Tree Species Classification

1-Project Overview

This project aims to classify tree species based on leaf images using deep learning.
The dataset used is the Flavia Leaf Dataset, which includes 32 plant species with clean, high-quality leaf images.
Five different deep learning architectures were trained and evaluated:

1. VGG19
2. ResNet50
3. Inception v1 (GoogLeNet)
4. Vision Transformer (ViT-B16)

Each model was evaluated using checkpoint 15, and all results, metrics, and visualizations were saved automatically.

2- Dataset Description – Flavia Leaf Dataset

- Contains 32 leaf categories.
- Each category includes approximately 50–70 images.
- Controlled lighting and white background.
- Suitable for image classification and pattern recognition tasks.
- Link: https://flavia.sourceforge.net

Preprocessing Steps

1. Dataset Organization

- Raw data located in data/Leaves/.
- Organized into class-specific folders in data/flavia/.

2. **Image Cleaning & Standardization**
   - All images converted to RGB to standardize color channels.
3. **Image Resizing**
   - Images resized to 224×224 pixels to ensure compatibility with deep learning models.
4. **Class Assignment**
   - Each image assigned to a class based on filename ranges defined in LABEL_RANGES.
   - Unmatched files recorded and excluded from the dataset.
5. **Train/Validation/Test Split (Stratified Split)**
   - Train: 70% of samples
   - Validation: 15% of samples
   - Test: 15% of samples
   - Split ensures proportional representation of all classes in each subset.
6. **Offline Data Augmentation (Training Set Balancing)**

- **Minority classes augmented to match the majority class count.**
- **Transformations include:**
  - **Random Horizontal Flip (p=0.5)**
  - **Random Rotation (±15°)**
  - **Color Jitter (brightness, contrast, saturation, hue adjustments)**

7. **Online Data Augmentation (During Training)**
   - **Additional transformations can be applied dynamically during model training to improve generalization and reduce overfitting.**
8. **Augmentation Demonstration**
   - **Grid of images created showing original + 4 augmented versions to visually validate augmentation effectiveness.**

3-Deep Learning Model Architectures

A)VGG19:

A deep CNN architecture built from stacked 3×3 convolutional layers followed by max-pooling layers and large fully connected layers.

Pros

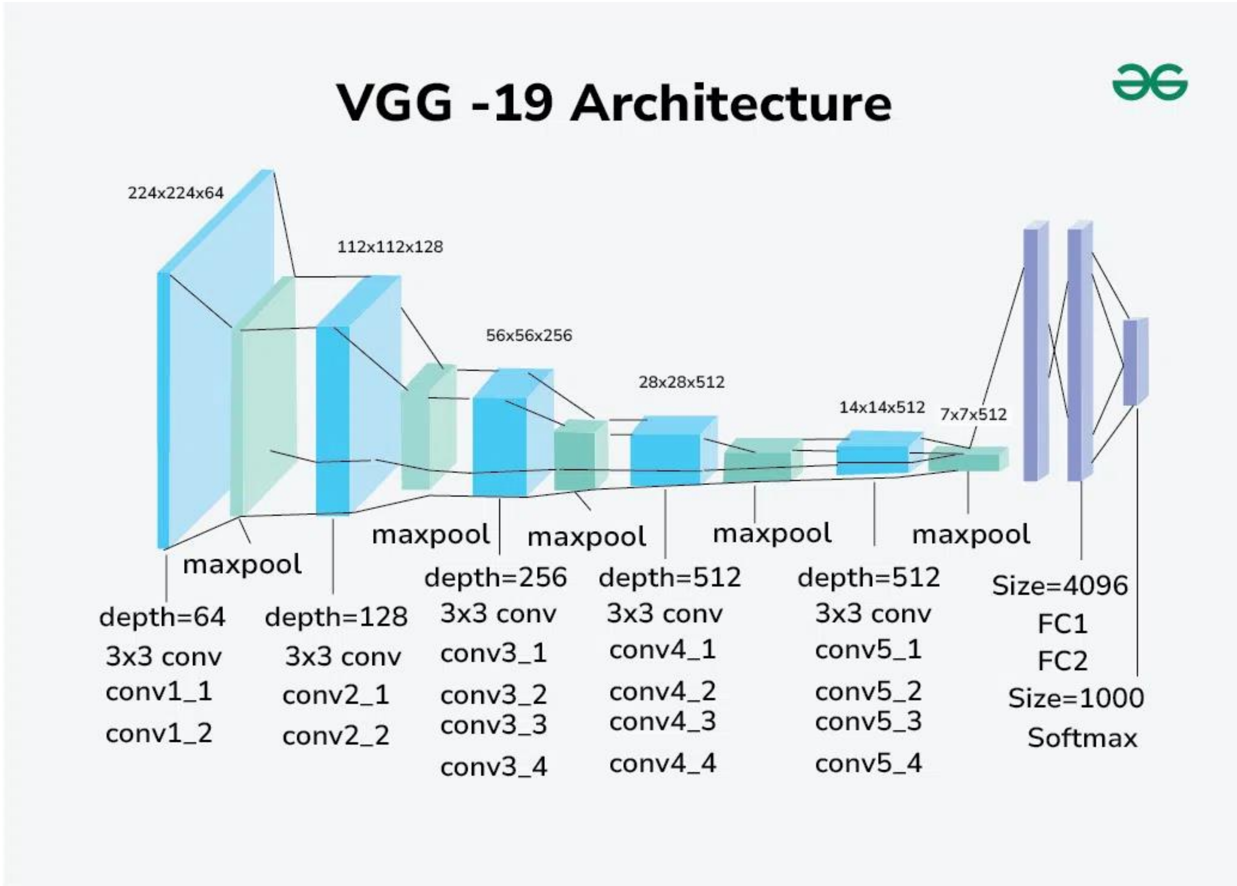
- **Simple and easy to implement**
- **Performs well for medium-sized datasets**
- **Strong baseline architecture**

Cons

- **Very large number of parameters**
- **Slow to train**
- **Prone to overfitting on small datasets**

Reference

Simonyan & Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition* (2015)



B) ResNet50

Introduces Residual Blocks with skip connections to allow training of very deep networks without gradient vanishing.

