% train, 10% validation, 10% test
  - Semi-supervised: 25% labeled + 75% unlabeled data
- Normalization of pixel values.[0-1]
- Data augmentation (rotation, translation, flipping, Shear transformation, color jitter, etc.) to increase diversity

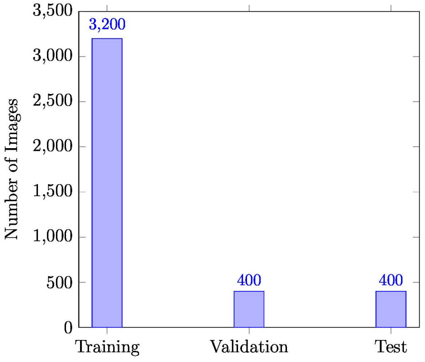


Figure 3: Supervised dataset distribution

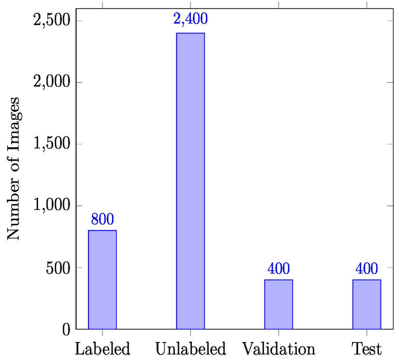


Figure 4: Semi-supervised dataset distribution

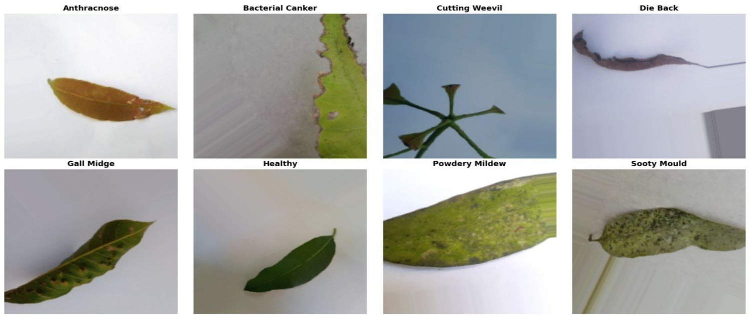


Figure 5: Augmented data

Methodology

Pretrained Model Architecture

- Models: VGG16, MobileNetV2, ResNet101, DenseNet121
- Global Average Pooling for spatial feature reduction
- Dense layers with ReLU and L2 regularization for strong feature learning
- Batch Normalization for stable and faster training
- Dropout layers to reduce overfitting
- Softmax output layer for multi-class classification

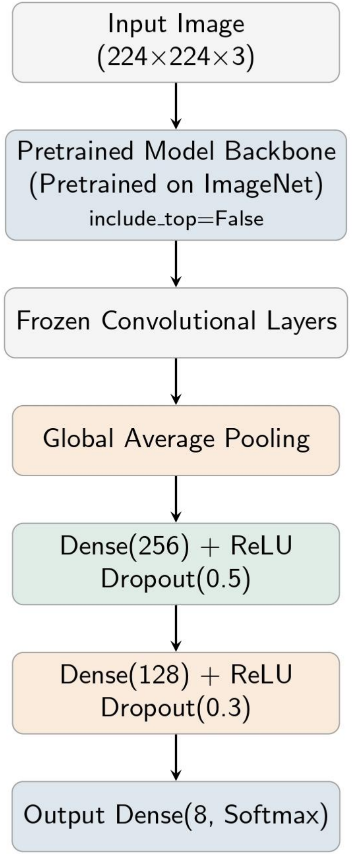


Figure 6: Pretrained model architecture

Methodology

Proposed Model Architecture

- Trainable params: 489,096 (1.87 MB)
- 4 convolutional blocks + classification head
- Filter progression: 32→64→128→256
- Progressive dropout: 20%→30%→40%→50%
- L2 regularization (1e-4) for all layers
- Batch Normalization: Applied after each convolutional layer

