Comparative Analysis of Various Architectures for Multiclass Image Classification

Authors:

- **Anirudh Pal(22040)** (Lead: Baseline Random Forest)
- **Gargi Shirpurkar Pande(22127)** (Lead: Custom CNN)
- **Vipul Sharma(22369)** (Lead: Hybrid CNN + RF)
- **Sundram Anand Pandita(22342)** (Lead: ResNet50 Transfer Learning)

1. Abstract

This report benchmarks four distinct machine learning architectures on the "Animals-10" dataset to evaluate the efficacy of classical versus deep learning approaches. We compare a baseline Random Forest (RF) trained on raw pixels, a custom Convolutional Neural Network (CNN), a Hybrid CNN-RF model, and a pre-trained ResNet50 via Transfer Learning. Results indicate a significant performance hierarchy: the Baseline RF achieves ~40% accuracy due to a lack of spatial invariance, while the Hybrid and Custom CNN models achieve ~73-74% by leveraging feature extraction. The ResNet50 model achieves state-of-the-art performance with 92.5% accuracy, demonstrating the immense value of Transfer Learning for limited datasets.

2. Introduction

The project aims to classify images into 10 distinct animal categories (labeled in Italian: *Cane, Gatto, Ragno, Farfalla, Gallina, Cavallo, Elefante, Mucca, Pecora, Scoiattolo*).

The core objective is to analyze the "Evolution of Image Classification" by implementing and comparing four progressively complex architectures. We specifically investigate why classical models fail on raw image data and how deep learning architectures resolve these limitations through hierarchical feature extraction.

3. Methodology

We implemented four distinct modeling approaches, ranging from classical baselines to state-of-the-art transfer learning.

3.1. Baseline: Random Forest (Lead: Anirudh Pal)

- **Approach:** We established a classical machine learning baseline using a Random Forest Classifier configured with an ensemble of 100 decision trees. To adapt the image data for this algorithm, the 3-channel RGB images (64 x 64 x 3) were subjected to a flattening transformation, converting them into high-dimensional 1D feature vectors of size 12,288.

- **Hypothesis:** The primary objective was to empirically test the limits of "shallow" learning. We hypothesized that while pixel intensity correlations might identify simple patterns (e.g., a green background implies "grass"), they would likely be insufficient for distinguishing complex objects without specific feature extraction mechanisms.

- **Limitation:** The fundamental constraint of this approach is the loss of **spatial topology**. By flattening the image, the model treats the data as a "Bag of Pixels," effectively discarding the relative positions of features (e.g., that an eye must be adjacent to a nose). This renders the model sensitive to translation and rotation, as it lacks the spatial invariance inherent in convolutional architectures.

3.2. Deep Learning: Custom CNN (Lead: Gargi S. Pande)

In this study, a Convolutional Neural Network (CNN) was implemented to perform multi-class image classification using the PyTorch deep learning framework. The objective was to automatically learn discriminative visual features from raw images.

All images were resized to a fixed spatial resolution of 150 × 150 pixels to ensure a consistent input size for the network. The dataset was then split into training (80%) and validation (20%) subsets using a fixed random seed (SEED = 123) to ensure reproducibility of the train–validation partition.

- **Approach:** An end-to-end Convolutional Neural Network trained from scratch.
- **Architecture:** 3 Convolutional Blocks (32 → 64 → 128 filters) followed by Max Pooling and Dense Layers.
- **Input:** 150 x 150 RGB images with data augmentation (Rotation, Flips).
- **Goal:** To learn hierarchical spatial features (edges, shapes) directly from the data via backpropagation.

Training was carried out for 30 epochs using the Adam optimizer with a learning rate of 0.001, and the cross-entropy loss function was used to measure the discrepancy between the predicted class scores and the true labels. At each epoch, the model was first trained on the training set: for every batch, a forward pass produced predictions, the loss was computed, and backpropagation was used to update the model parameters.

Following training, I plotted the confusion matrix and other graphs to understand the success of the model, the graphs present the per-class precision, recall, and F1-scores. The highest-performing classes were ragno (spider) and gallina (chicken), with F1-scores of approximately 0.87 and 0.82, respectively. These high values suggest that the model could clearly identify distinct texture and structural features, such as the multiple legs of spiders or the feather patterns of chickens, which helped it distinguish them from other categories. Similarly, *farfalla* (butterfly) and *cavallo* (horse) also exhibited high precision and recall scores in the range of 0.79–0.84, reflecting strong visual separability.

- The most confident predictions occurred for ragno with 680 correct classifications, followed by cane with 572 correct instances.
- Moderate confusion was observed between *mucca–pecora* and *cane–gatto* pairs, consistent with their natural visual similarity.