Pros

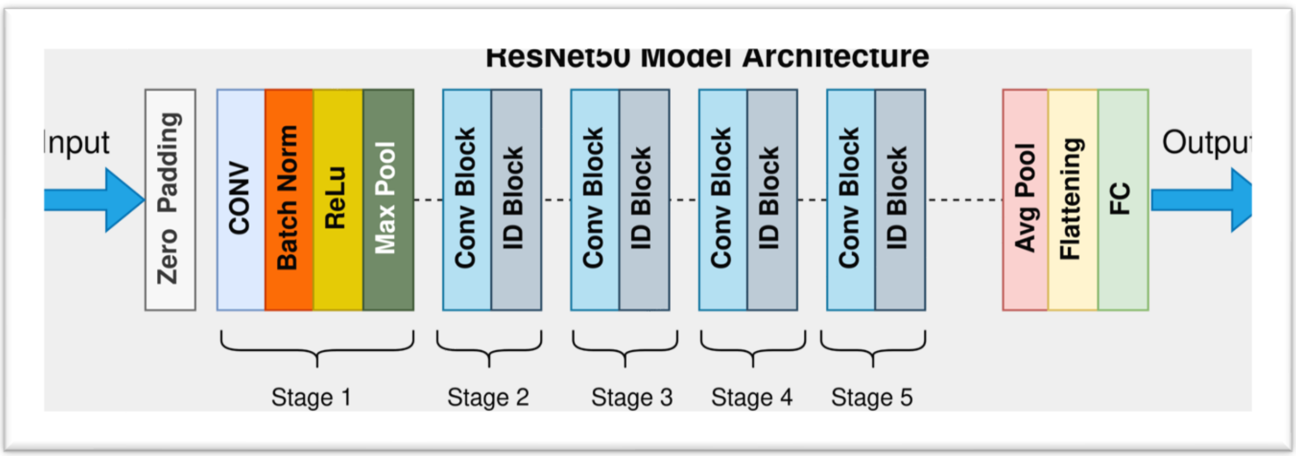
- **Excellent accuracy**
- **Trains efficiently even at large depth**
- **Strong generalization**

Cons

- **More complex architecture**
- **Heavier than MobileNet**

Reference

He et al., *Deep Residual Learning for Image Recognition* (2015)



C) Inception v1 (GoogLeNet)

Uses Inception Modules, combining multiple filter sizes (1×1, 3×3, 5×5) in parallel to capture multi-scale features.

Pros

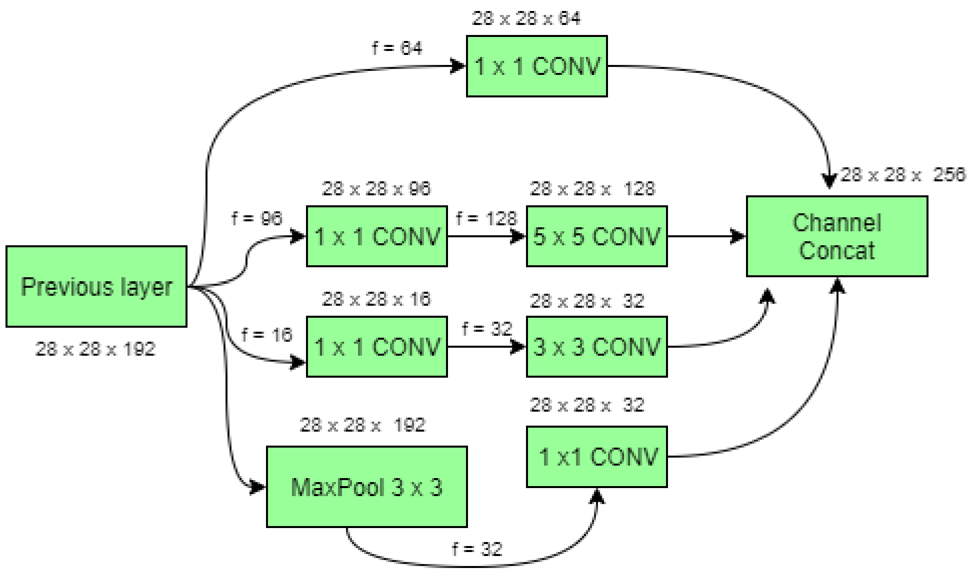
- **Very efficient**
- **Fewer parameters than VGG**
- **Excellent at analyzing fine texture details**

Cons

- **More complex to implement**
- **Harder to modify**

Reference

Szegedy et al., ***Going Deeper with Convolutions*** **(2015)**



D) Vision Transformer (ViT-B16)

A transformer-based vision model that splits images into patches and processes them like sequences in NLP.

Pros

- **Strong global feature extraction**
- **Excellent performance on large datasets**
- **Captures long-range dependencies**

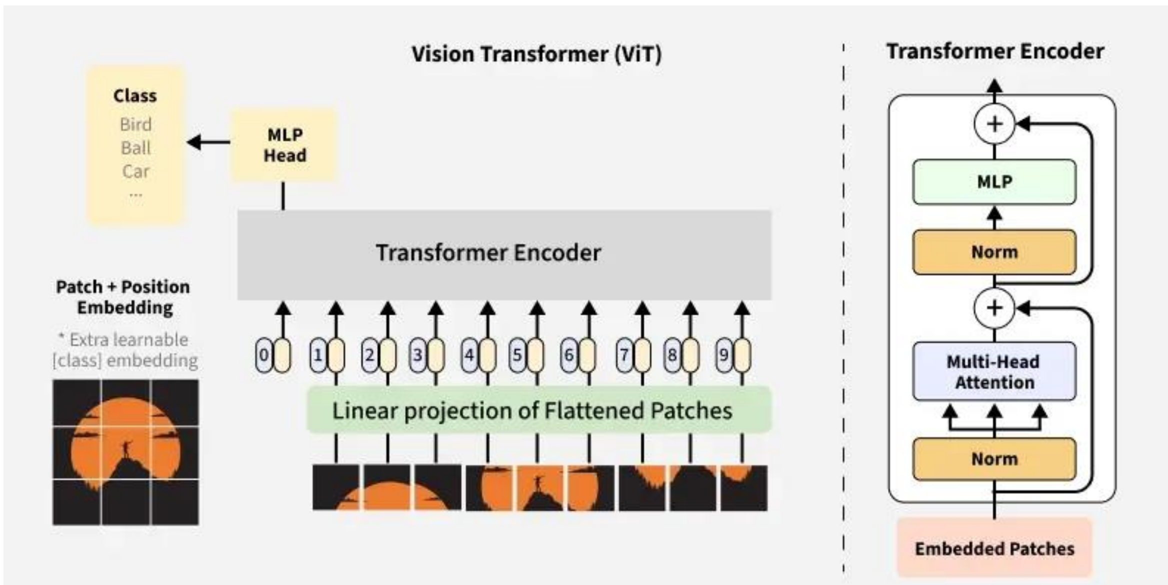
Cons

- **Requires large training data**

- Sensitive to overfitting on small datasets (like Flavia)

Reference

Dosovitskiy et al., *An Image is Worth 16×16 Words* (2021)



4-Evaluation Metrics Explained

All metrics were calculated both per-class and overall, providing a detailed performance breakdown..

A) Accuracy

Proportion of correctly classified samples.

B) Precision

How many predicted labels are correct.

Good for imbalanced datasets.