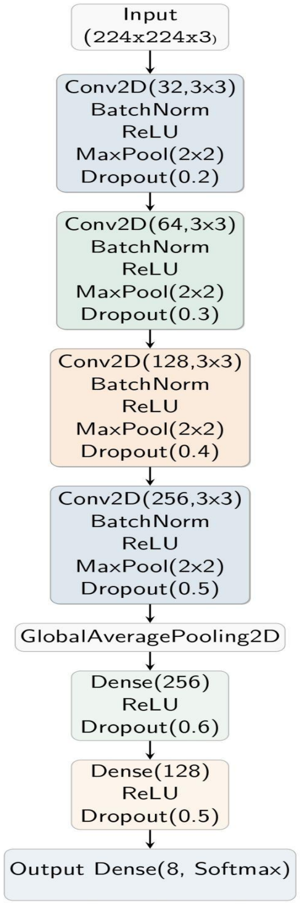


Figure 7: Proposed model architecture

Methodology

Supervised Learning

- Labeled data for training
- Models: VGG16, MobileNetV2, ResNet101, DenseNet121, Proposed CNN

Semi-supervised Learning

- Phase 1: Train on 25% of labeled data
- Pseudo-labeling: generates predictions on unlabeled data, with high confidence(≥95%).
- Phase 2: Train by labeled + confident pseudo-labeled data.

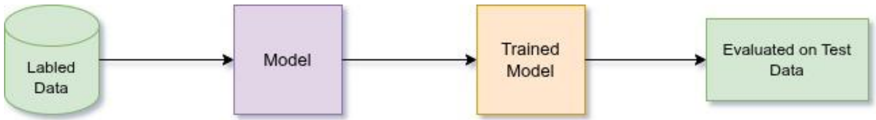


Figure 8: Supervised Learning

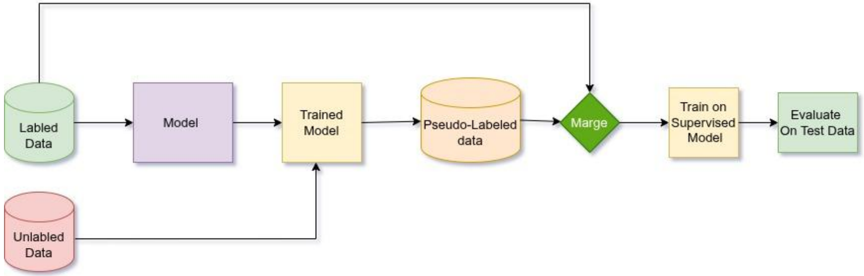


Figure 9: Semi-supervised Learning

Experimental Setup

Hardware Configuration

GPU: NVIDIA Tesla T4 (16 GB VRAM)
RAM: 29 GB (Kaggle), 12.7 GB (Colab)
CPU: Intel Xeon (cloud backend)
Storage: Cloud SSD
Environment: Linux (Ubuntu 20.04)
Platforms Used: Kaggle & Google Colab

Software Configuration

Framework: TensorFlow 2.8, Keras
Language: Python 3.8
Libraries: NumPy, Pandas, Matplotlib, Scikit-learn, Open-CV

Training Configuration

Batch size: 32
Number of epochs: Up to 100 with early stopping
Optimizer: Adam
Learning rate scheduling: ReduceLROnPlateau
Loss function: Categorical cross-entropy

Evaluation matrix

Primary: Accuracy
Secondary: Precision, Recall, F1-Score

-Confusion Matrix, Grad-CAM, t-SNE

Results

Model Performance Comparison

Model	Learning Type	Validation Accuracy	Test Accuracy
VGG16	Supervised	0.9875	0.9850
VGG16	Semi-Supervised	0.8950	0.8775
MobileNetV2	Supervised	0.9100	0.9125
MobileNetV2	Semi-Supervised	0.9750	0.9850
DenseNet121	Supervised	0.9575	0.9450
DenseNet121	Semi-Supervised	0.9875	0.9700
ResNet101	Supervised	0.9975	0.9800
ResNet101	Semi-Supervised	0.3600	0.3500
Proposed Model	Supervised	0.9900	0.9875
Proposed Model	Semi-Supervised	0.9075	0.9275