- The categories *farfalla* and *gallina* showed well-separated clusters with minimal confusion, indicating that the model effectively captured color and texture cues such as wing patterns and feather distribution.
- Overall, the CNN was successful in learning meaningful visual features and delivered strong performance for some classes (such as *ragno*, *gallina*, *farfalla* and *cavallo*, with F1-scores around 0.80–0.87).
- However, its performance dropped sharply for other classes, especially *gatto* and *mucca*, where F1-scores were close to 0.50 and the confusion matrix showed frequent confusion with visually similar animals. This indicates that, while effective as a proof-of-concept, the model lacks robustness and balanced performance across all categories, and would benefit from improvements such as deeper or pre-trained architectures, targeted data augmentation and class-balancing strategies.

3.3. Hybrid Model: CNN + RF (Lead: Vipul Sharma)

- This project implements a hybrid machine learning pipeline designed to classify images into 10 distinct animal categories. The system leverages a custom Convolutional Neural Network (CNN) for automated feature extraction and utilizes a Random Forest classifier for the final predictive inference. The model achieved an overall accuracy of **72.99%** and a weighted AUC score of **0.9543** on an independent test set.

Data Preparation

- **Dataset:** The dataset consists of raw images categorized into 10 classes: *cane (dog), cavallo (horse), elefante (elephant), farfalla (butterfly), gallina (chicken), gatto (cat), mucca (cow), pecora (sheep), ragno (spider), and (squirrel).*
- **Preprocessing:** All images were resized to **150x150 pixels**.
- **Augmentation:** To improve generalization, training data underwent random horizontal flips, rotations (±10 degrees), and affine scaling (0.9x - 1.1x).
- **Splitting:** The data was partitioned into:
  - **Training:** 70%
  - **Validation:** 15%
  - **Test:** 15%

Model Architecture

The pipeline operates in two distinct stages:

1. **Deep Learning Feature Extraction (CNN):** A custom 3-layer CNN was constructed

Shutterstock

to learn spatial hierarchies in the images. * **Structure:** Three convolutional blocks with 32, 64, and 128 filters respectively. Each block includes a ReLU activation and Max Pooling. * **Bottleneck:** The convolutional output is flattened and passed through a fully connected layer (512 units) with Dropout (0.5) to prevent overfitting.

2. **Ensemble Classification (Random Forest):** Instead of using the CNN's Softmax layer for final classification, the 512-dimensional feature vectors extracted from the CNN were used to train a Random Forest classifier . This leverages the CNN's ability to interpret visual data and the Random Forest's robustness in handling high-dimensional feature spaces.

Hyperparameters

Parameter	**Value**
Input Resolution	150 x 150 x 3
Batch Size	32
Optimizer	Adam
Learning Rate	0.001
Training Epochs	30
Random Seed	123
Feature Vector Size	512
RF Estimators	100

Class-Specific Performance

The model showed variance in performance across different classes:

- **Best Performing Class:** *Ragno (Spider)* with an F1-score of **0.85** and Recall of **0.90**.

- **Lowest Performing Class:** *Gatto (Cat)* with an F1-score of **0.48** and Recall of **0.38**, indicating the model frequently confused cats with other quadrupeds (likely dogs).

Conclusion

The hybrid CNN-RF approach successfully established a high-dimensional feature space capable of distinguishing between animal species with high confidence (95% AUC). While the feature extractor is robust, the lower recall on specific classes suggests that future iterations could benefit from fine-tuning the CNN specifically on confused classes or increasing the complexity of data augmentation.

3.4. Transfer Learning: ResNet50 (Lead: Sundram A. Pandita)

- **Approach:** A ResNet50 architecture pre-trained on ImageNet (1.2 million images).
- **Configuration:** Feature extraction layers were frozen, and a custom classification head was fine-tuned.
- **Training:** Trained for 5 epochs using CrossEntropyLoss and Adam optimizer.
- **Goal:** To leverage pre-learned knowledge of complex textures and shapes ("Transfer Learning") to overcome data scarcity.

4. Experimental Results

The models were evaluated on an unseen Test Set. The results confirm a clear performance hierarchy driven by the quality of feature representation.

Table 1: Final Performance Comparison

Model Architecture	Lead	Accuracy	F1-Score	Key Observation
Baseline RF	Anirudh Pal	**40.8%**	**33.6%**	Fails on spatial structure; confuses similar textures.
Hybrid (CNN+RF)	Vipul Sharma	**72.7%**	**72.0%**	Robust performance; validates CNN feature quality.
Custom CNN	Gargi S.P.	**71.0%**	**68.7%**	Significant gain via spatial feature learning.

ResNet50	Sundram A.P.	**92.5%**	**92.0%**	State-of-the-art; benefits from ImageNet knowledge.

Figure 1: Comparative Analysis of Model Accuracy across all four architectures.

5. Analysis & Discussion

5.1. The Limitations of Raw Pixels (RF Baseline)

The Baseline RF model struggled significantly (~40% accuracy). Feature Importance analysis reveals the cause: the model focused entirely on the center pixel blob, ignoring edges and shapes.