C) Recall

How many true samples are correctly detected.

Important when missing a class is costly

D) F1-Score

Harmonic mean of precision and recall.

E) Confusion Matrix

Shows how predictions compare to true labels.

F) ROC Curve

Plots True Positive Rate vs. False Positive Rate.

G) AUC Score

Measures the area under the ROC curve.

AUC close to 1.0 → excellent classifier.

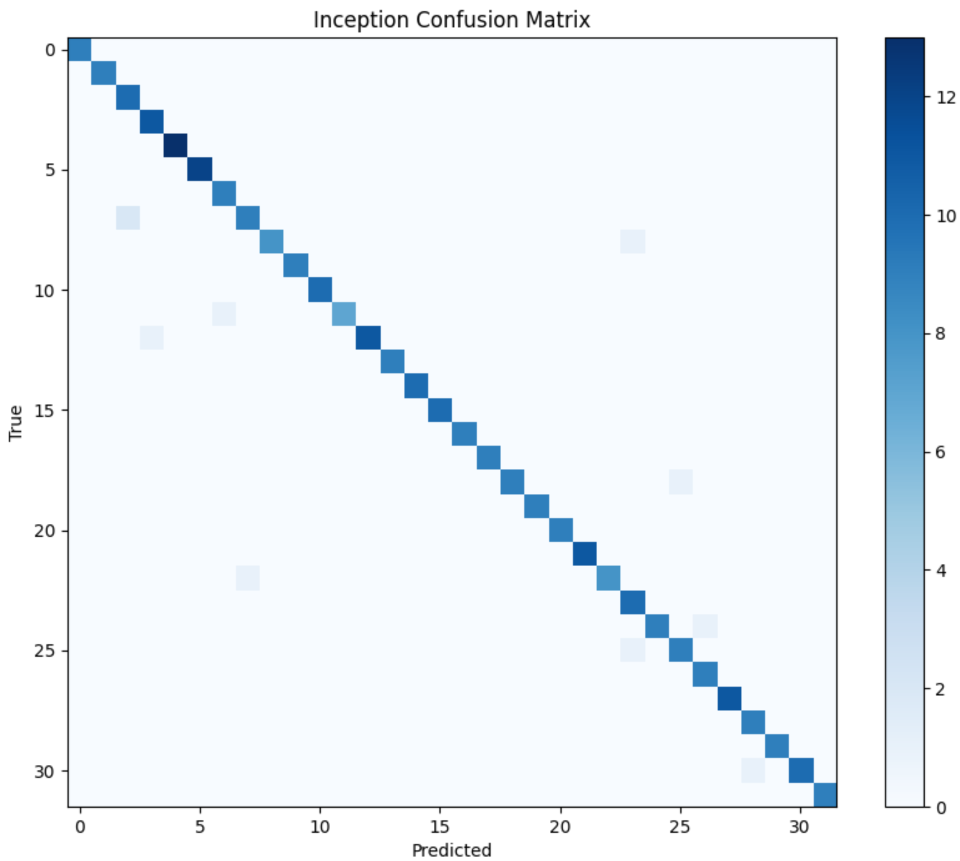
5-Result Visualization (All Saved Automatically)

For each model, the following graphs are saved in:

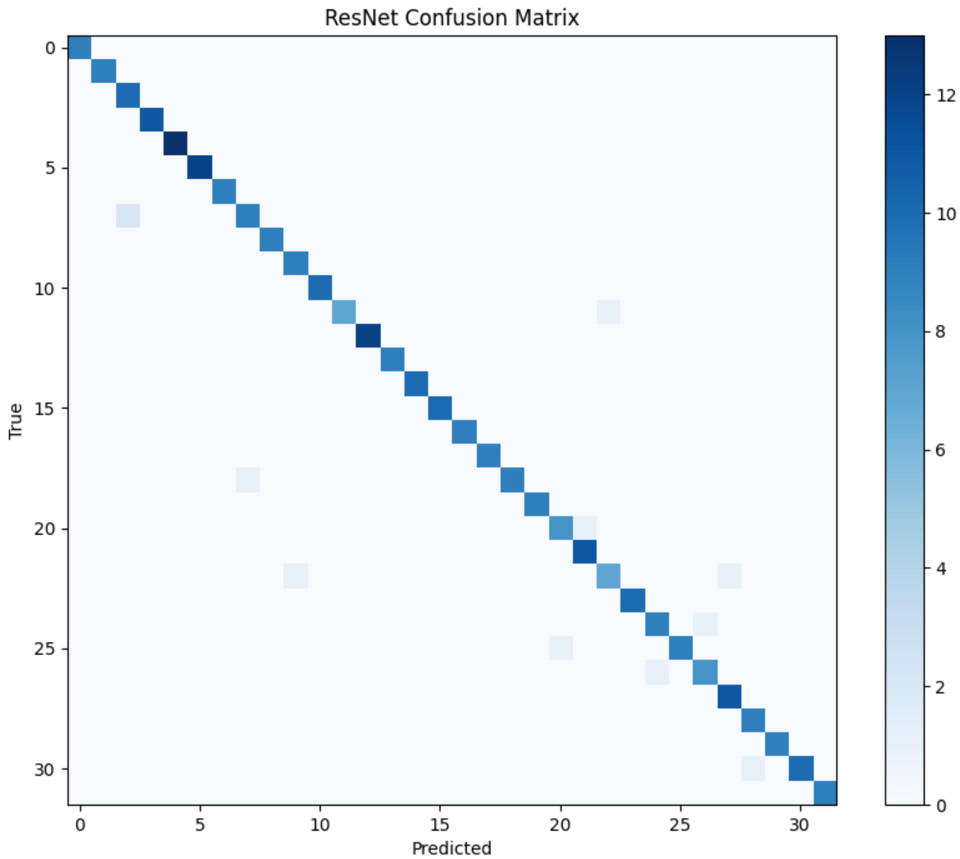
(outputs/<model_name>/evaluation/)

1. **Confusion Matrix**
   - **Shows misclassifications and confusion between similar leaves**