Table 1: Validation and Test Accuracy Comparison

Model	Learning Type	Precision	Recall	F1-Score
VGG16	Supervised	0.99	0.98	0.98
VGG16	Semi-Supervised	0.88	0.88	0.87
MobileNetV2	Supervised	0.91	0.91	0.91
MobileNetV2	Semi-Supervised	0.98	0.98	0.98
DenseNet121	Supervised	0.95	0.95	0.94
DenseNet121	Semi-Supervised	0.97	0.97	0.97
ResNet101	Supervised	0.98	0.98	0.98
ResNet101	Semi-Supervised	0.31	0.35	0.27
Proposed Model	Supervised	0.99	0.99	0.99
Proposed Model	Semi-Supervised	0.93	0.93	0.93

Table 2: Precision, Recall, and F1-Score Comparison

Results

Proposed Model Performance Curve

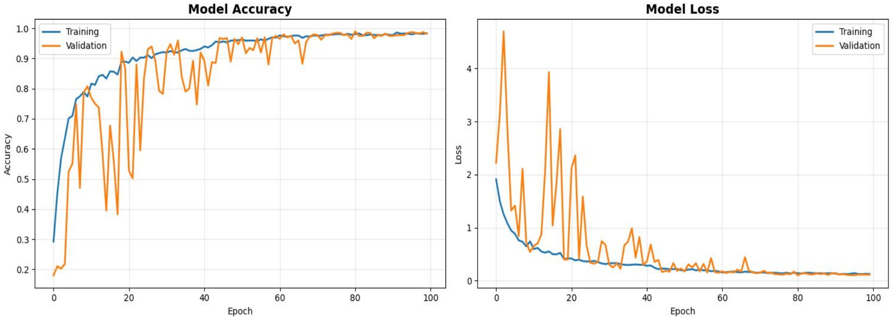


Figure 10: Supervised proposed model accuracy and loss curves

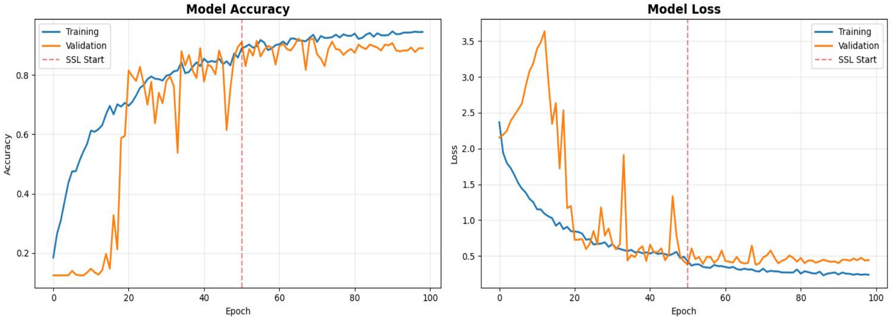


Figure 11: Semi-supervised proposed model accuracy and loss curves