This confirms that raw pixels lack **Spatial Invariance**. The model frequently confused animals with similar textures (e.g., *Cane* vs. *Gatto*) because it looked at color intensity rather than body shape. However, it performed decently on classes with distinct backgrounds, such as *Ragno* (Spider) and *Farfalla* (Butterfly).

5.2. The Hybrid Stability (CNN + RF)

The Hybrid model achieved 72.66% accuracy with a Weighted AUC of 0.95. The Random Forest successfully classified the 512-dimensional feature vectors extracted by the CNN.

- **Strengths:** High precision on distinct classes like *Ragno* (Spider) and *Farfalla* (Butterfly).
- **Weaknesses:** It still struggled with *Gatto* (Cat, Recall 0.39), indicating that even the custom CNN features had some overlap for furry animals, leading to confusion.

5.3. The ResNet Superiority

ResNet50 outperformed all other methods with **92.5% accuracy**. The model demonstrated exceptional robustness, with stable convergence in just 5 epochs.

- **Why it won:** The pre-trained weights provided a rich "visual vocabulary" (textures, shapes) that the custom CNN could not learn from scratch on a small dataset.
- **Error Analysis:** Minor confusion remained only between visually nearly identical classes (Cow vs. Horse), likely due to background bias (green pastures).

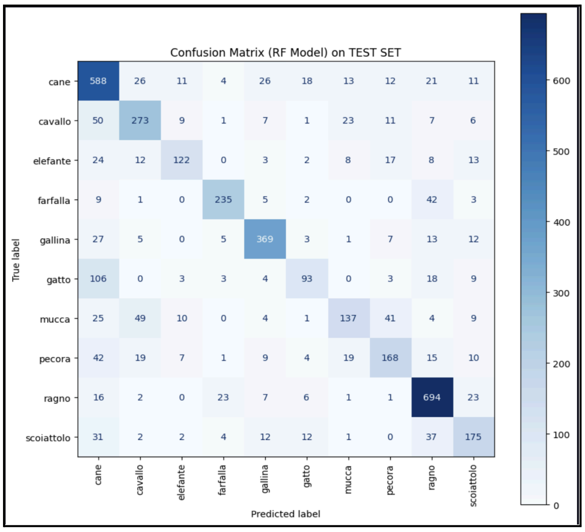
6. Conclusion

This comparative study validates the critical role of **Feature Representation** in computer vision.

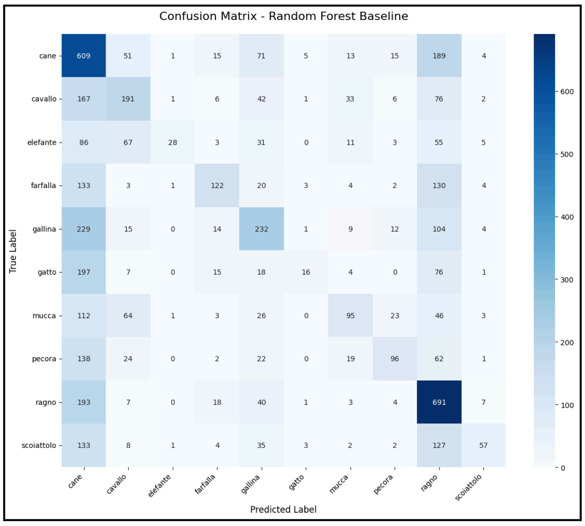
1. **Raw Pixels are Insufficient:** Random Forests hit a "performance ceiling" (~40%) because they lack spatial awareness.
2. **Convolution is Key:** Integrating Convolutional layers (CNN) nearly doubled the accuracy by allowing the model to "see" shapes.
3. **Transfer Learning is Optimal:** Leveraging ResNet50 bridged the gap to human-level performance (~92.5%), proving it to be the optimal strategy for limited-data scenarios.

