RESEARCH ON AI APPLICATIONS FOR PLANT LEAF DISEASE DETECTION AND CLASSIFICATION

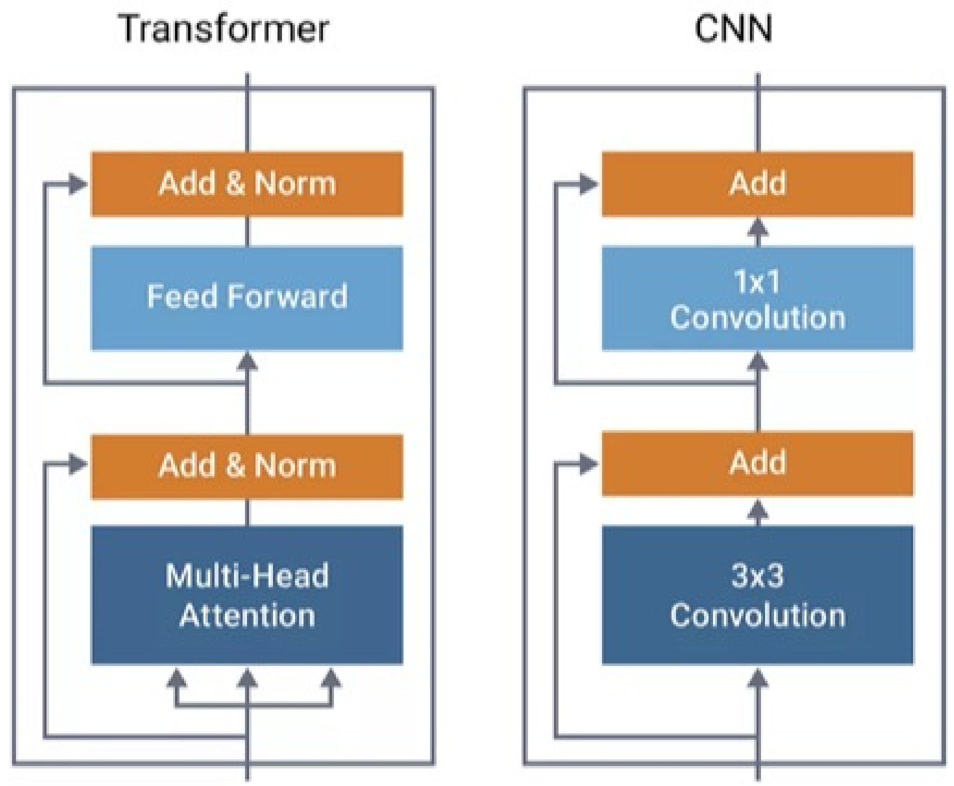
By Skin Issue

INTRODUCTION

Demand for sustainable agriculture requires accurate plant disease diagnostics
Major challenges:
- High symptom variability
- Environmental noise
- Need for generalizable recognition across species

A unified deep learning framework using:
- Convolutional Neural Networks (CNNs)
- Transformer-based architectures

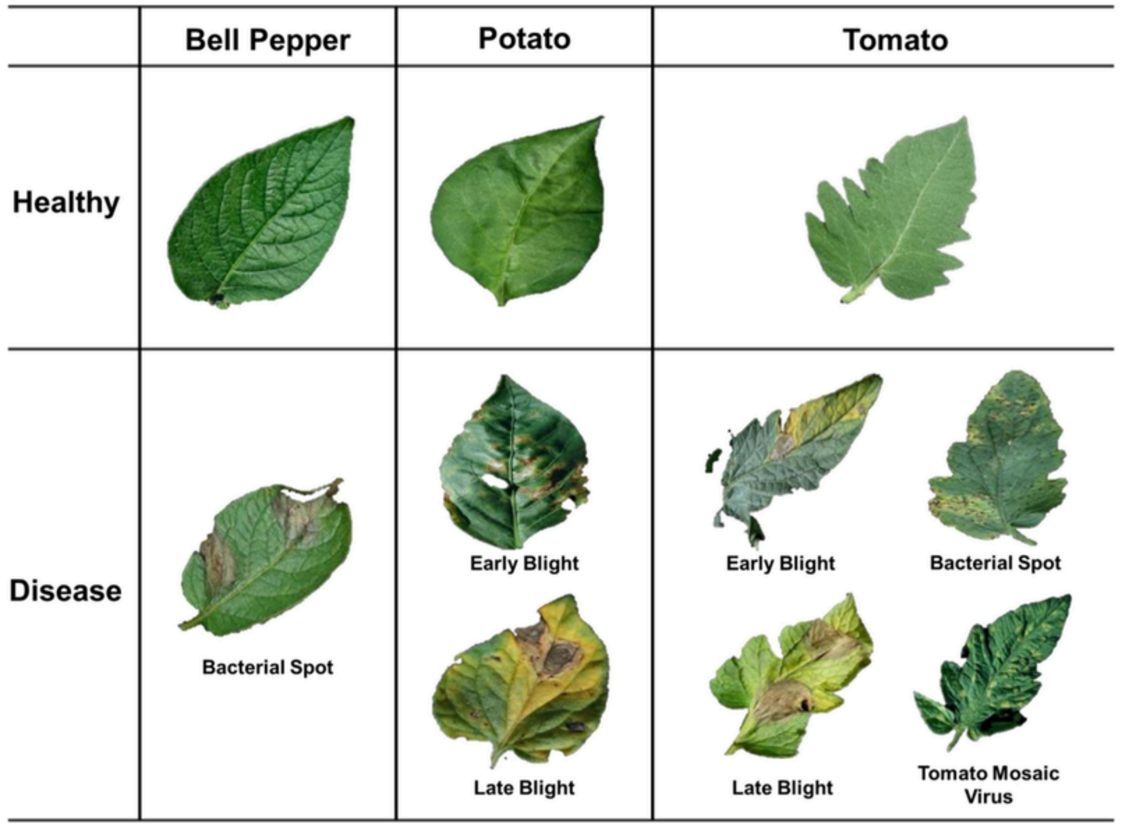
Machine Learning algorithms for classification based on extracted features
Focus on feature extraction from leaf images for disease classification
Emphasis on model robustness under complex input conditions



INTRODUCTION

Goals:

- Develop a framework for automatic leaf disease recognition
- Analyze and compare performance using CNNs, Transformers and Machine Learning
- Improve classification accuracy under complex visual conditions
- Address multi-label and multi-class scenarios in plant pathology
- Establish a generalizable approach across different plant species

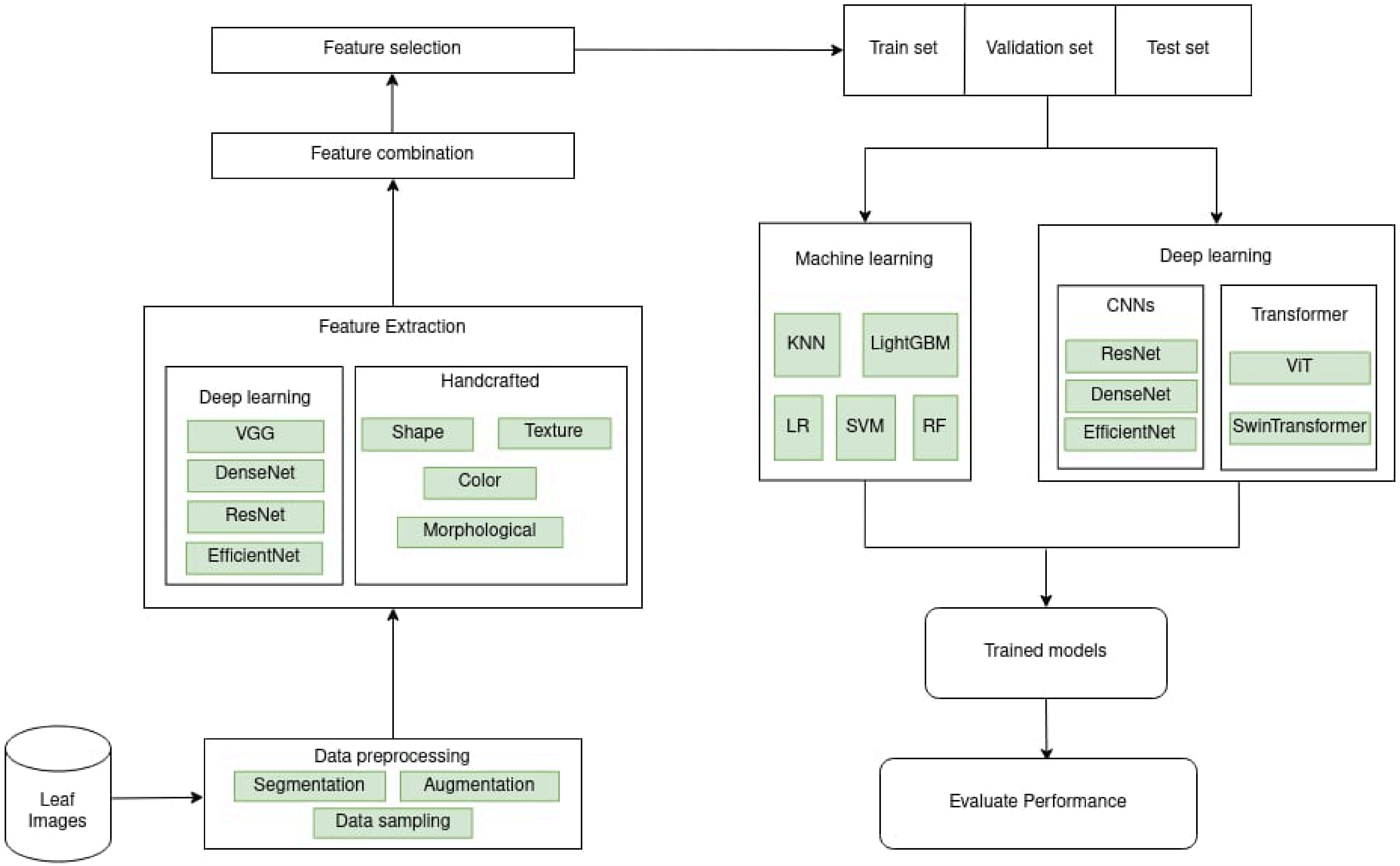


RELATED WORK

Zhang et al. (X. Ma et al.)	2024	ERCP-Net (Residual structure + Adaptive Channel Attention)	PlantVillage & AI Challenger 2018	99.82% & 86.21%
Calma et al.	2023	MobileNetV3 (lightweight CNN)	Augmented cassava leaf & stem; improved generalization	93,20%
Ensemble DL (EfficientNet, SEResNeXt, ViT, DeIT,	2023	Ensemble of SOTA DL models	Cassava disease detection system	90,75%
Transfer learning (EfficientNet-B0)	2025	EfficientNetB0 + data augmentation	Rcgn on cassava leaves, stems & tubers	94% (leaves), 90% (stems), 92% (tubers)

METHODOLOGY

Workflow Diagram:



METHODOLOGY

Dataset

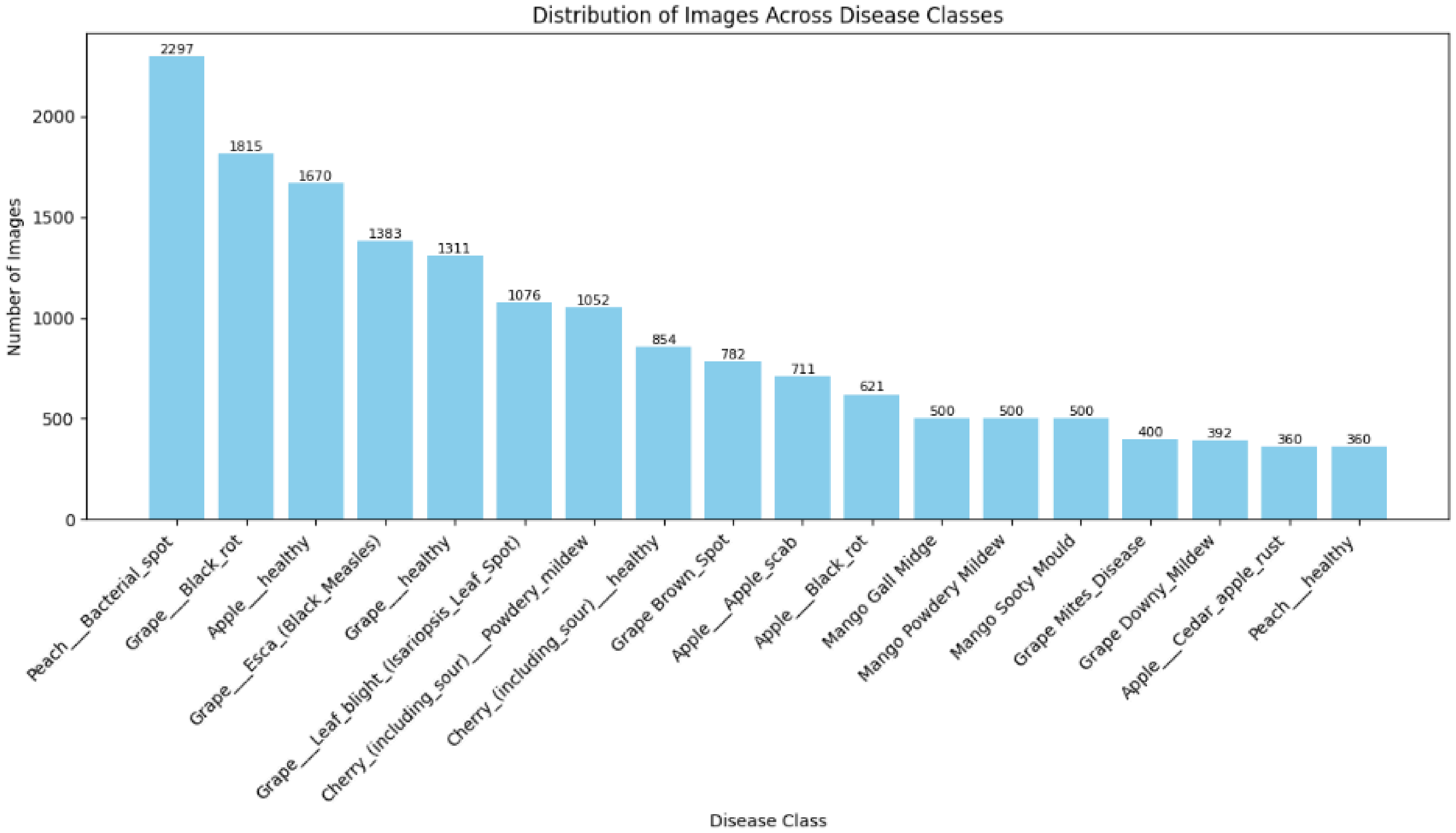
Total images: 16584

Image Size: 256x256

Image Format: JPG

Number of Class: 18

Train - Test - Val: 70 - 15 - 15



METHODOLOGY

Data Sampling & Preprocessing

Sampling:

- Collected a diverse set of plant leaf photos.
- Included both complex and simple backgrounds
- Cover all real-world variations (overlapping leaf, light condition)

Preprocessing:

- Resized to 224x224.
- Normalized pixels to [0, 1].

Objective: Ensure a robust and accurate classification model for real-world conditions.

METHODOLOGY

Segmentation pipeline

Sample data

Simple cases:
Isolated leaf and background

Grabcut/SAM

Yolov11-Segmentation

Segmented images

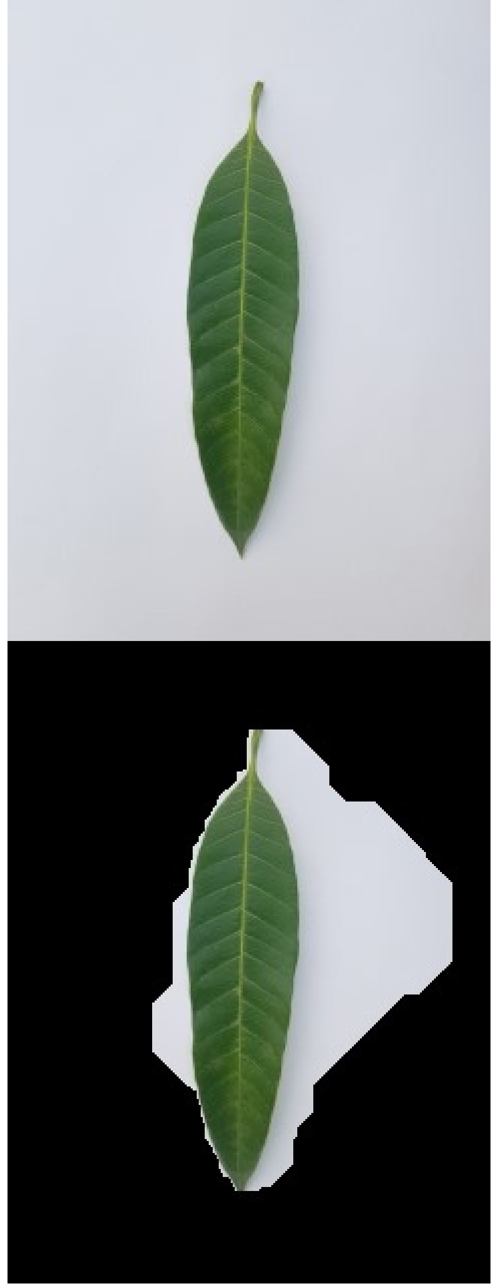
Complex cases:
Blended main leaf with background

Label manually

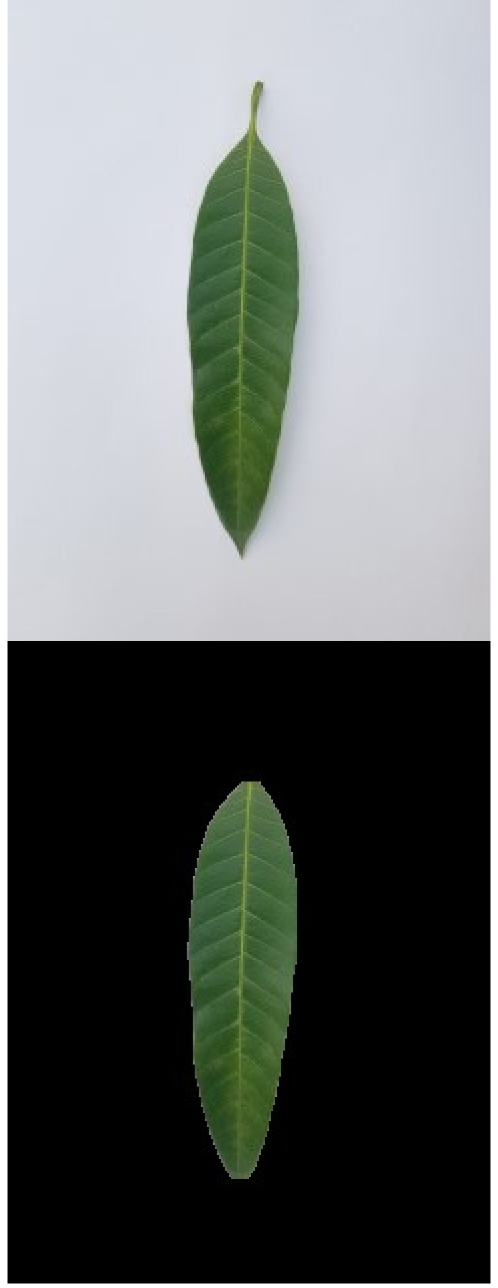
METHODOLOGY

Image Segmentation

Grabcut



SAM



LabelStudio

METHODOLOGY

Data Augmentation

Original image

Horizonal flip

Vertical flip

Brightness

Saturation

Contrast

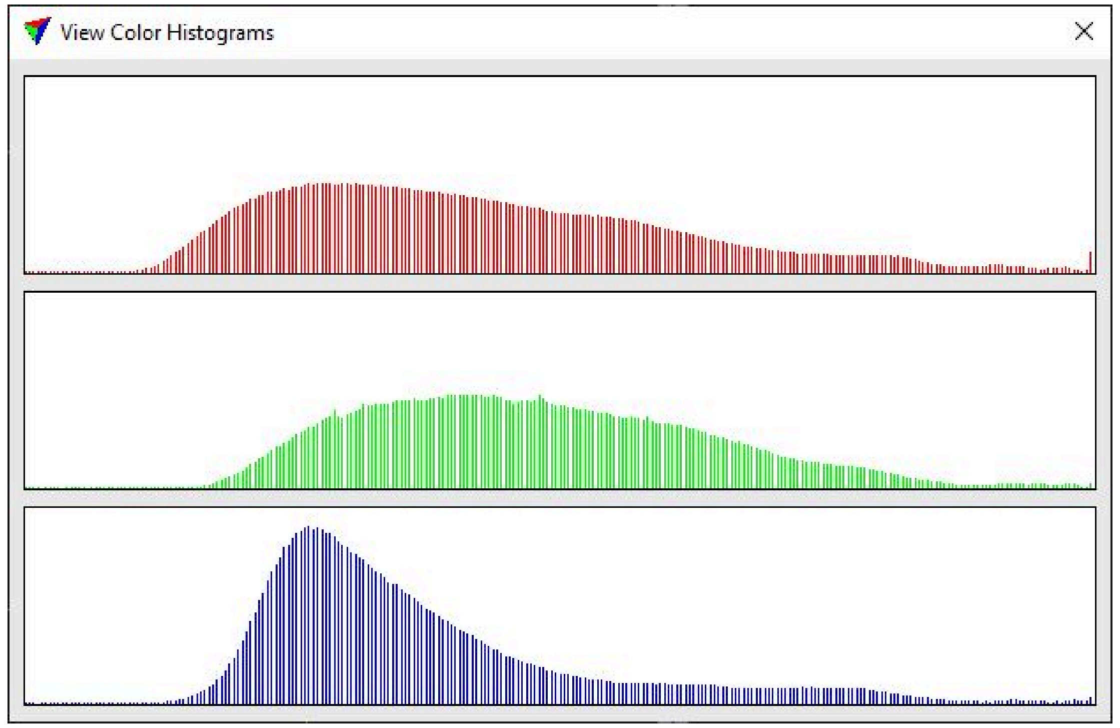
Rotate

METHODOLOGY

Feature Extraction (Machine learning)

Color Features

- Color Space: RGB, HSV, Lab, YCbCr, XYZ
- Statistical Summaries:
    - Mean: Indicates the dominant or overall color of the leaf.
    - Standard Deviation: Measures the dispersion or variation of pixel values around the mean value.
    - Skewness: Measures the asymmetry of pixel intensity distribution within a color channel.
    - Kurtosis: Measures the "peakedness" of the distribution and the "thickness" of its tails.