Results

Proposed Model Confusion matrix

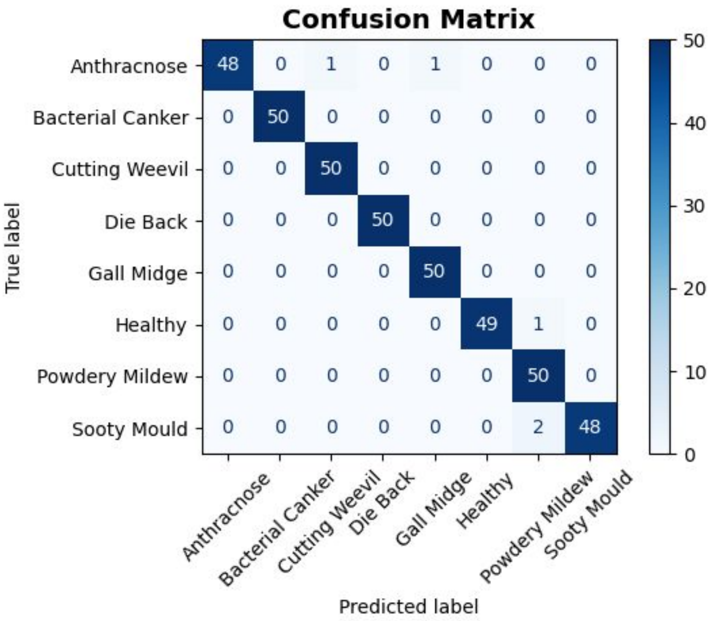


Figure 12: Confusion matrix of the supervised proposed model

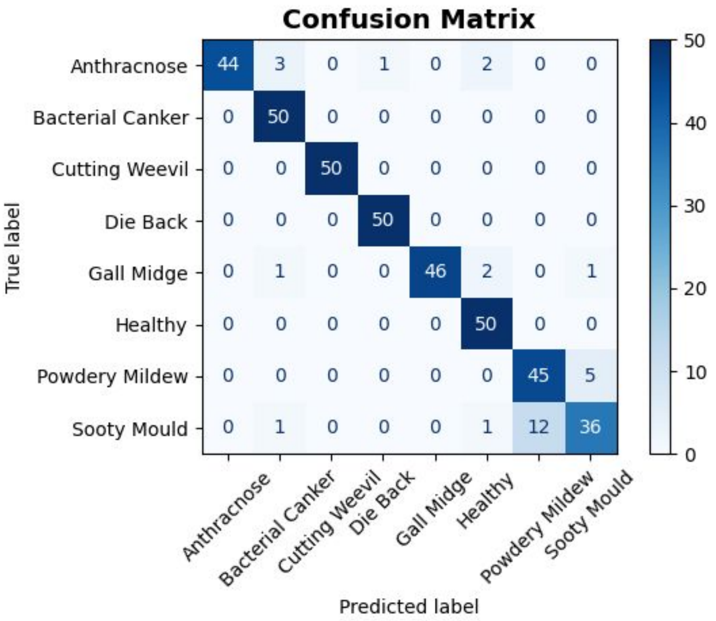


Figure 13: Confusion matrix of the semi-supervised proposed model

Discussion

- Supervised models performed strongly, especially ResNet101 and the Proposed Model
- Semi-supervised learning helped lighter models like DenseNet121 and MobileNetV2
- Some deep models struggled with noisy pseudo-labels, causing accuracy drops
- The Proposed Model provides a good balance of accuracy, speed, and stability with fewer labeled images

Model Visualization & Interpretation

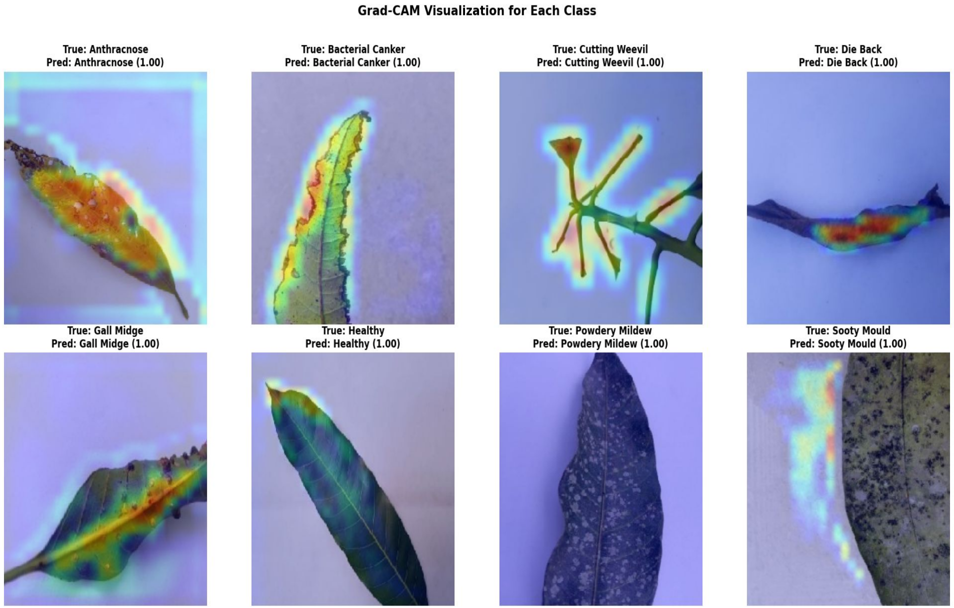


Figure 14: Grad-CAM heatmap visualization.