7. Appendix: Visual Evidence & Metrics

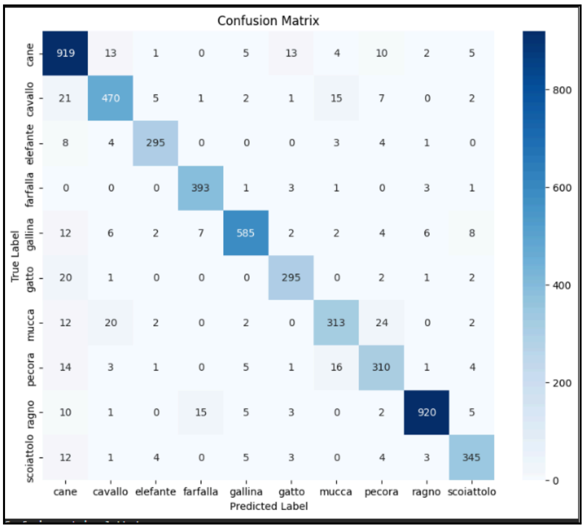
A. Confusion Matrices



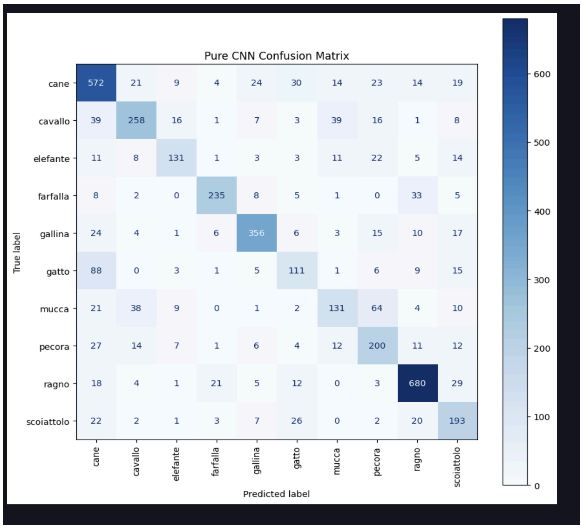
(Fig A.1 :Confusion Matrix for (RF+CNN) model)



(Fig A.2 :Confusion Matrix for RF baseline model)

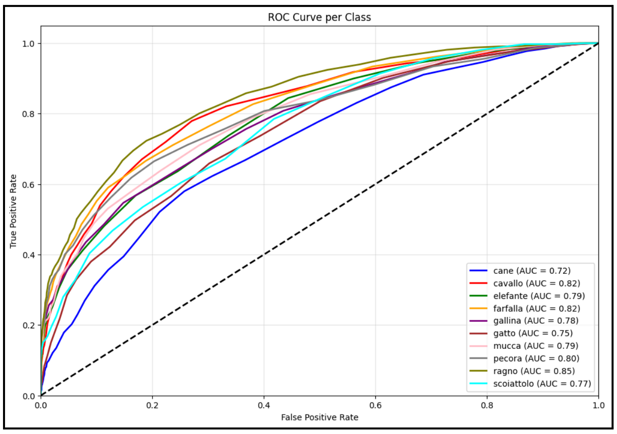


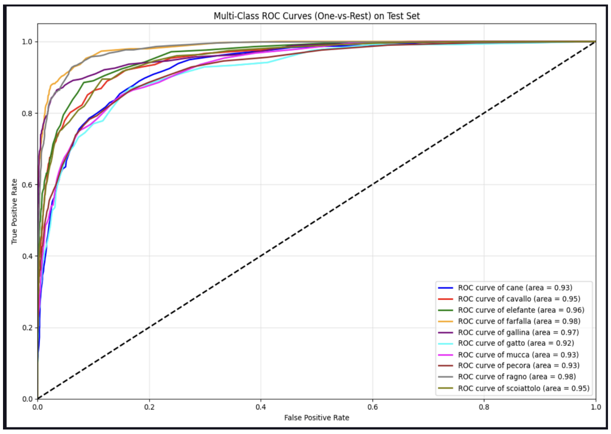
(Fig A.3 :Confusion Matrix for ResNet model)



(Fig A.4 :Confusion Matrix for CNN model)

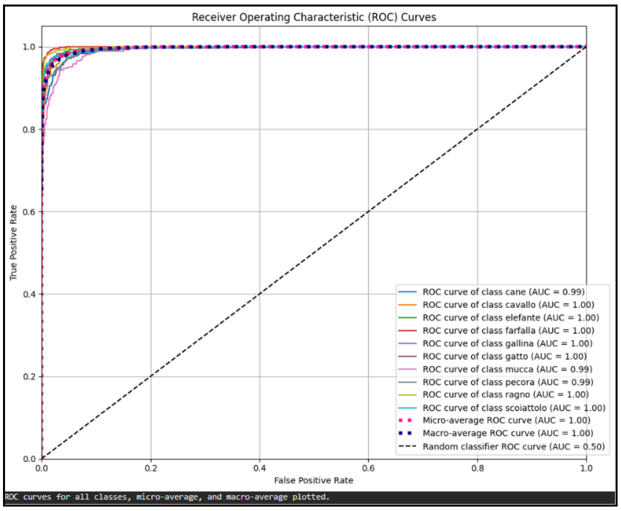
B. ROC Curves



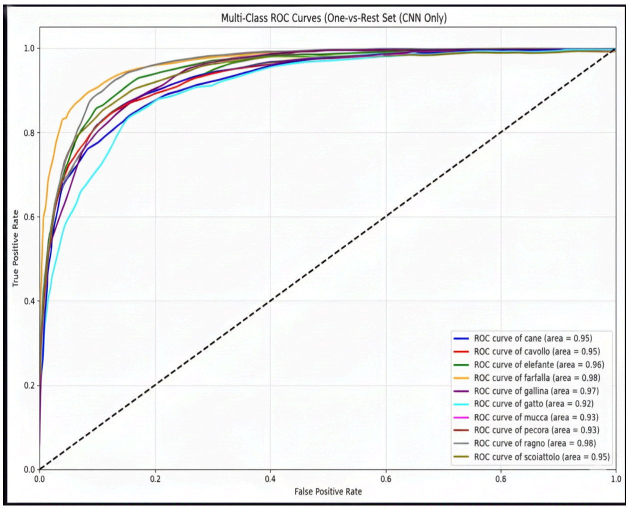


(Fig B.1 :ROC curves for RF model)