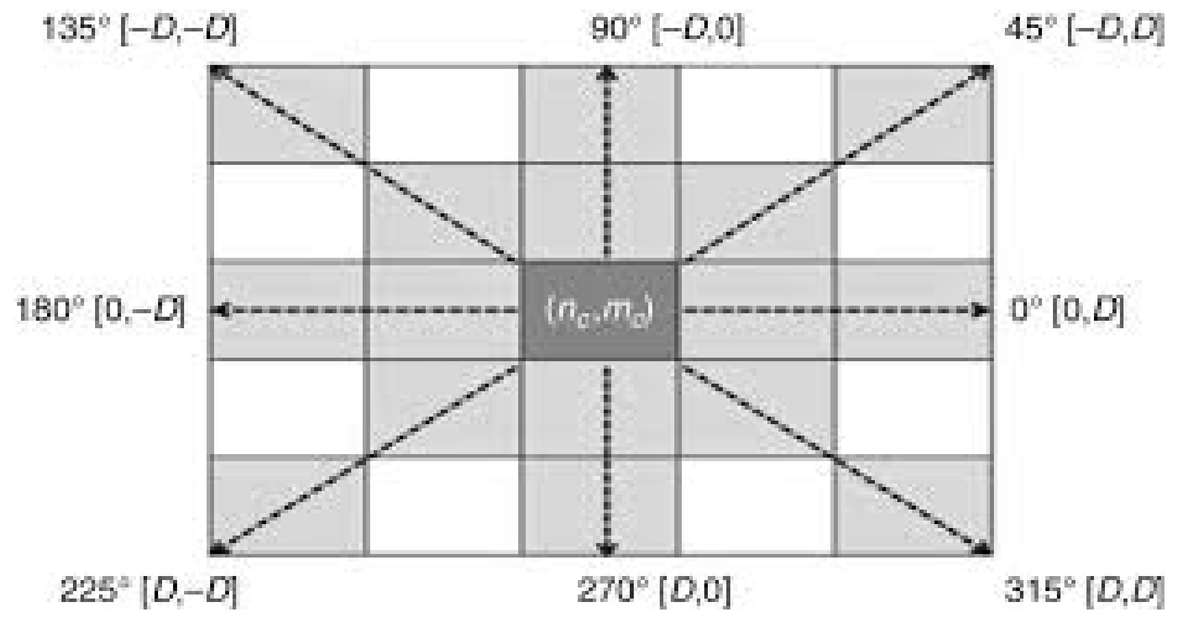


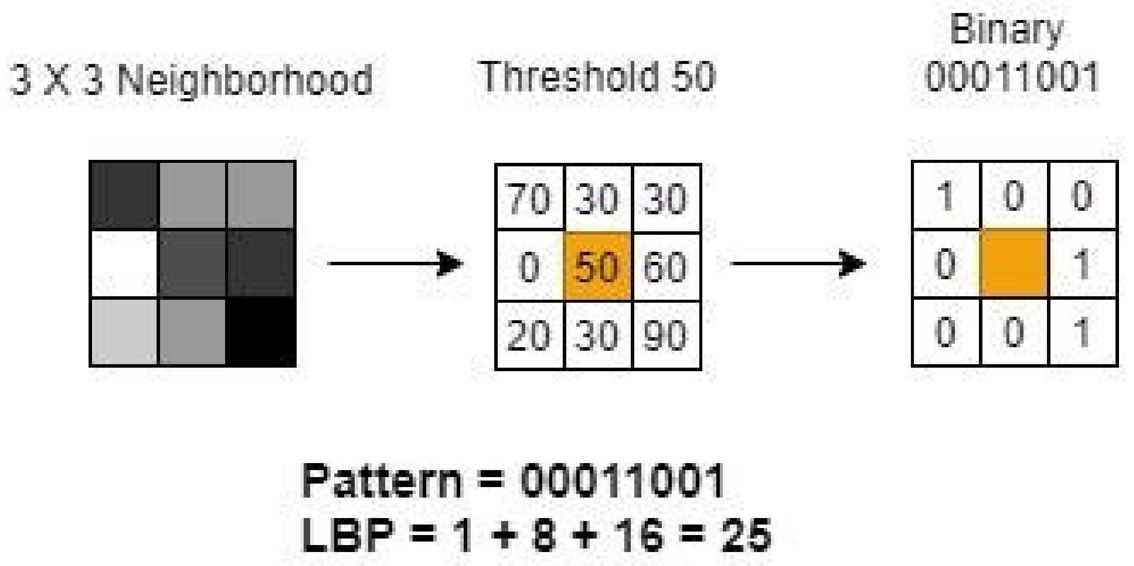
METHODOLOGY

Feature Extraction (Machine learning)

Texture Features:

- GLCM (Gray-Level Co-occurrence Matrix): Typically used to detect changes in coarseness, fineness, uniformity, or repetitive patterns on the leaf surface caused by disease. Examples include blotches, cracks, or alterations in leaf vein density.
- LBP (Local Binary Pattern): Commonly used to identify local micro-texture patterns, such as small spots, well-defined necrotic regions, or changes in the epidermal texture of the leaf due to fungi or bacteria.



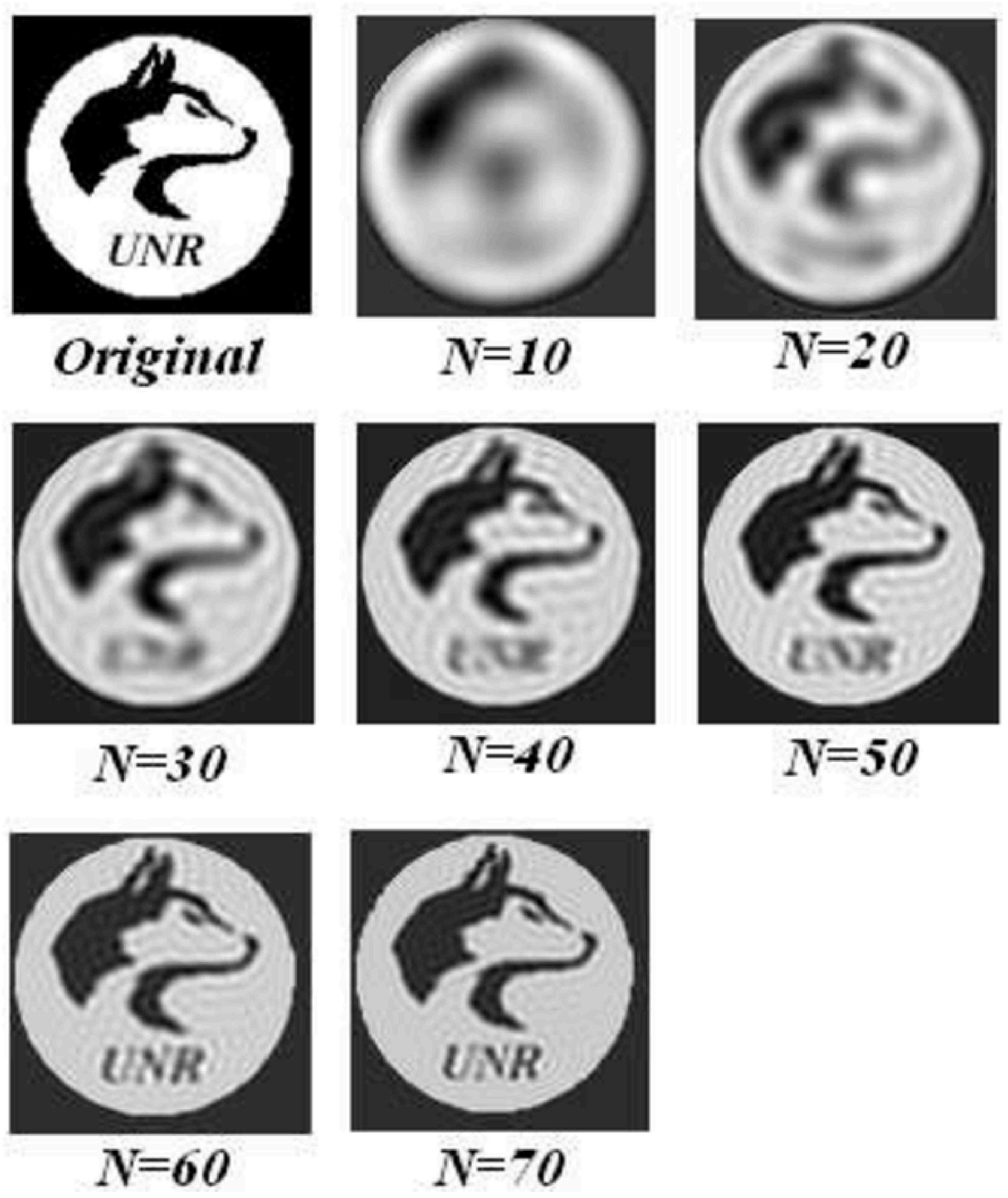


METHODOLOGY

Feature Extraction (Machine learning)

Shape Features:
Zernike Moments can analyze the overall shape of the entire leaf. Zernike Moments are highly robust to geometric transformations.

- Classifying diseases that cause significant changes or deformities to the leaf's general shape.
- Differentiating between various plant species based on their unique leaf shapes.



METHODOLOGY

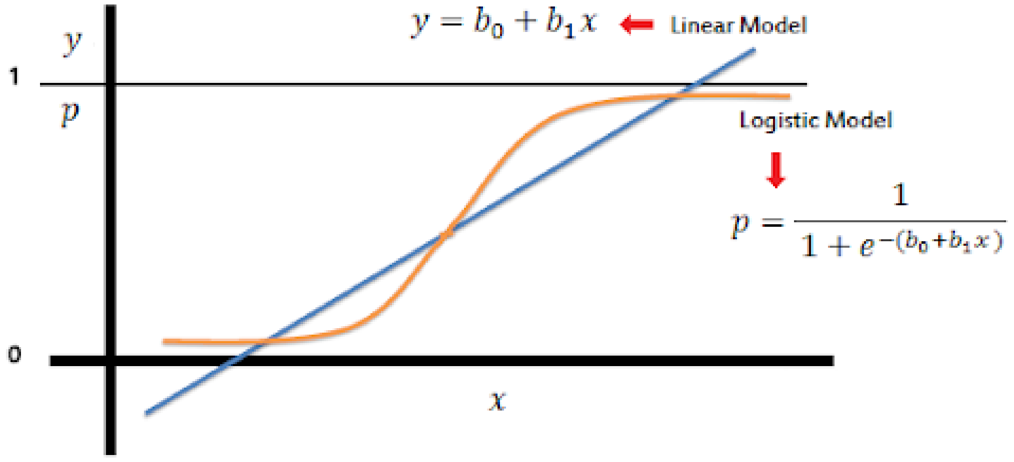
Feature Extraction (Machine learning)

Morphological Features:

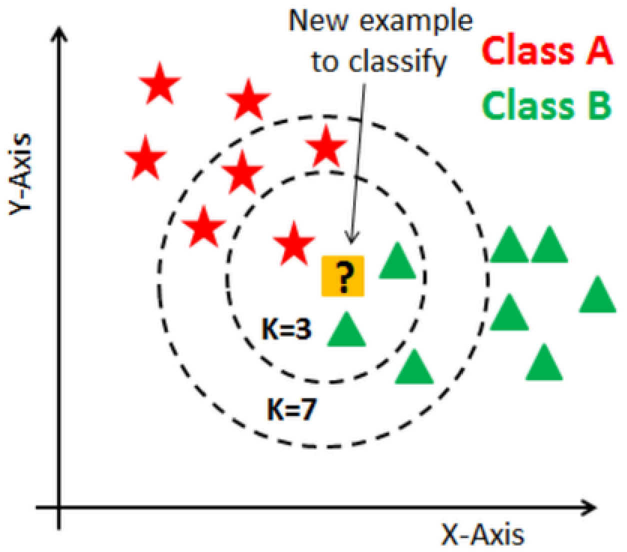
- Area & Perimeter: The total number of pixels that make up the segmented leaf and the total length of the border
- Aspect Ratio: The ratio of the width to the height of the object's bounding box
- Roundness: Measures how closely the shape resembles a perfect circle.
- Solidity & Convexity: Indicating the degree to which a shape is "filled" or "indented”.
- Symmetry: How closely the leaf can be divided into two (or more) identical halves.
- Equivalent Diameter: Standardizing the size of an object to the diameter of a comparable circle.