     Inception Confusion Matrix

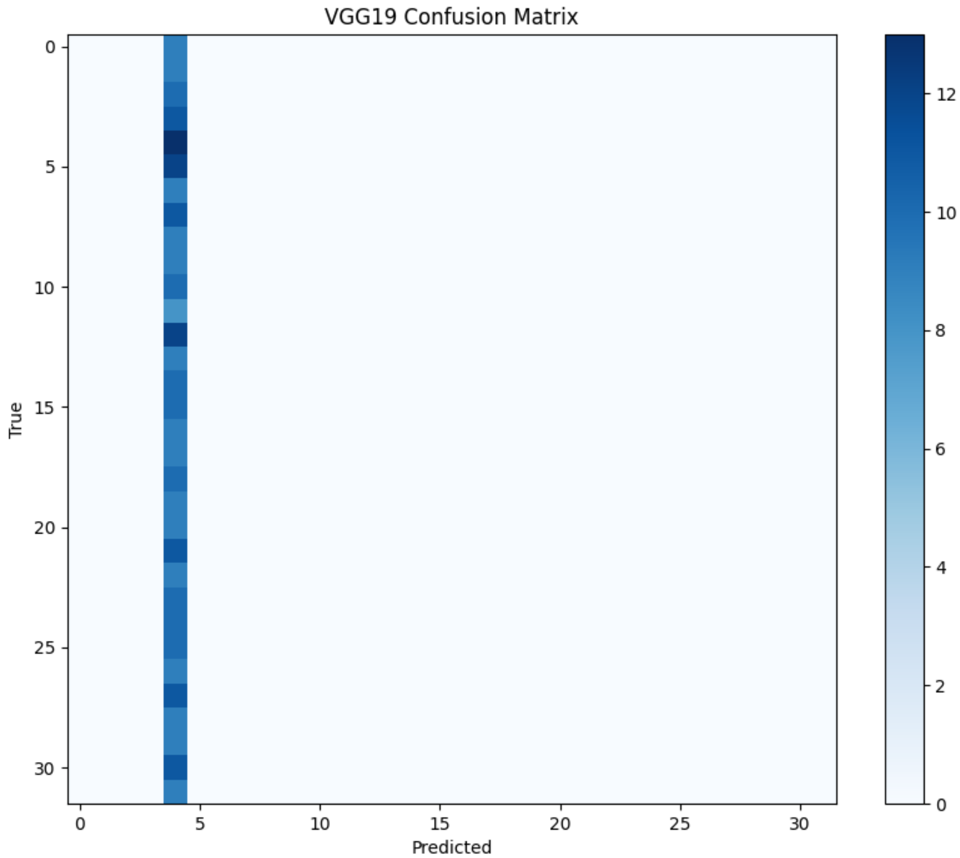


   -
   - **ResNet Confusion Matrix**

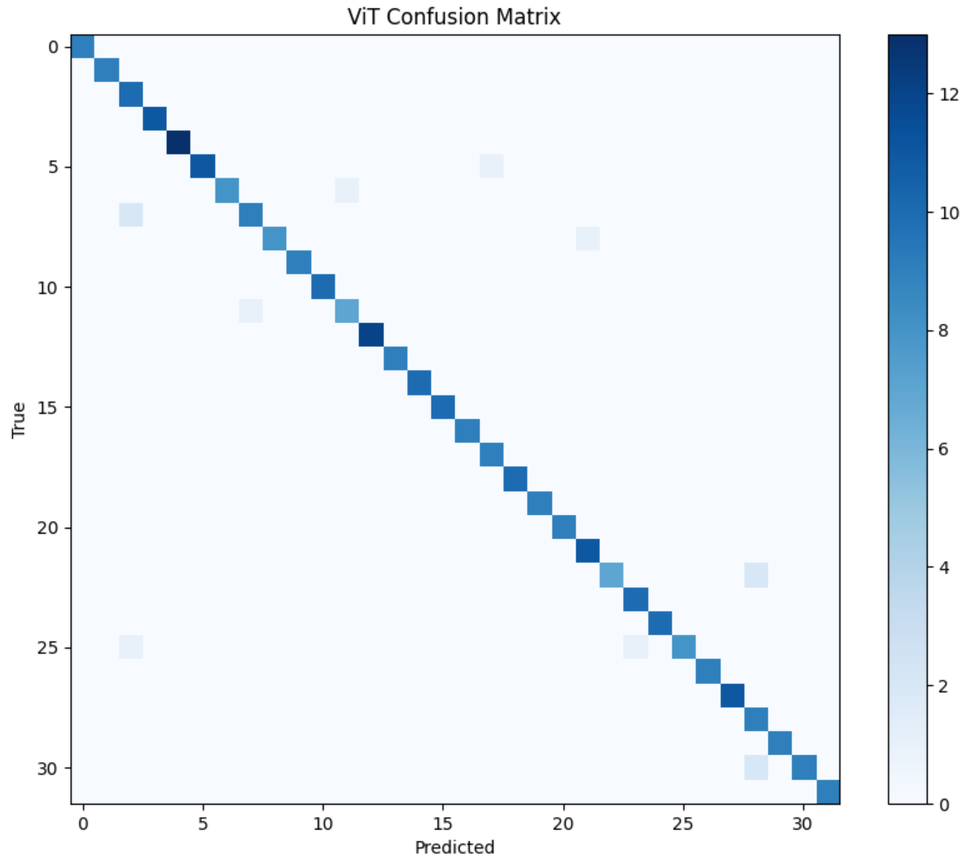

ResNet Confusion Matrix
True
Predicted
0
5
10
15
20
25
30
0
2
4
6
8
10
12

-

- **VGG19 Confusion Matrix**

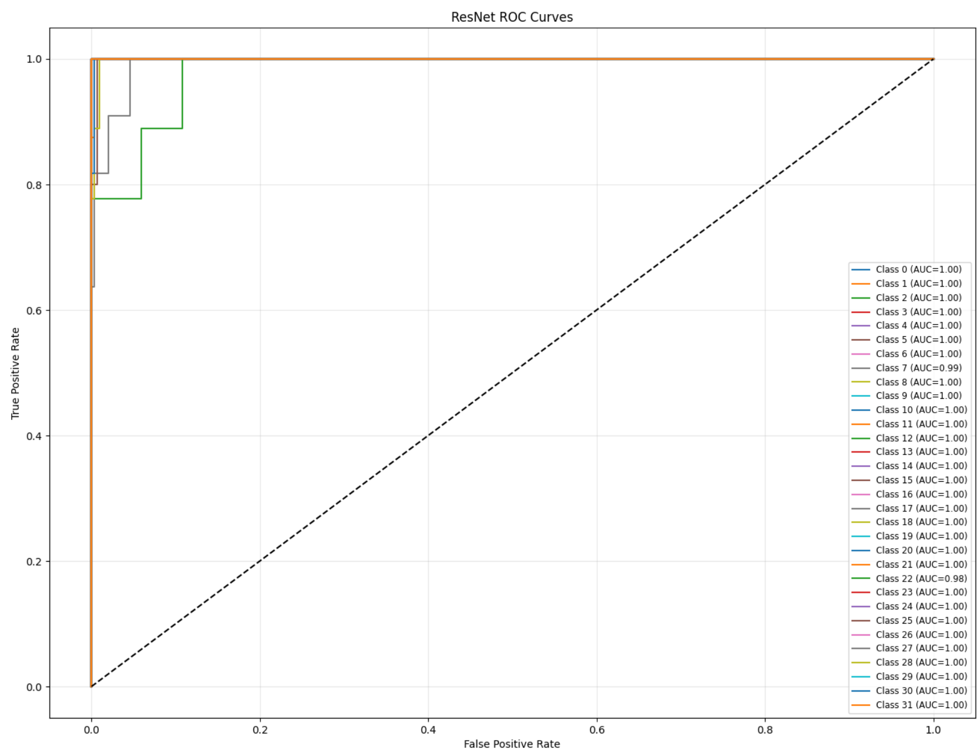


- **VIT Confusion Matrix**

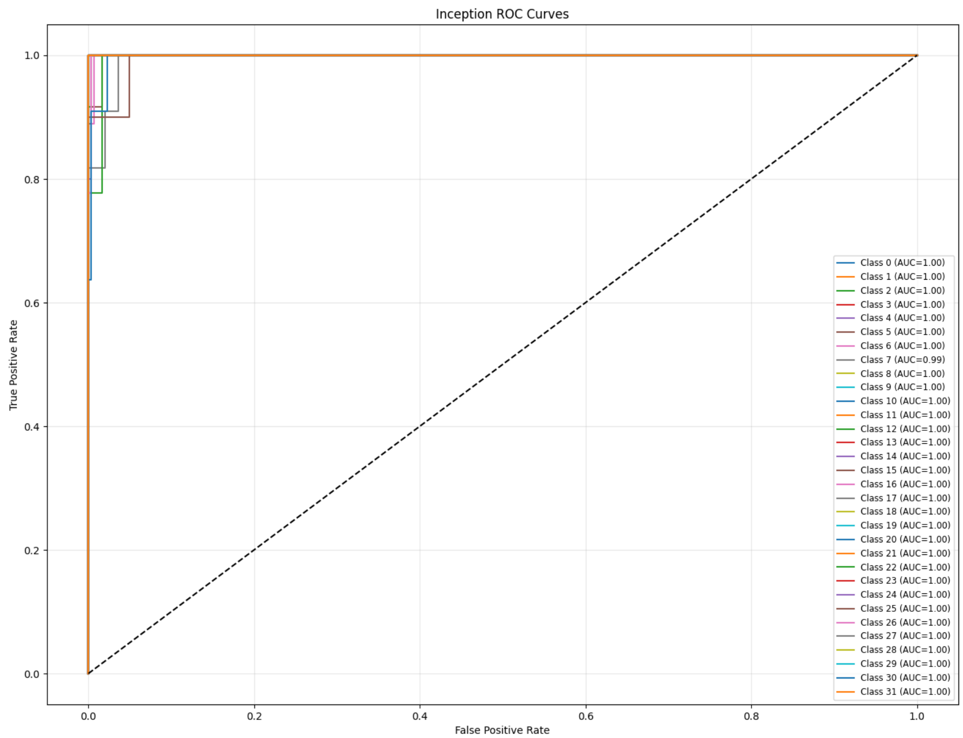


2-ROC Curves (per class)