(Fig B.2 :ROC curves for (RF+CNN) model)



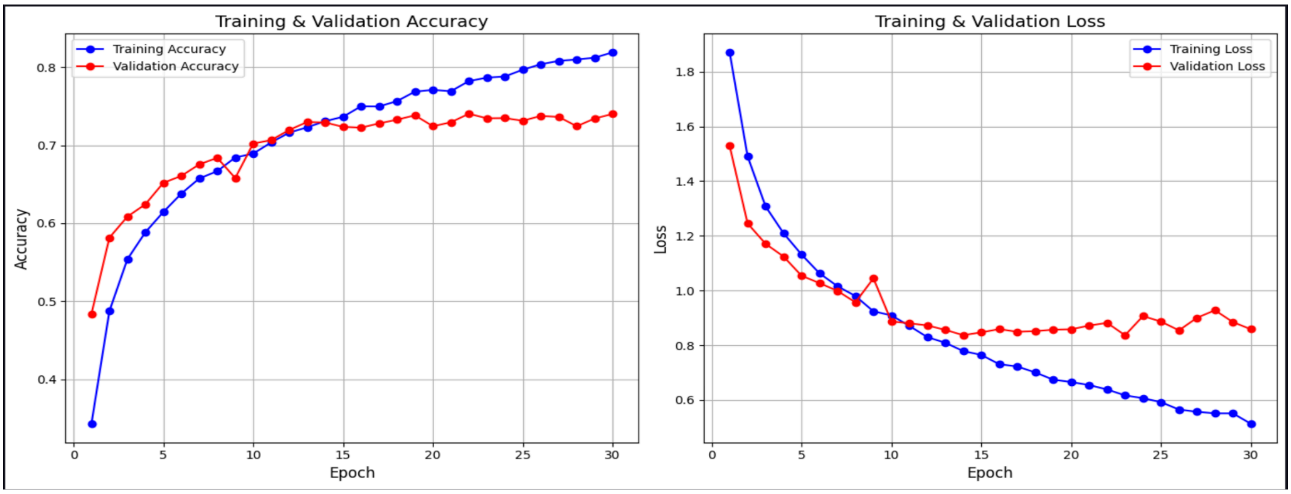
```
ROC curves for all classes, micro-average, and macro-average plotted.
```



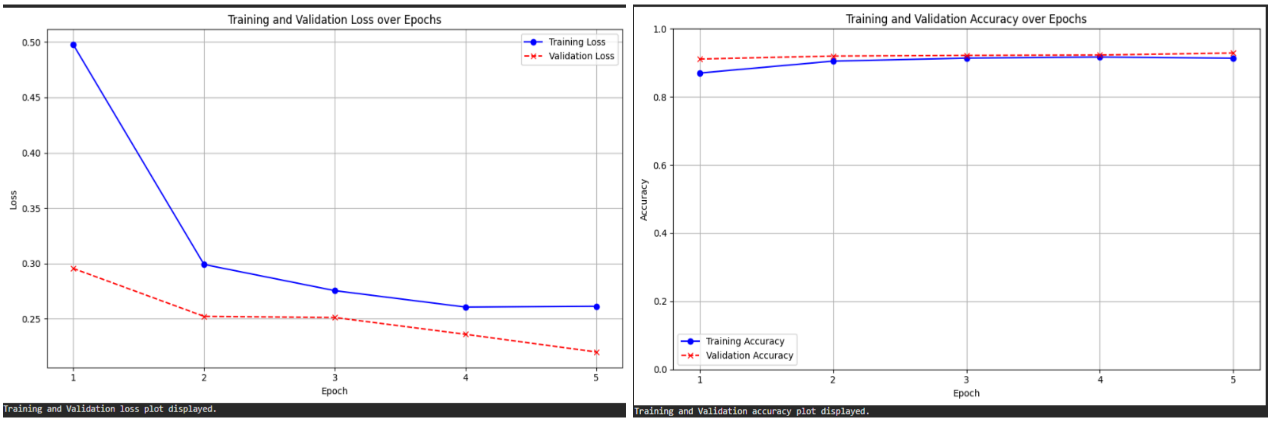
(Fig B.3 :ROC curves for Resnet model)

(Fig B.4 :ROC curves for CNN model)