METHODOLOGY

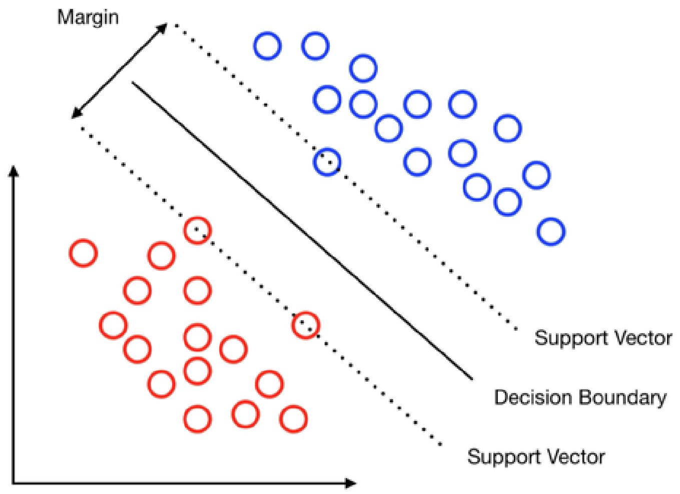
Machine Learning Classifiers:



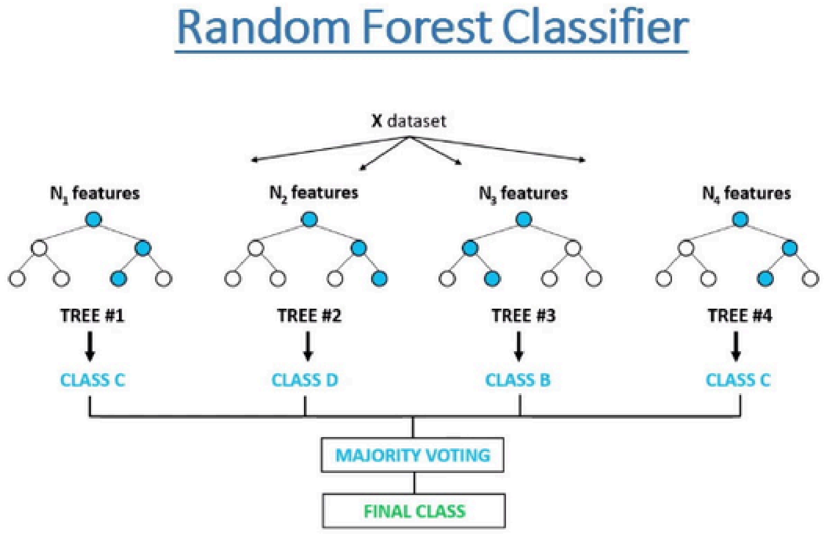
- Logistic Regression



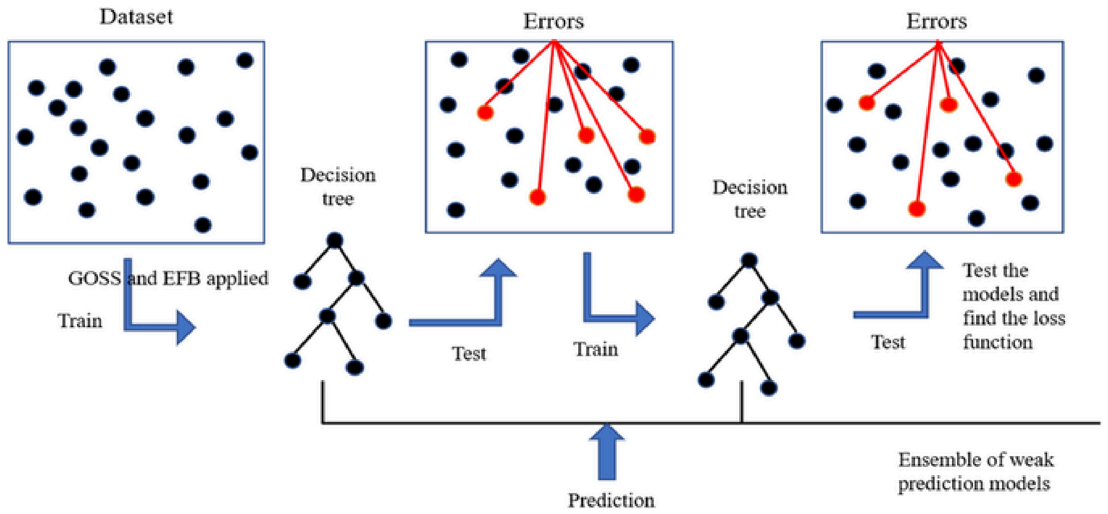
- K-Nearest Neighbors



- Support Vector Machine



- Random Forest

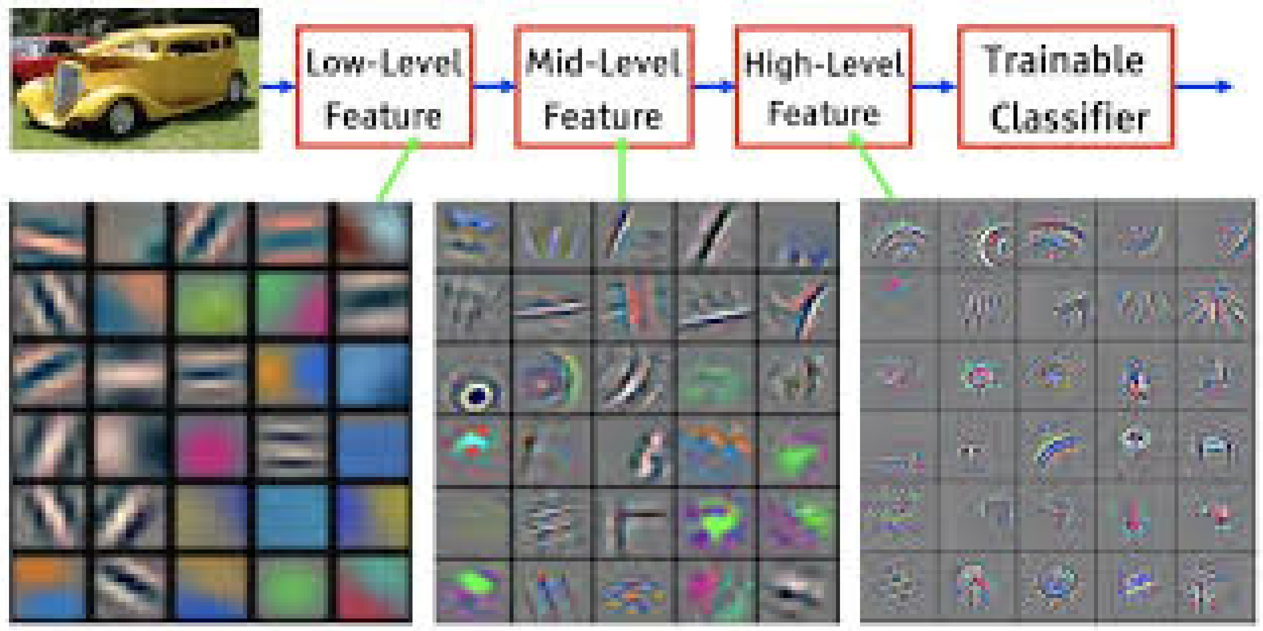


- LightGBM

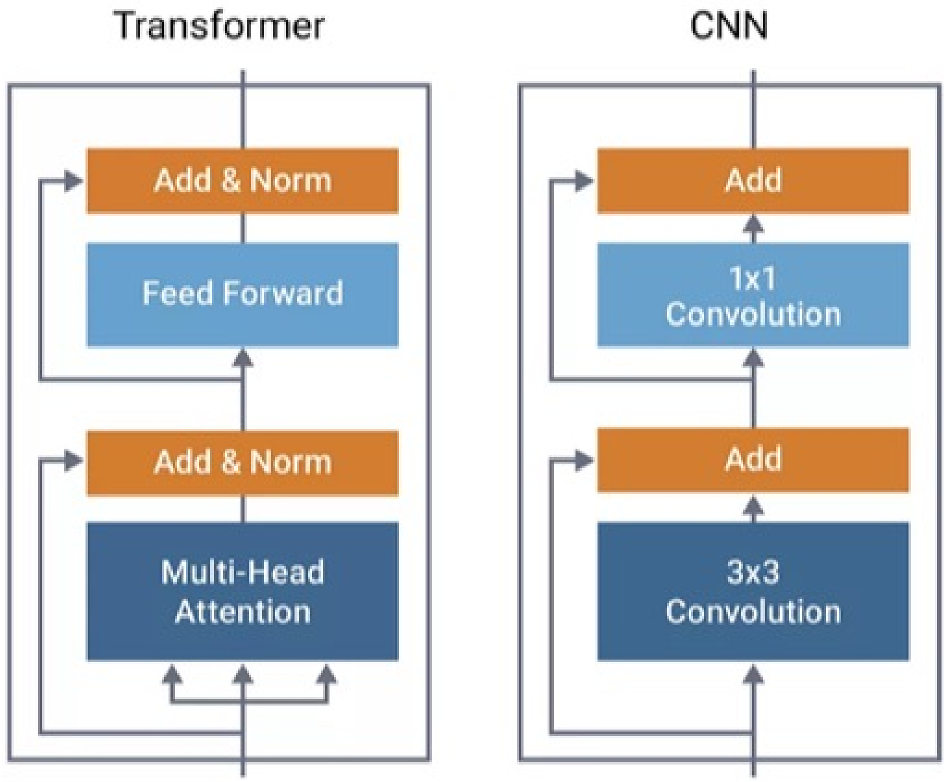
METHODOLOGY

Deep Learning-based

Deep Feature Extraction: Using pre-trained models (DenseNet201, ResNet50/ResNet50V2, ResNet101, EfficientNetB0) to extract powerful features and fed into traditional machine learning classifiers.



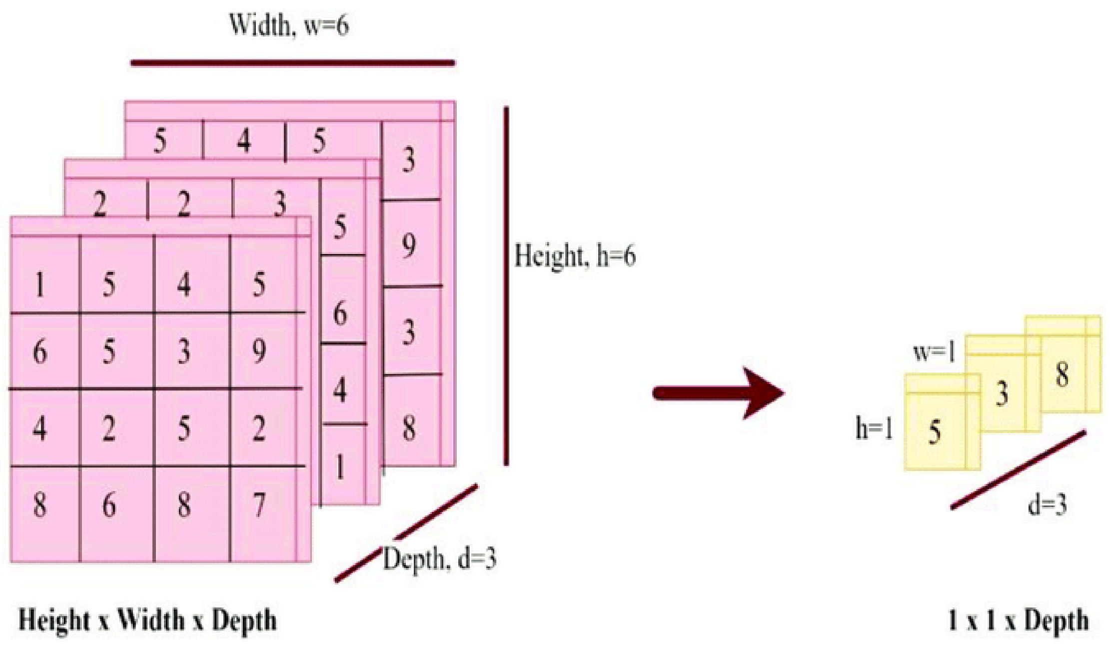
End-to-End Deep Learning: Directly training deep neural networks (CNNs and Transformers) for direct disease categorization.