- **Demonstrates sensitivity at various thresholds**
  - **ResNet ROC Curves**

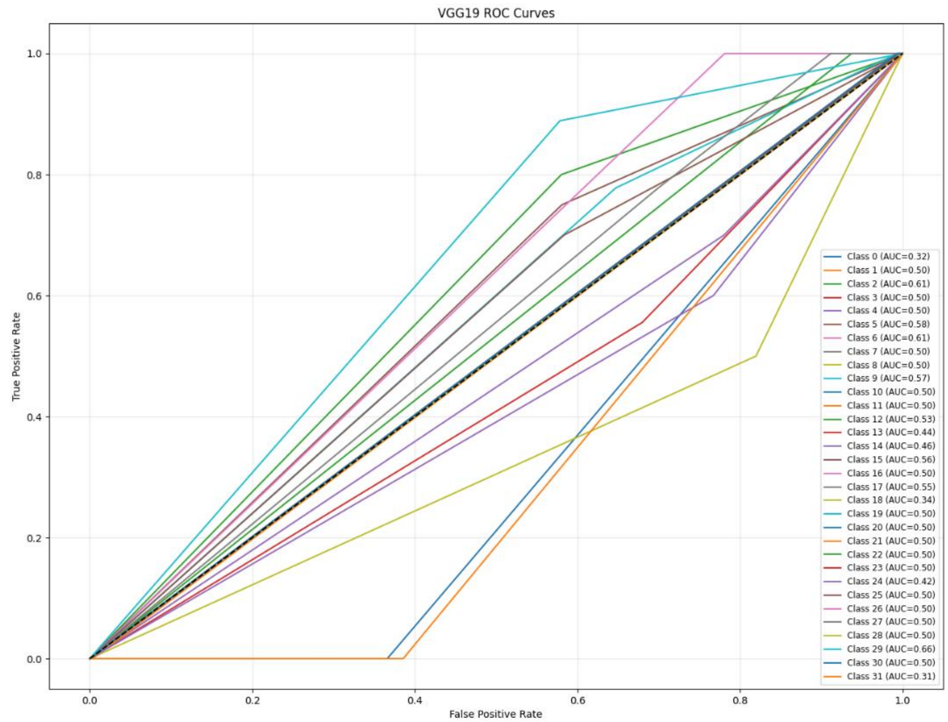


  - **Inception ROC Curves**

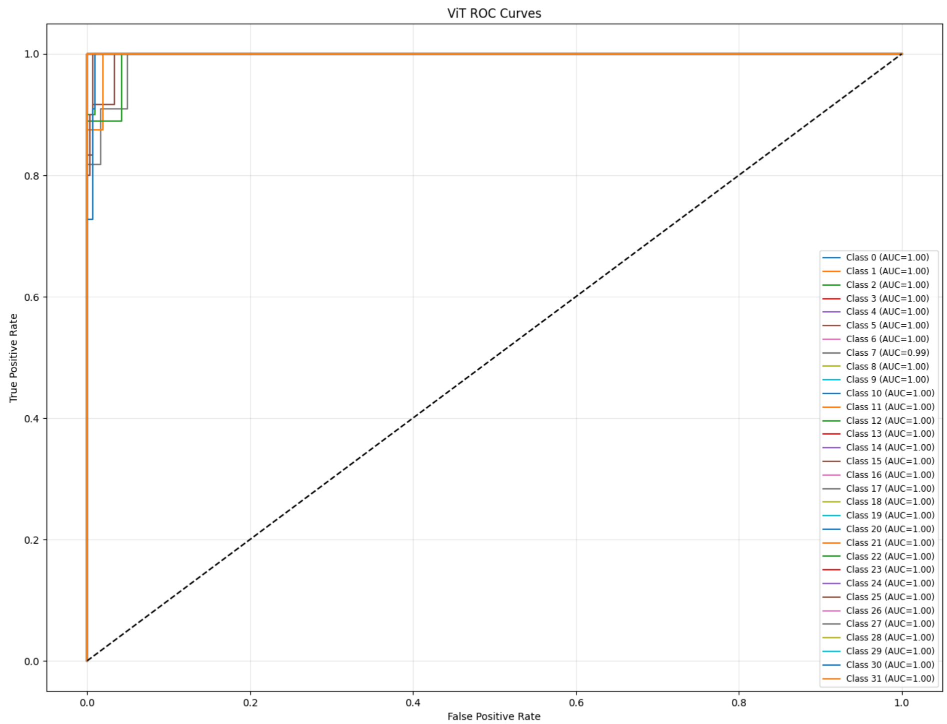


Inception ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC=1.00)
Class 1 (AUC=1.00)
Class 2 (AUC=1.00)
Class 3 (AUC=1.00)
Class 4 (AUC=1.00)
Class 5 (AUC=1.00)
Class 6 (AUC=1.00)
Class 7 (AUC=0.99)
Class 8 (AUC=1.00)
Class 9 (AUC=1.00)
Class 10 (AUC=1.00)
Class 11 (AUC=1.00)
Class 12 (AUC=1.00)
Class 13 (AUC=1.00)
Class 14 (AUC=1.00)
Class 15 (AUC=1.00)
Class 16 (AUC=1.00)