C. Training History (Deep Learning Models)



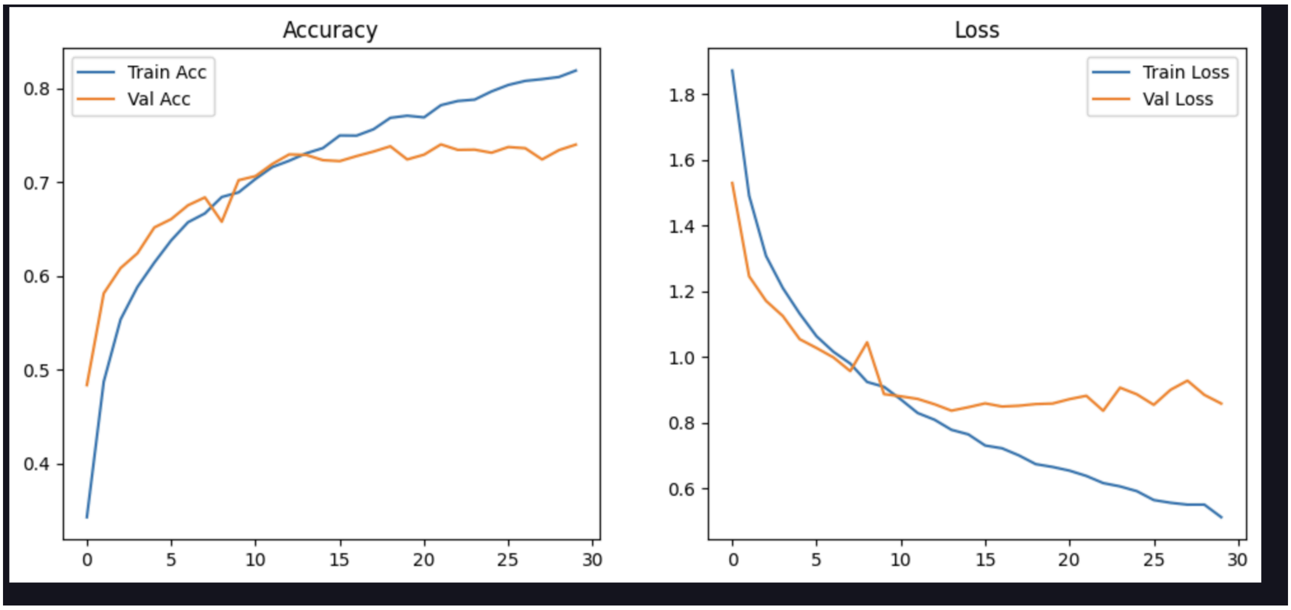
(Fig C.1 :Line graphs for training and validation loss and accuracy vs epochs for CNN+RF model)



```
Training and Validation loss plot displayed.
```

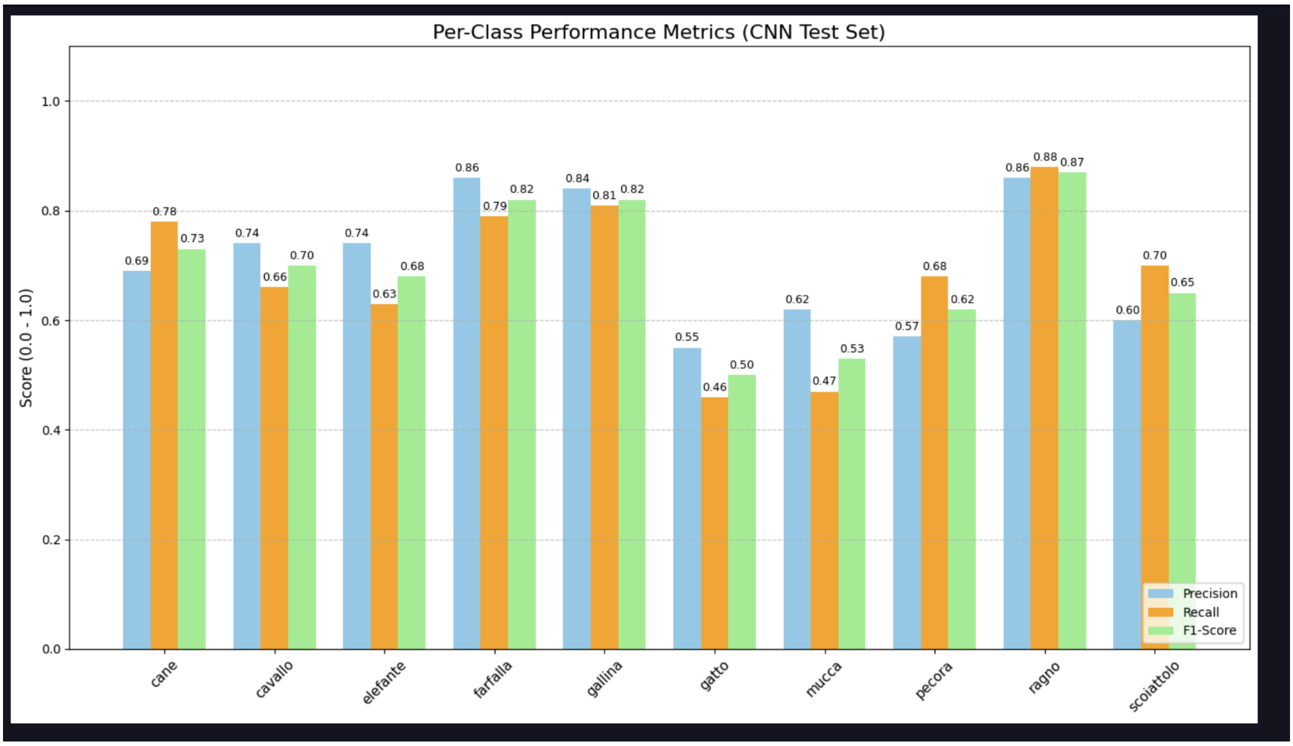
```
Training and Validation accuracy plot displayed.
```