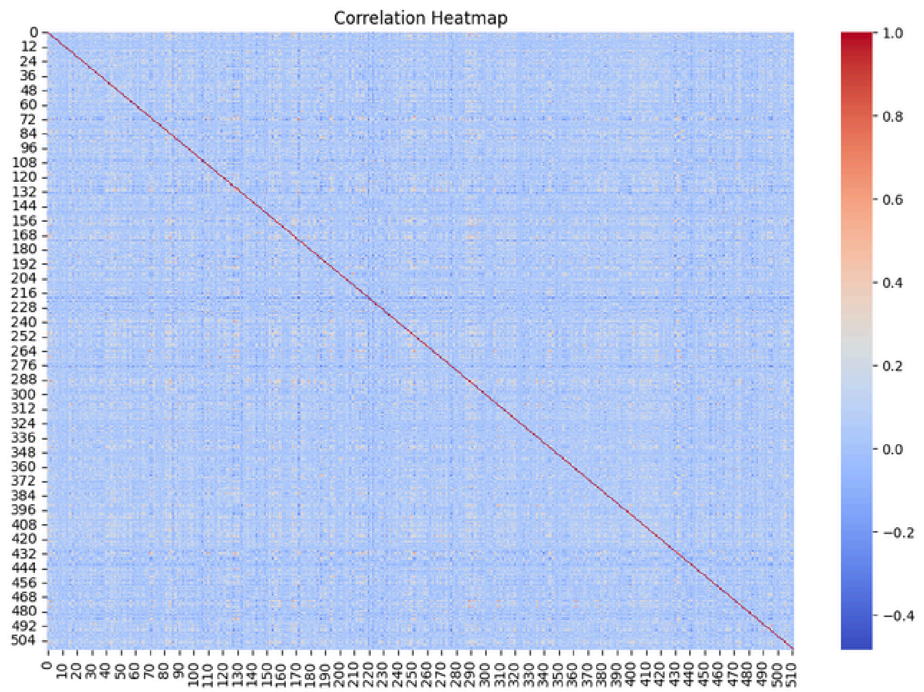


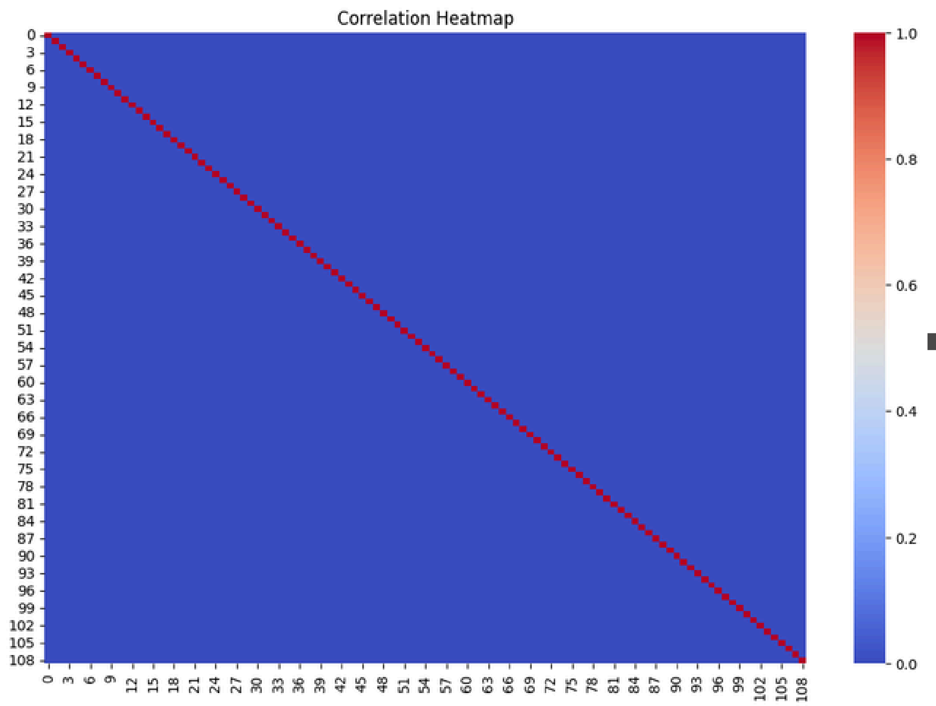
METHODOLOGY

Deep Feature Extraction:

- Pretrained Models Used: VGGs, ResNets, DenseNets, EfficientNets
- Feature Source: Extracted from GAP layer of each model
- Input Size: Images resized to 224×224 (CNN-based)
- Apply PCA for dimensionality reduction (Kaiser criterion: eigenvalue > 1)
- Feed features into Machine Learning classifiers







METHODOLOGY

End-to-End Deeplearning

Limitations of Traditional Methods	Advantages of End-to-End Deep Learning
Manual Feature Engineering	Automatic Feature Learning
Requires domain expertise to craft features	Learns complex patterns directly from raw data
Relies on segmentation or preprocessing models	Operates directly on raw images without segmentation
Often sensitive to noise and irrelevant features	Robust to noise by using hierarchical feature abstraction

Objective: Our goal is to surpass the limitations of conventional, manually-engineered feature pipelines. We aim to deliver a more flexible and efficient framework for accurate and effective plant leaf disease classification, directly from unprocessed visual data. This paves the way for intelligent crop monitoring and precision agriculture.

METHODOLOGY

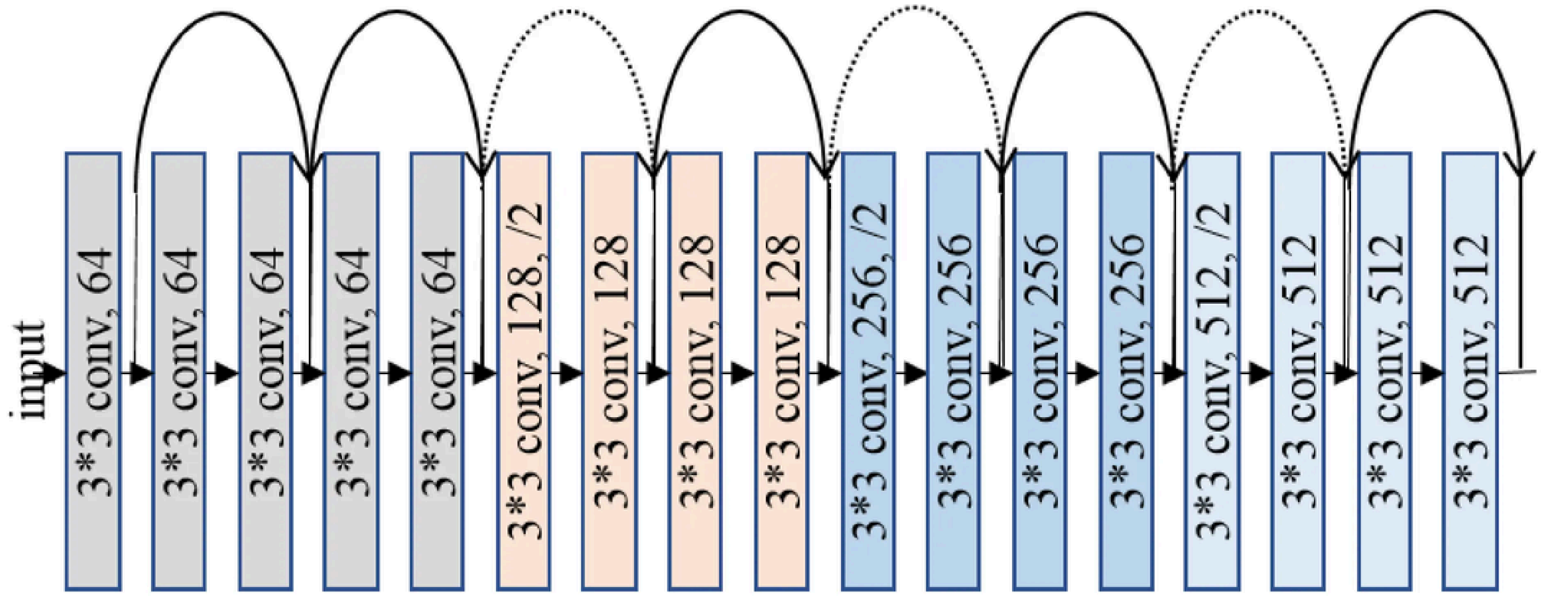
ResNet-Based Models (ResNet50, ResNet50V2, ResNet101)

Key characteristic:

- Introduces skip (residual) connections to allow better gradient flow.
- Solves the vanishing gradient problem, enabling deeper training.
- Residual blocks focus on learning only what's new.

Strengths:

- Stable training of deep networks.
- Strong baseline across many image tasks.

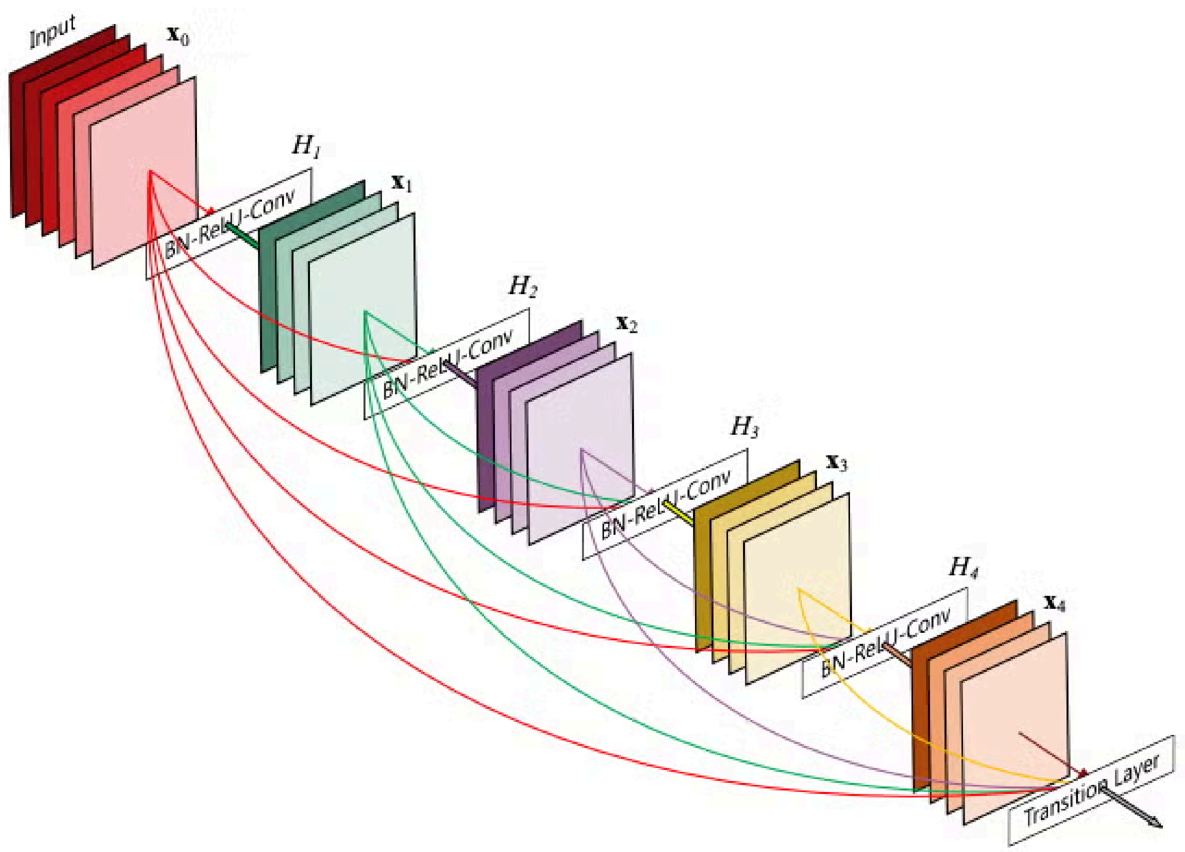


METHODOLOGY

DenseNet

Key characteristic:

- Each layer is connected to all previous layers (dense connectivity).
- Promotes feature reuse and strong gradient flow.
- Achieves high accuracy with fewer parameters than ResNet.