Class 17 (AUC=1.00)
Class 18 (AUC=1.00)
Class 19 (AUC=1.00)
Class 20 (AUC=1.00)
Class 21 (AUC=1.00)
Class 22 (AUC=1.00)
Class 23 (AUC=1.00)
Class 24 (AUC=1.00)
Class 25 (AUC=1.00)
Class 26 (AUC=1.00)
Class 27 (AUC=1.00)
Class 28 (AUC=1.00)
Class 29 (AUC=1.00)
Class 30 (AUC=1.00)
Class 31 (AUC=1.00)

- ## VGG19 ROC Curves

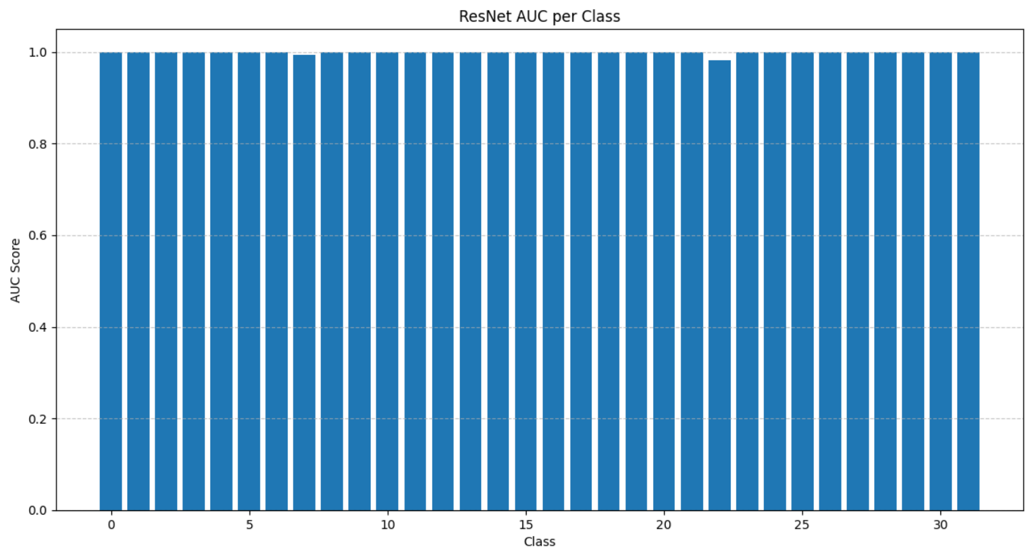


- ### VIT ROC Curves

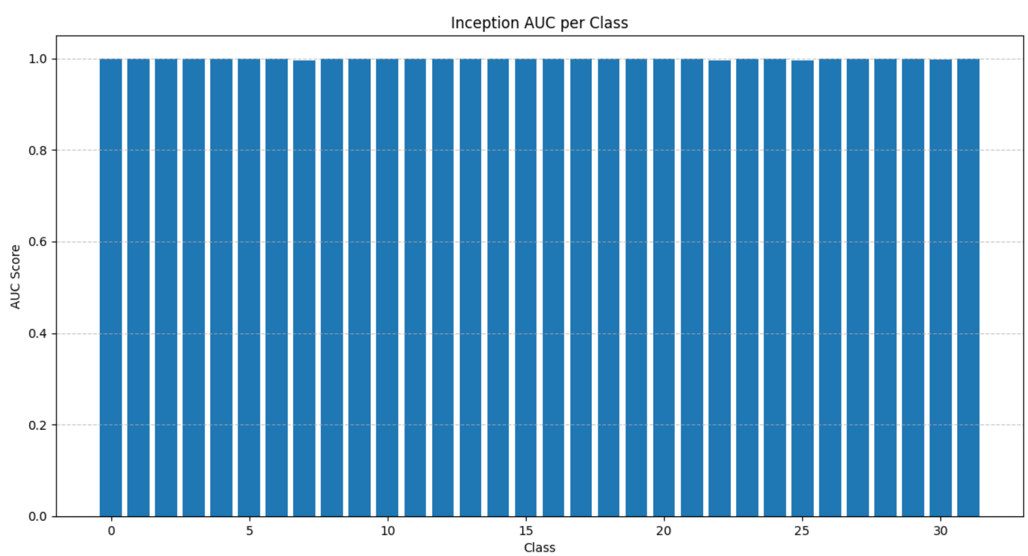


3. AUC Bar Chart (per class)

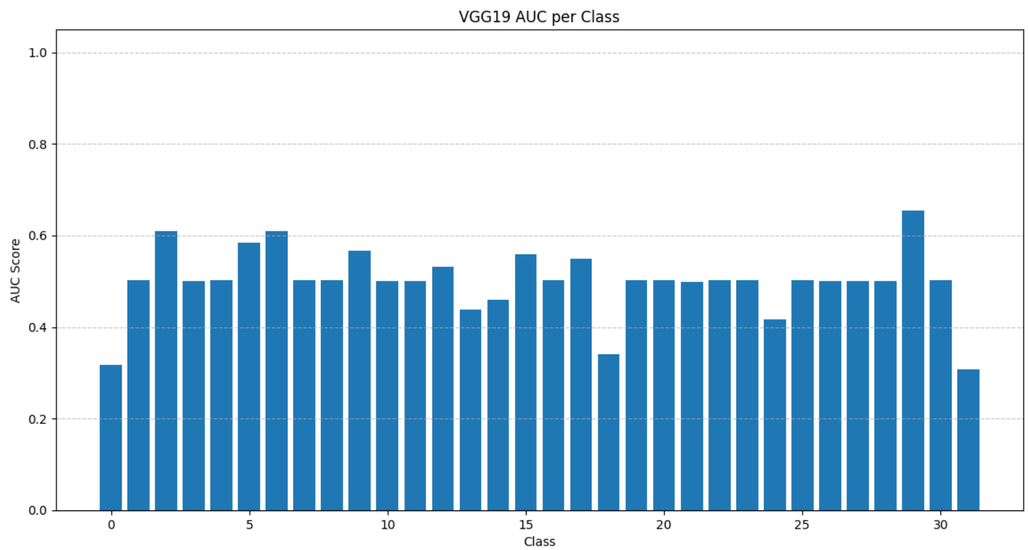
- **Useful to see which classes are harder**
  - **ResNet AUC Bar**