(Fig C.2 :Line graphs for Loss vs epochs for Resnet model)

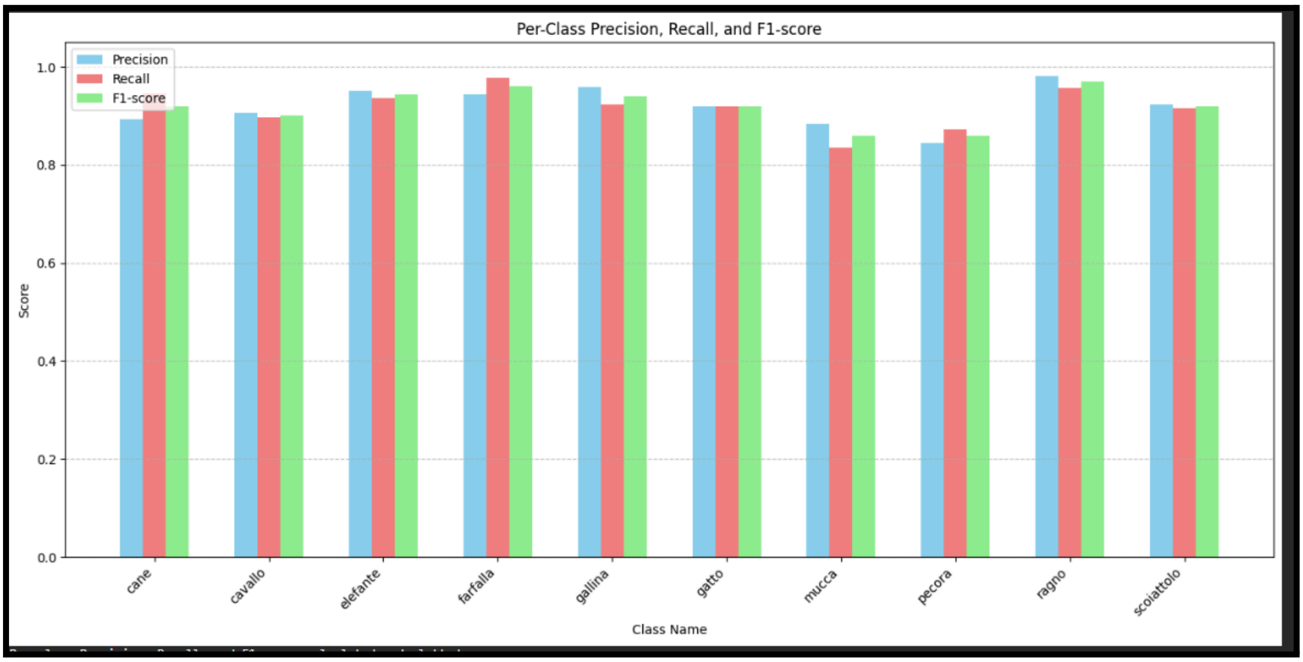


(Fig C.3 :Line graphs for training and validation loss and accuracy vs epochs for CNN model)