Strengths:

- Parameter-efficient yet powerful.
- Strong generalization even on small datasets.

METHODOLOGY

Efficient Net

Key characteristic:

- Uses compound scaling to balance depth, width, and resolution.
- EfficientNet is built using Mobile Inverted Bottleneck Convolution (MBConv) blocks

Strengths:

- Lightweight and fast with minimal resources.
- Scales well from mobile to cloud-level deployment.

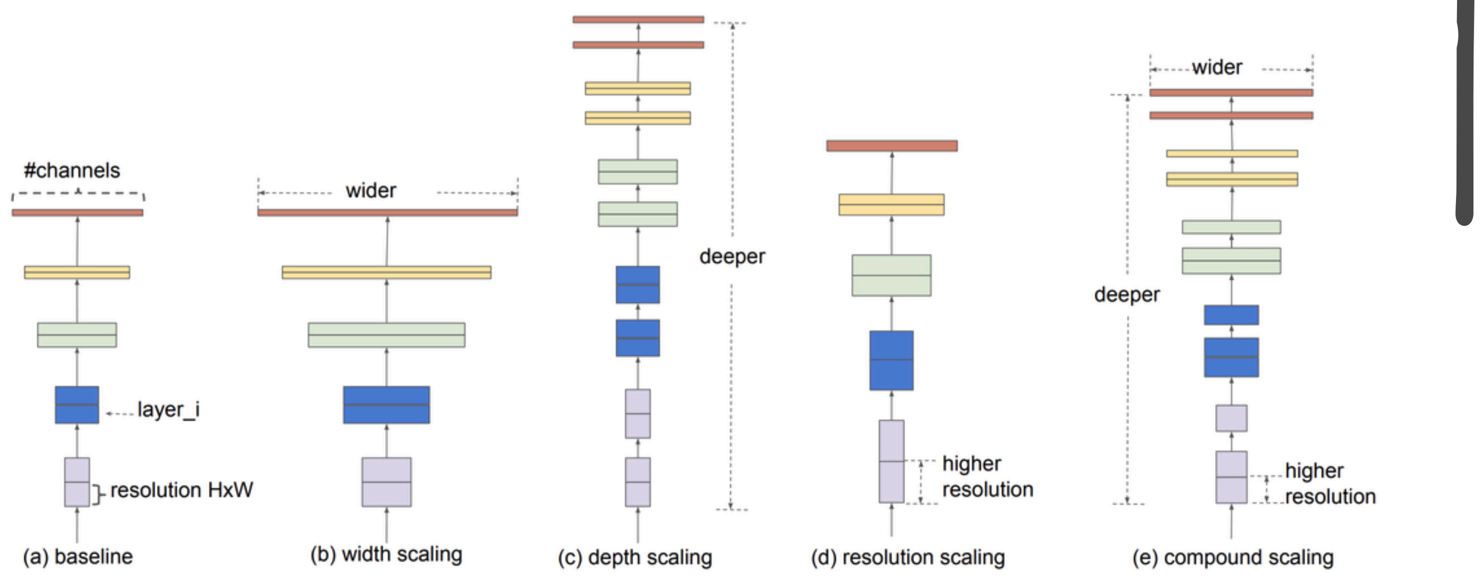


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

METHODOLOGY

Summary CNN-Based models

- ResNet introduces skip connections to enable the training of very deep networks, making it a popular baseline in many plant disease classification tasks. Its ability to handle complex leaf patterns and minimize vanishing gradients has made it effective on datasets like PlantVillage. Studies such as “Plant Disease Detection Using Deep Learning Techniques ” (J. of Intelligent & Applied Planning, 2025), Fine-tuned ResNet152 and DenseNet169 on PlantVillage
- DenseNet connects each layer to all previous ones, encouraging feature reuse and improving gradient flow. This makes it ideal for small or imbalanced datasets in plant pathology. For example, Zhang et al. (2020) applied DenseNet121 to apple leaf disease detection and achieved better generalization than deeper ResNet models.
- EfficientNet scales depth, width, and resolution in a balanced way, offering state-of-the-art accuracy with fewer parameters. It is highly suitable for resource-constrained applications such as mobile-based plant disease diagnosis. Liu et al. (2022) showed EfficientNet-B0 outperforming heavier models in tomato disease classification tasks with minimal computational cost.

METHODOLOGY

Feature	**ResNet**	**DenseNet**	**EfficientNet**
Key Idea	Skip (residual) connections	Dense connections between all layers	Compound scaling (depth, width, resolution)
Gradient Flow	Improved via skip connections	Excellent via dense connections	Stable with fewer layers
Parameter Efficiency	Moderate	High (reuses features)	Very high (optimized by NAS)
Accuracy	Strong baseline	Often better than ResNet with fewer params	State-of-the-art in many benchmarks
Training Speed	Fast	Slower due to dense connections	Very fast (lightweight)
Best Use Case	Deep classification tasks	Tasks requiring feature reuse	Tasks needing both speed & accuracy

METHODOLOGY

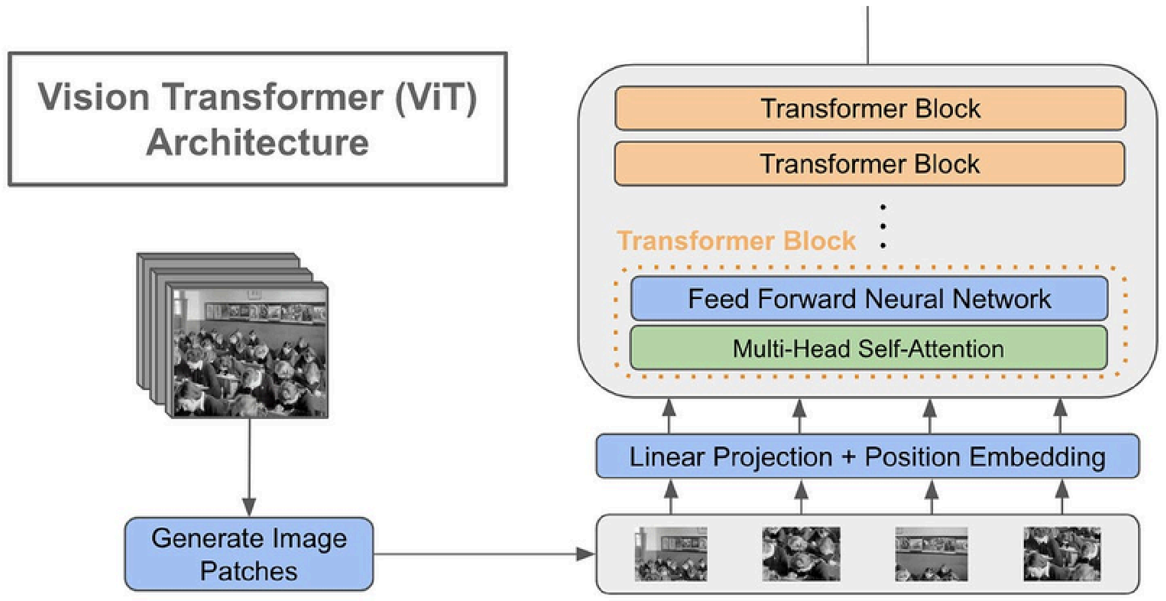
ViT (Vision Transformer)

Key characteristic:

- Treats images as sequences of **non-overlapping patches**, similar to words in NLP.
- Uses **pure transformer blocks** with multi-head self-attention (MHSA), without any convolutional layers.
- Requires large datasets to learn effectively due to **a lack of inductive bias** like locality or translation invariance.

Strengths:

- Excellent at capturing **global context** in images.
- Scales well with model size and data volume.
- Outperforms CNNs on large-scale datasets (ImageNet-21K, JFT-300M).



METHODOLOGY

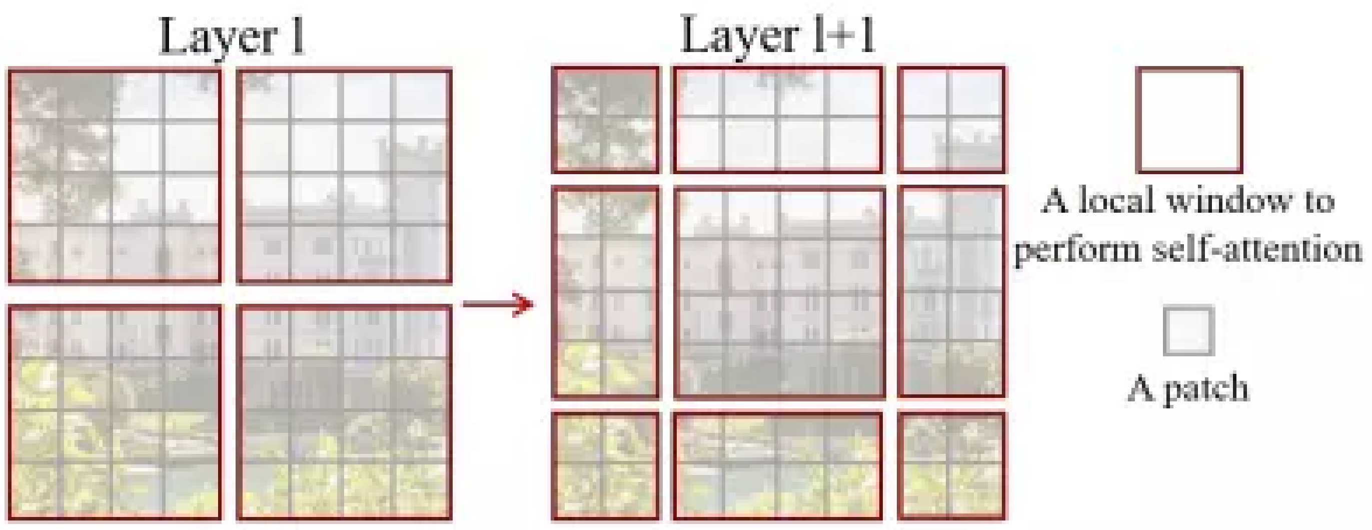
Swin Transformer

Key characteristic:

- Introduces **hierarchical architecture** using **shifted window attention** (local + global modeling).
- Maintains **linear computational complexity** with respect to image size.
- Mimics CNNs with features like multi-scale representation and locality.

Strengths:

- Strong performance on both **classification** and **dense prediction** tasks (e.g., detection, segmentation).
- Efficient and scalable for both small and large input resolutions.
- Combines the **best of CNN and Transformer worlds** (local bias + global attention).



METHODOLOGY

Summary Transformer-Based Models

- Vision Transformer (ViT) treats an image as a sequence of fixed-size patches and processes them using standard Transformer blocks. Its ability to model long-range dependencies and capture global context makes it powerful for high-resolution image classification tasks. However, ViT requires large-scale datasets for effective training. Studies such as Dosovitskiy et al. (2020) demonstrated ViT's superiority over CNNs on large benchmarks like ImageNet-21K.