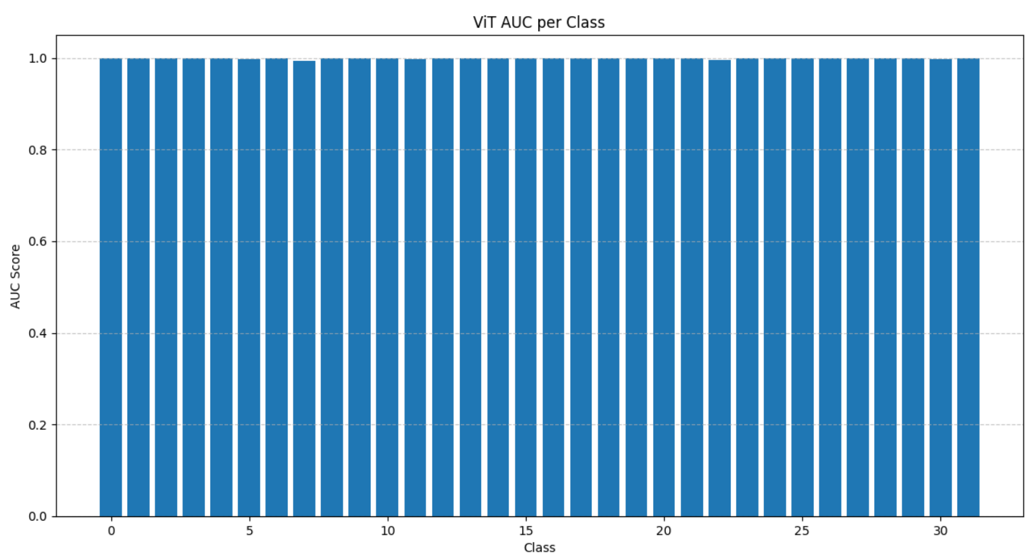
  - **Inception AUC Bar**



- ### VGG19 AUC Bar

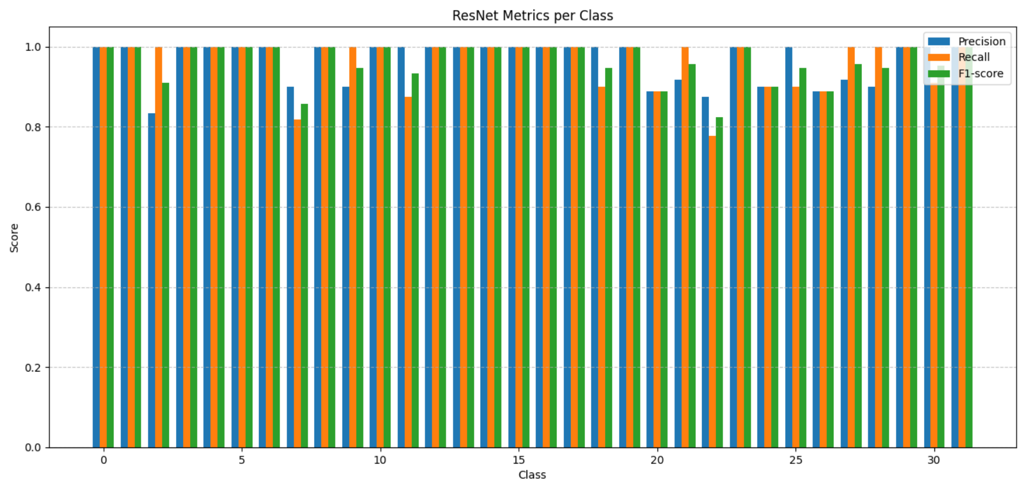


- ### VIT AUC Bar

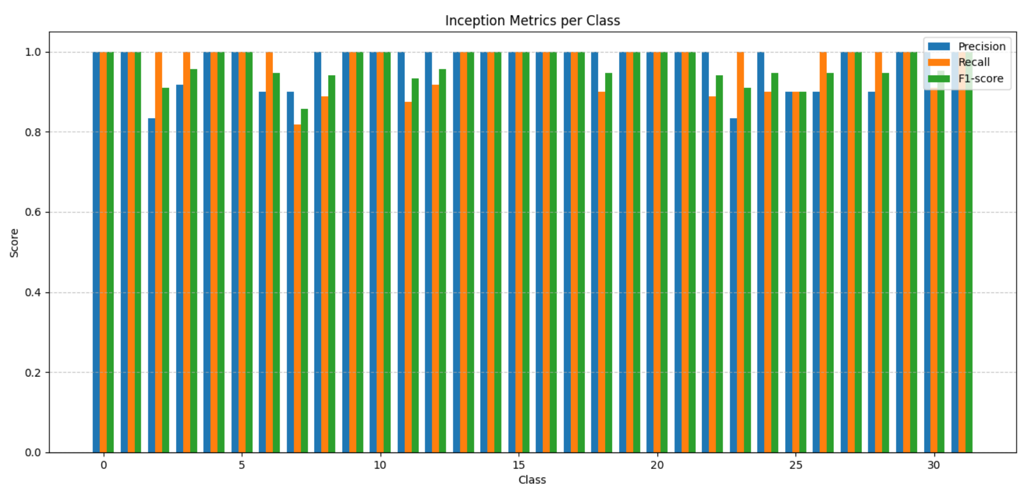


4. Per-Class Metrics Bar Plot

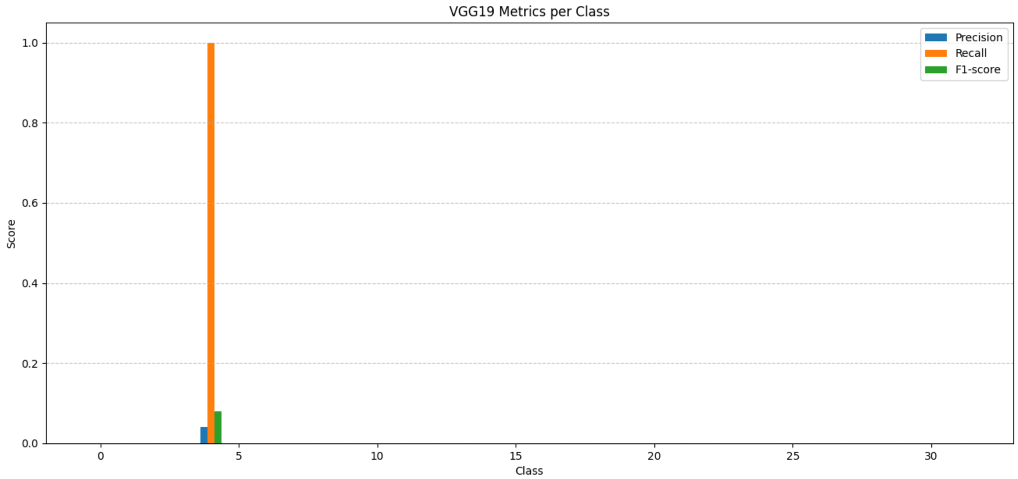
- **Precision, Recall, F1 for each class**
  - **ResNet Metrics per Class**



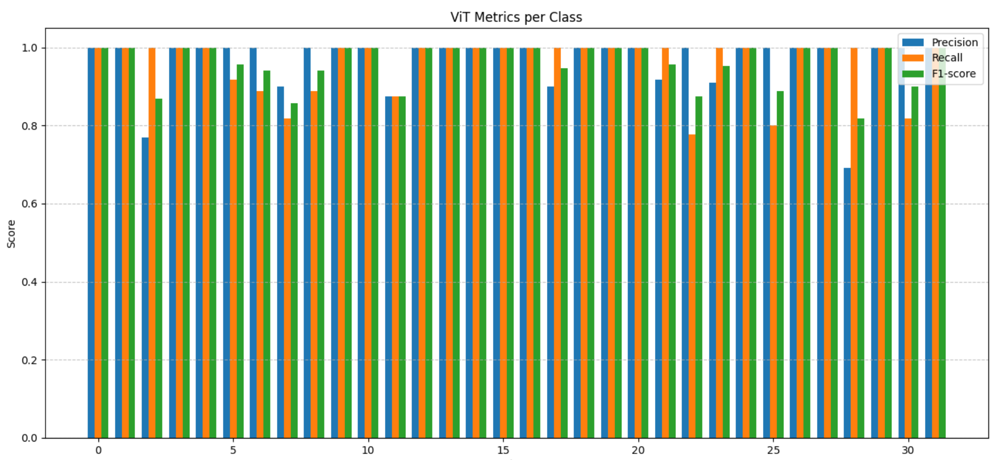
  - **Inception Metrics per Class**



- ### VGG19 Metrics per Class

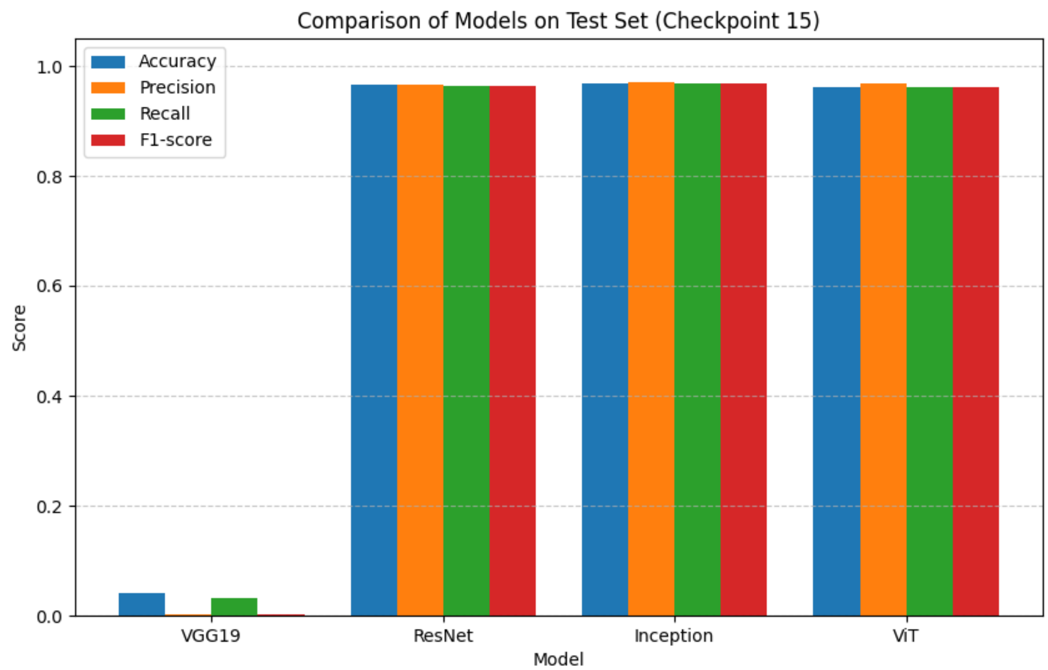


- ### VIT Metrics per Class



5. Model Comparison Chart

(File: evaluation/comparison_all_models.png)



6-Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-score	Notes
Inception v1	0.9683	0.9714	0.9686	0.9685	Best overall performance, excellent multi-scale feature extraction
ResNet50	0.9651	0.9662	0.9643	0.9642	Very strong, robust residual learning, stable results
ViT	0.9619	0.9676	0.9620	0.9618	Performs well but Transformer models need larger datasets
VGG19	0.0413	0.0013	0.0312	0.0025	Completely failed training (likely wrong preprocessing or last layer mismatch)

7-Why Certain Models Perform Better?

ResNet50 performs best because:

- **Residual learning helps capture complex patterns**
- **Deep but stable training**
- **Excellent at capturing textures and shapes in leaf veins**

Inception performs well because:

- **Multi-scale feature extraction fits leaf texture variations**
- **Efficient architecture with strong generalization**

ViT performs average because:

- **Transformers require large datasets to avoid overfitting**
- **Flavia dataset (~1900 images) is relatively small**

VGG19 is limited due to:

- **Extremely large number of parameters**
- **Easily overfits small datasets**

VGG-19 Pre-trained:

Model	Accuracy	Precision	Recall	F1-score	Notes
Inception v1	0.9651	0.9686	0.9662	0.9653	Best overall performance, excellent multi-scale feature extraction
ResNet50	0.9714	0.9726	0.9717	0.9709	Very strong, robust residual learning, stable results
ViT	0.9619	0.9676	0.9620	0.9618	Performs well but Transformer models need larger datasets
VGG19	0.0413	0.0013	0.0312	0.0025	Completely failed training (likely wrong preprocessing or last layer mismatch)
VGG-19 Pre-trained	0.9905	0.9914	0.9891	0.9897	Excellent feature extractor; best for leaf classification