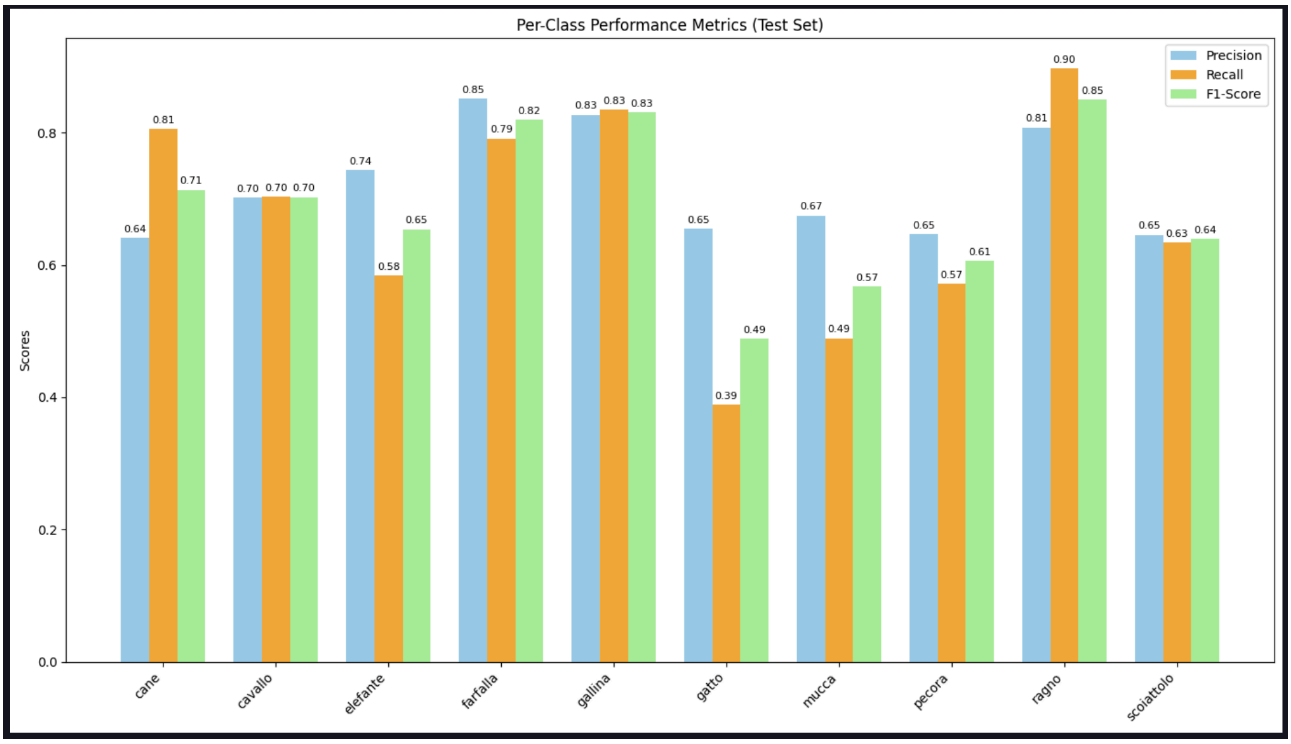
D. Per class metrics



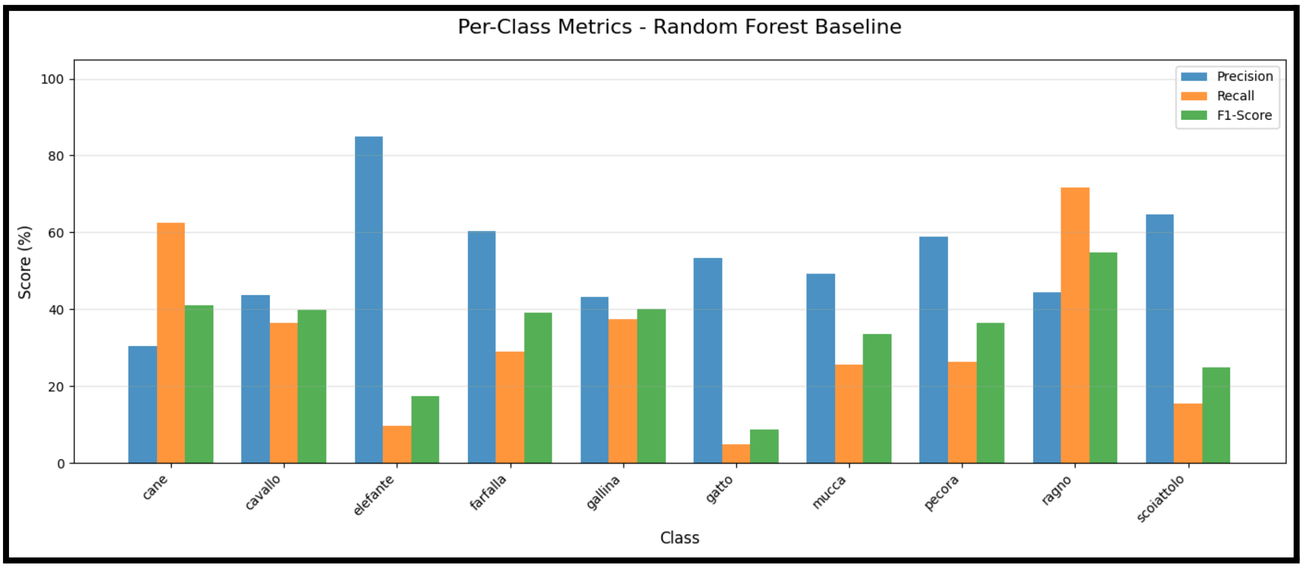
(Fig D.1 :Per class precision,recall and F1 score for CNN model)



(Fig D.2 :Per class precision,recall and F1 score for Resnet model)



(Fig D.3 :Per class precision,recall and F1 score for CNN+RF model)



(Fig D.4 :Per class precision,recall and F1 score for RF model)