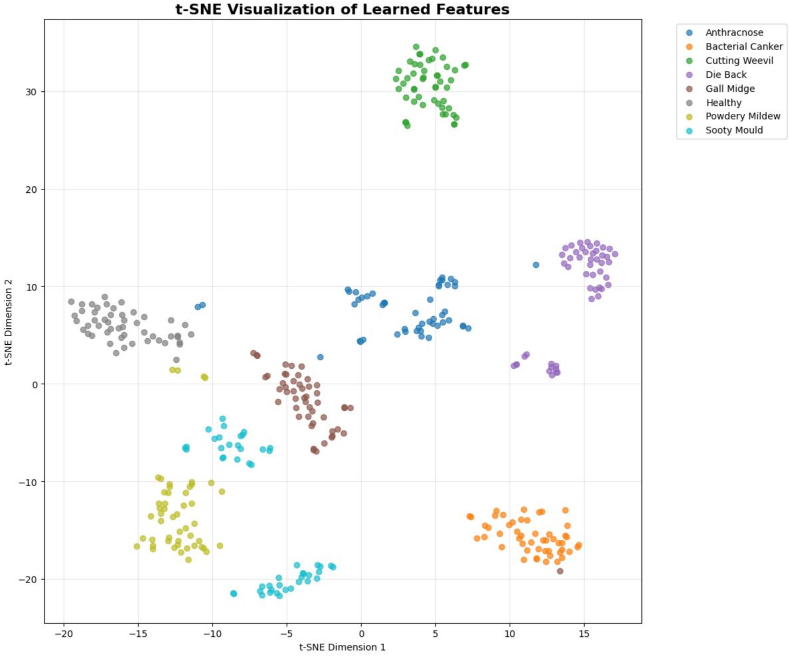


Figure 15: t-SNE feature space visualization.

Conclusion, Limitations, and Future Work

Conclusion

- Proposed Model is efficient, stable, and practical for real-world deployment.
- Semi-supervised learning can reduce labeling effort but depends on model robustness.
- Supervised learning remains reliable when labeled data are sufficient.
- Highlights the potential for mobile and edge-based agricultural systems.

Limitations

- Duplicate and highly similar images
- Limited diversity in images
- Data collection challenges
- Time and hardware constraints

Future Works

- Enhanced dataset collection
- Model selection and optimization
- Continued research in mango leaf disease classification

System Interface and Availability

PlantDoc Advisor App

- Real-time disease detection in field
- Simple user interface for farmers
- On-device inference with TFLite model
- Cloud integration for treatment advice

Technology Stack

- Flutter
- TensorFlow Lite
- Firebase
- Firestore

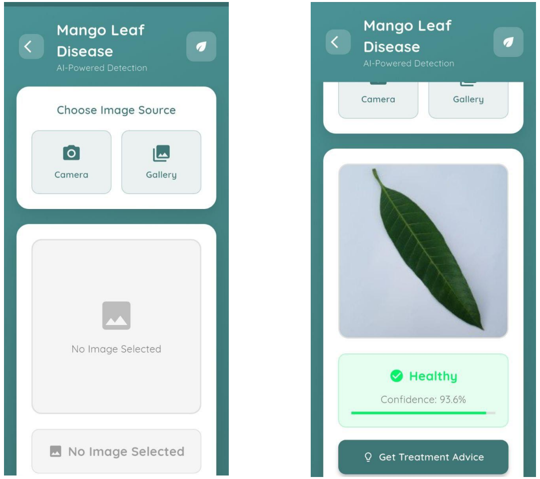
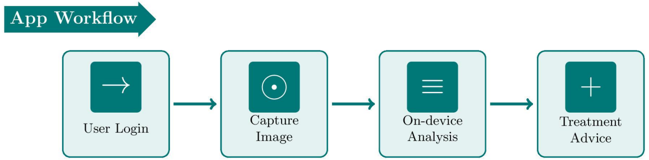


Figure 16: UI of image selection and classification result



THANK YOU