- Swin Transformer introduces a hierarchical Transformer using shifted windows to compute self-attention locally and efficiently. It combines the strengths of CNNs (locality and scalability) with global attention from Transformers. Liu et al. (2021) showed Swin Transformer outperforming both CNN and ViT in image classification, object detection, and segmentation tasks, making it ideal for versatile agricultural vision systems.

METHODOLOGY

Feature	**Vision Transformer (ViT)**	**Swin Transformer**
Key Idea	Applies standard Transformer directly on image patches	Uses hierarchical structure with shifted windows
Image Representation	Divides image into **fixed-size non-overlapping patches**	Builds **local windows** and shifts them across layers
Scalability	Scales well but needs **large-scale pretraining**	Scales **efficiently** even on **smaller datasets**
Hierarchical Feature Map	No – all tokens are equal	Yes – supports **multi-scale representation**
Computation	High – global attention is expensive	Lower – local attention in windows is more efficient
Strengths	Strong in **global feature learning**, excels in large datasets	Balances **local & global context**, efficient & flexible
Best Use Case	Large image datasets (e.g., ImageNet-21K, JFT)	Versatile tasks incl. detection, segmentation, classification

METHODOLOGY

Training Details

Loss Function

- Used Cross-Entropy for multi-class classification (18 classes, treated independently).
- Addressed class imbalance by applying pos_weight (inverse of class frequency).

→ Minority classes get higher weights → prevent bias toward majority classes.

- This ensures fairer learning across all disease categories.

Optimizer & Learning Rate Schedule

- Used Adam optimizer (adaptive learning rate, robust to noisy/sparse gradients).
- Initial learning rate: 2×10^{-5}
- Applied Early Stopping:
  - Max: 500 epochs
  - Stop if no validation improvement for 10 epochs (patience = 10)

→ Helps avoid overfitting and unnecessary computation.

$$L(x, y) = -\text{weight}_y \cdot \log\left(\frac{\exp(x_y)}{\sum_{c=1}^{C} \exp(x_c)}\right)$$

RESULT

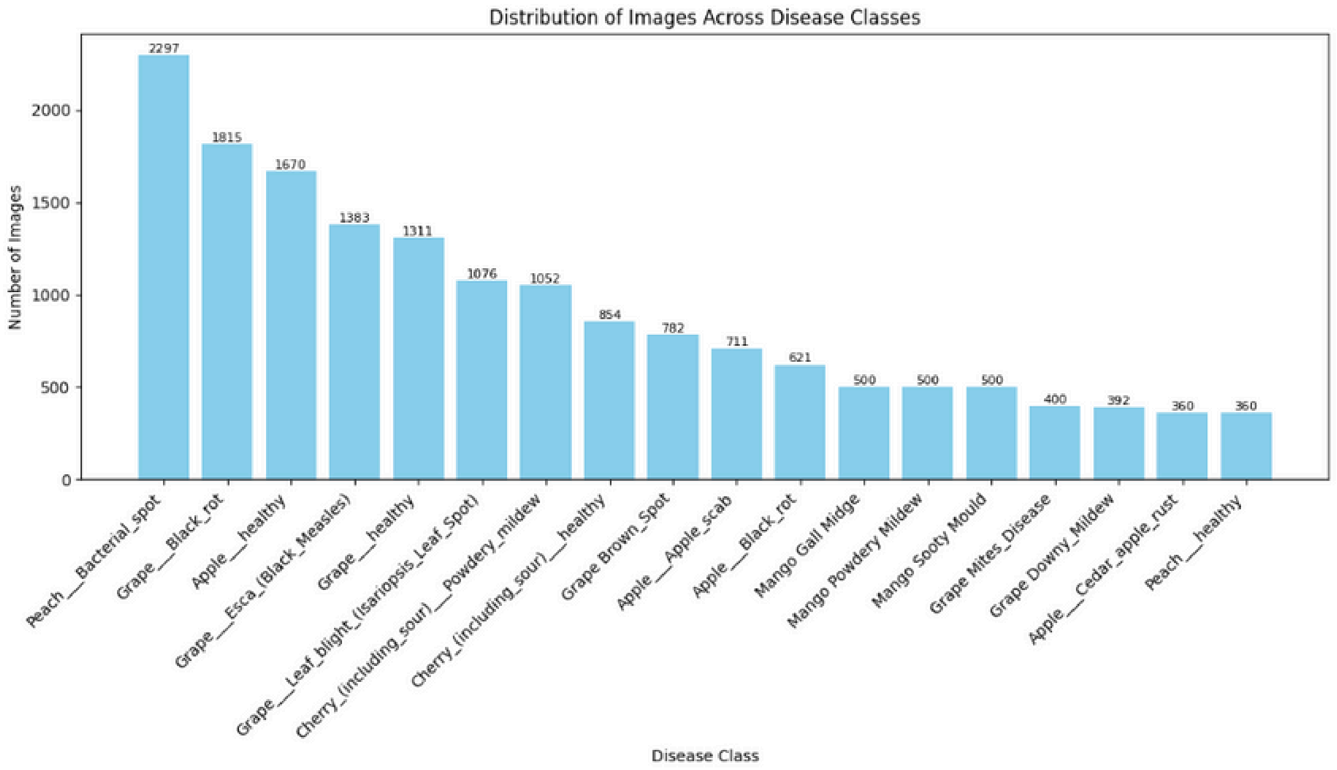
RESULTS

Traditional Computer Vision Approach’s Results

Due to significant class imbalance, overall Accuracy may be misleading — it can overrepresent performance on majority classes.

We prioritize F1-score as the main metric, since it balances Precision and Recall, giving a fairer view of performance, especially for rare but critical disease cases.

Other metrics like Precision, Recall, and Accuracy are also reported for completeness.



Class distribution of classes

RESULTS

Traditional Computer Vision Approach's Results

Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbors	65,46%	65,67%	65,46%	65,07%
LightGBM	**75,11%**	**74,53%**	**75,11%**	**74,67%**
Logistic Regression	73,41%	72,79%	73,41%	72,55%
Random Forest	71,86%	71,02%	71,86%	71,05%
SVM	72,98%	72,62%	72,98%	71,67%

Table 1: Results from using morphological, texture, shape features(%)

Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbors	66,34%	66,17%	66,34%	65,71%
LightGBM	**76,42%**	**76,24%**	**76,42%**	**76,07%**
Logistic Regression	72,04%	72,01%	72,04%	71,39%
Random Forest	73,62%	73,09%	73,62%	72,80%
SVM	69,60%	67,52%	69,60%	67,84%

Table 2: Results from only using Color features(%)

Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbors	51,05%	50,73%	51,05%	50,36%
LightGBM	**54,31%**	**53,34%**	**54,31%**	**53,41%**
Logistic Regression	44,08%	42,13%	44,08%	39,52%
Random Forest	52,85%	51,62%	52,85%	51,43%
SVM	51,69%	48,86%	51,69%	47,57%

Table 3: Results from Texture features(%)

Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbors	78,34%	78,50%	78,34%	78,03%
LightGBM	85,81%	85,34%	85,81%	85,46%
Logistic Regression	**87,24%**	**86,85%**	**87,24%**	**86,86%**
Random Forest	82,94%	82,54%	82,94%	82,60%
SVM	84,62%	84,12%	84,62%	83,95%

Table 4: Color, Texture, Morphological, Shape Features(%)

RESULTS

Deep features approach

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	**91,61%**	**91,45%**	**91,61%**	**91,50%**
SVM	91,67%	91,23%	91,67%	91,32%
Random Forest	88,18%	87,57%	88,18%	87,52%
K-Nearest Neighbors	90,30%	90,06%	90,30%	90,11%

Table 5: DenseNet deep features results (%)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	**90,49%**	**90,11%**	**90,49%**	**90,24%**
SVM	90,94%	90,53%	90,04%	90,22%
Random Forest	85,74%	84,94%	85,74%	84,93%
K-Nearest Neighbors	87,96%	87,63%	87,96%	87,71%

Table 6: EfficientNet deep features results (%)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	89,82%	89,74%	89,82%	89,76%
SVM	**90,94%**	**90,40%**	**90,94%**	**90,40%**
Random Forest	86,50%	86,03%	86,50%	85,79%
K-Nearest Neighbors	88,24%	87,89%	88,24%	87,98%

Table 7: Resnet deep features results (%)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	**87,36%**	**86,81%**	**87,36%**	**86,95%**
SVM	87,17%	85,68%	87,17%	86,23%
Random Forest	81,64%	80,67%	81,64%	80,41%
K-Nearest Neighbors	84,98%	84,46%	84,98%	84,57%

Table 8: VGG deep features results (%)

RESULTS

Deep Learning-based

Model	Accuracy	Precision	Recall	F1 Score
ResNet-50	92,22%	96,32%	94,79%	92,42%
ResNet-101	91,77%	91,92%	92,58%	91,42%
ResNet-50v2	92,39%	92,80%	90,98%	91,88%
Densenet-201	**95,70%**	**96,32%**	**94,79%**	**95,55%**
EfficientNet-B0	92,37%	93,13%	92,02%	92,57%
ViT	92,29%	92,21%	90,94%	91,57%
SwinTransformer	92,09%	92,57%	91,30%	91,93%

Table 9: Deep Learning-based result(%)

RESULTS

Classification report

	precision	recall	f1-score	support
Apple___Apple_scab	1.0000	0.9859	0.9929	71
Apple___Black_rot	1.0000	1.0000	1.0000	62
Apple___Cedar_apple_rust	1.0000	1.0000	1.0000	36
Apple___healthy	0.9882	1.0000	0.9940	167
Cherry_(including_sour)___Powdery_mildew	1.0000	1.0000	1.0000	105
Cherry_(including_sour)___healthy	1.0000	0.9882	0.9941	85
Grape Brown_Spot	0.8507	0.7308	0.7862	78
Grape Downy_Mildew	0.4255	0.5128	0.4651	39
Grape Mites_Disease	0.7381	0.7750	0.7561	40
Grape___Black_rot	0.9141	0.8187	0.8638	182
Grape___Esca_(Black_Measles)	1.0000	0.9784	0.9891	139
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.0000	1.0000	1.0000	108
Grape___healthy	0.8377	0.9847	0.9053	131
Mango Gall Midge	1.0000	1.0000	1.0000	50
Mango Powdery Mildew	1.0000	1.0000	1.0000	50
Mango Sooty Mould	1.0000	1.0000	1.0000	50
Peach___Bacterial_spot	1.0000	1.0000	1.0000	230
Peach___healthy	1.0000	1.0000	1.0000	36
accuracy			0.9464	1659
macro avg	0.9308	0.9319	0.9304	1659
weighted avg	0.9497	0.9464	0.9469	1659

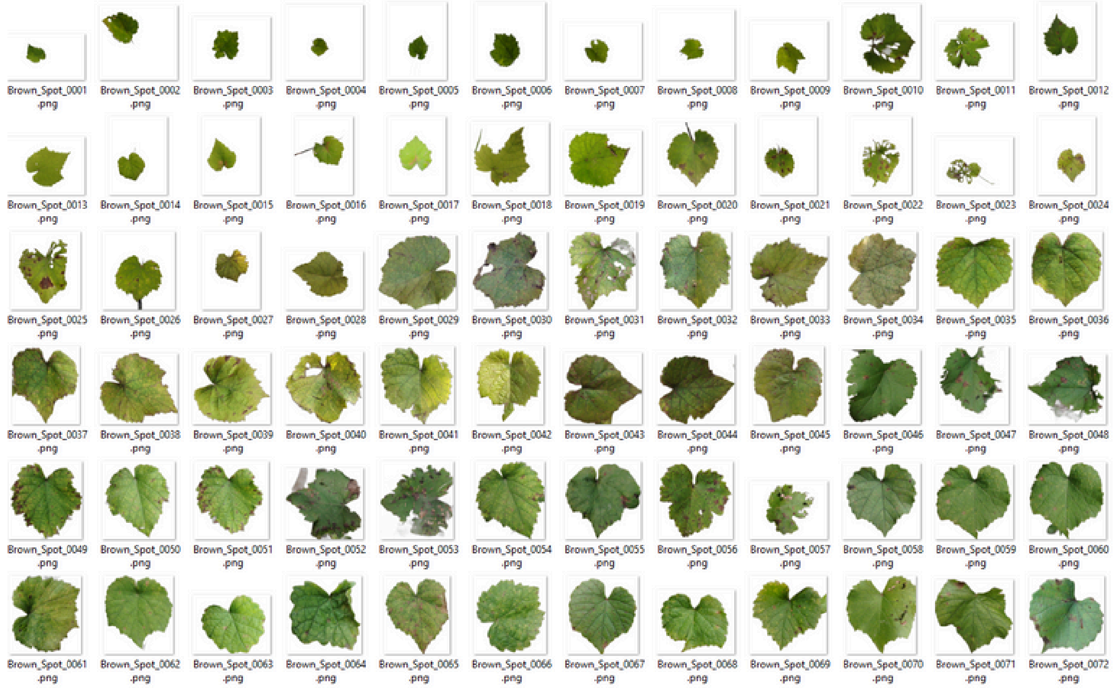
Table 10: Classification report

RESULTS

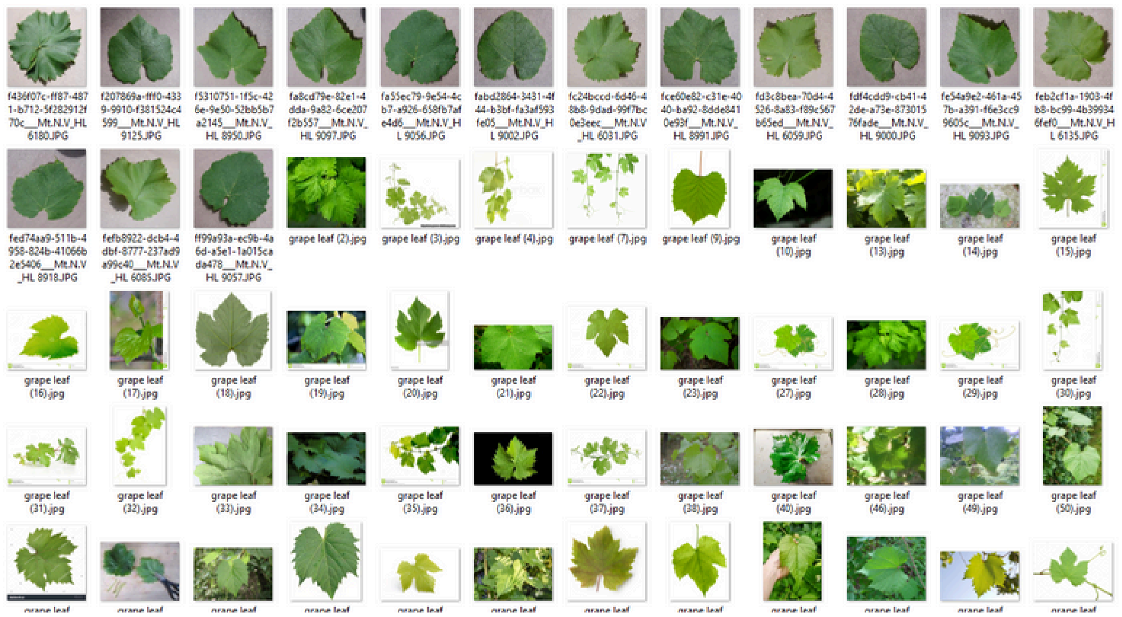
Classification report

Grape Mites_Disease (F1 = 0.75)

Grape Downy_Mildew (F1 = 0.46)



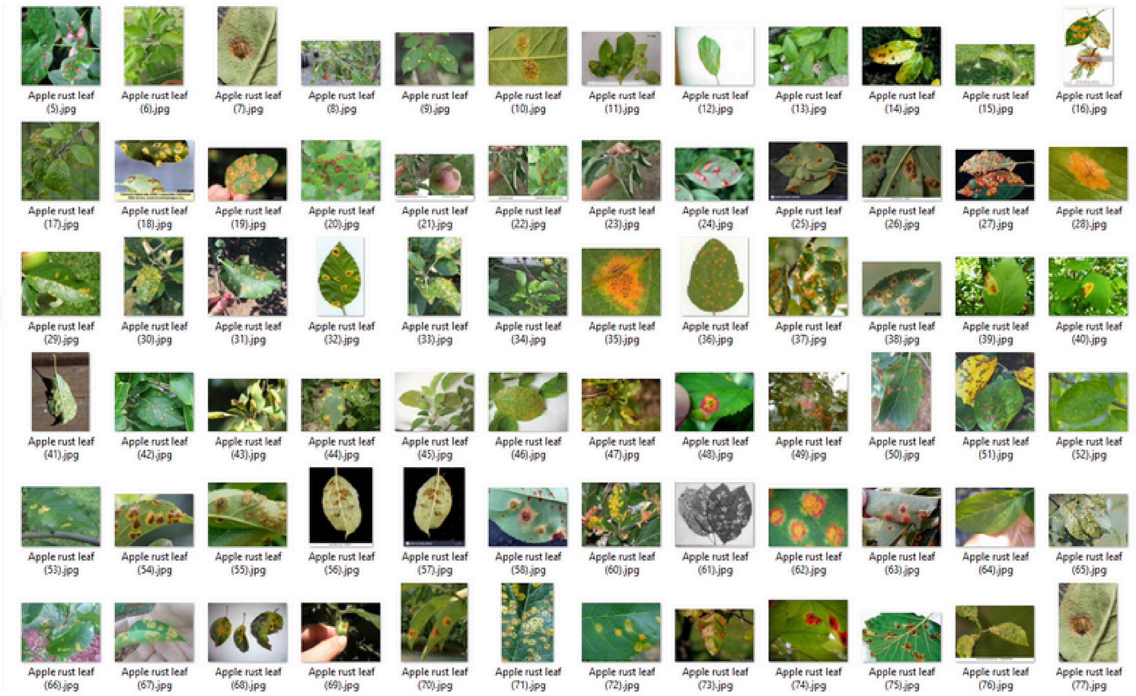
Grape Brown_Spot (F1 = 0.78)



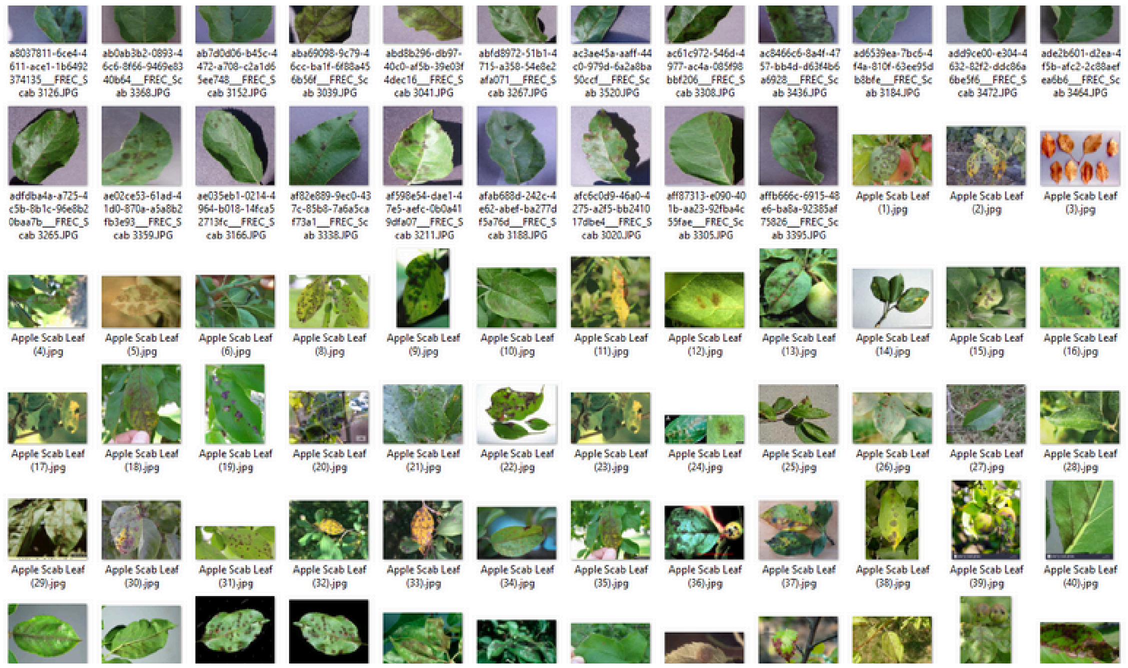
Grape healthy (F1 = 0.9)

RESULTS

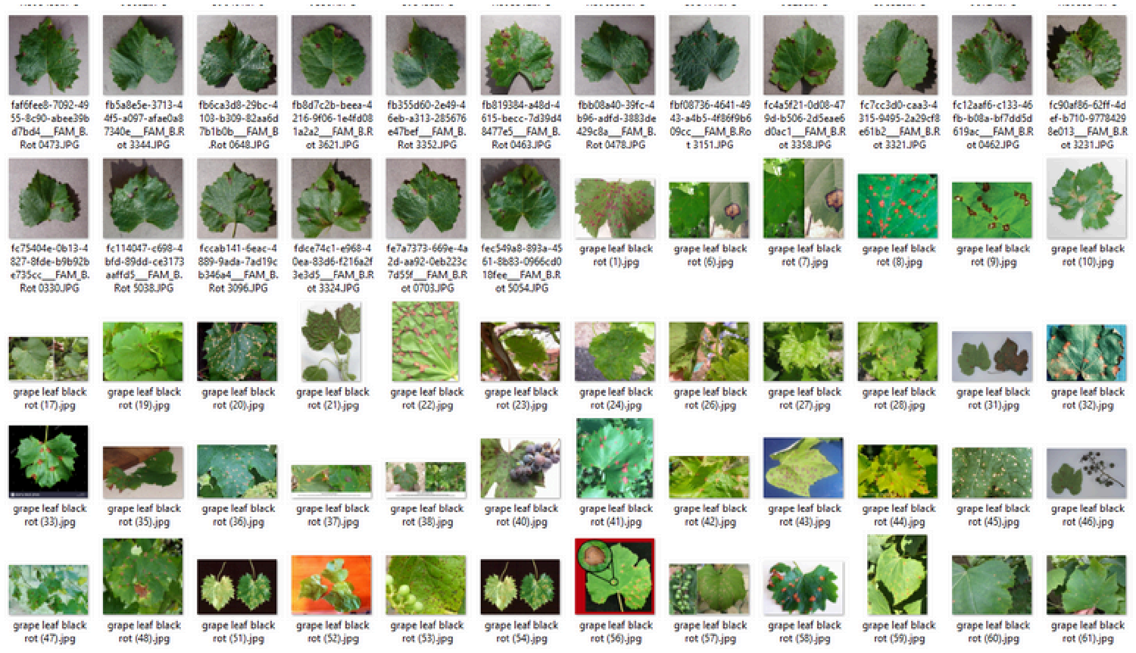
Classification report



Cedar Apple Rust (F1 = 1)



Apple scab (F1= 0.99)



Grape blackrot (F1 = 0.86)

Apple healthy (F1= 0.99)

Q&A

THANK YOU