ANIMAL10 DATASET IMAGE CLASSIFICATION USING PYTORCH

VANSH DATTA

1303819

INTRODUCTION

- We chose PyTorch because it is flexible and efficient when building and training neural networks. The Kaggle Animals-10 dataset presents a realistic and challenging classification task, given the variety of the animals and visual similarity between certain classes. Employing transfer learning and proper evaluation techniques, we aim to train an optimal and accurate classifier.

DATA PREPROCESSING

- Resizing: Scales all images to 224x224 pixels.
- Random Horizontal Flip: Included to account for left-right variations.
- Random Rotation: Helps the model handle differently rotated inputs.
- Color Jitter: Modifies brightness and contrast to account for lighting situations.
- Normalization: Converts pixel values to a standard range, helping with quicker convergence during training.

MODEL ARCHITECTURE

- This project uses transfer learning with ResNet18, an ImageNet pre-trained convolutional neural network. Some of the most critical reasons behind this choice are:
- Proven performance on image classification.
- Reduced training time due to pre-trained weights.
- Balanced model complexity and speed.
- ResNet18's last layer is replaced so that it outputs 10 class probabilities instead of the default 1000. The earlier layers are frozen or fine-tuned depending upon training performance.
- The model is trained with:
- Cross-Entropy Loss: Suitable for multi-class classification.
- Adam Optimizer: Renowned for adaptive learning rate and quick convergence.
- The model trains using a GPU (if available), speeding up training significantly.

TRAINING PROCESS

- Training occurs over a number of epochs (e.g., 10–20), with each epoch including:
- Forward pass: Model predicts output probabilities.
- Backward pass: Update weights and compute gradients.
- Evaluation on the validation set after each epoch.
- Following each epoch:
- Record training accuracy and loss.
- Compute validation accuracy to monitor generalization

EVALUATION AND RESULTS

After training, the model's performance is evaluated on the validation set against two main measures:

- Overall Accuracy: Measures how many predictions are correct.
- Confusion Matrix: Illustrates the distribution of correct and incorrect predictions across all classes.

Results Summary :

•Final Validation Accuracy: 96.39%

•Best Epoch: Epoch 10

Most Confused Class Pairs:

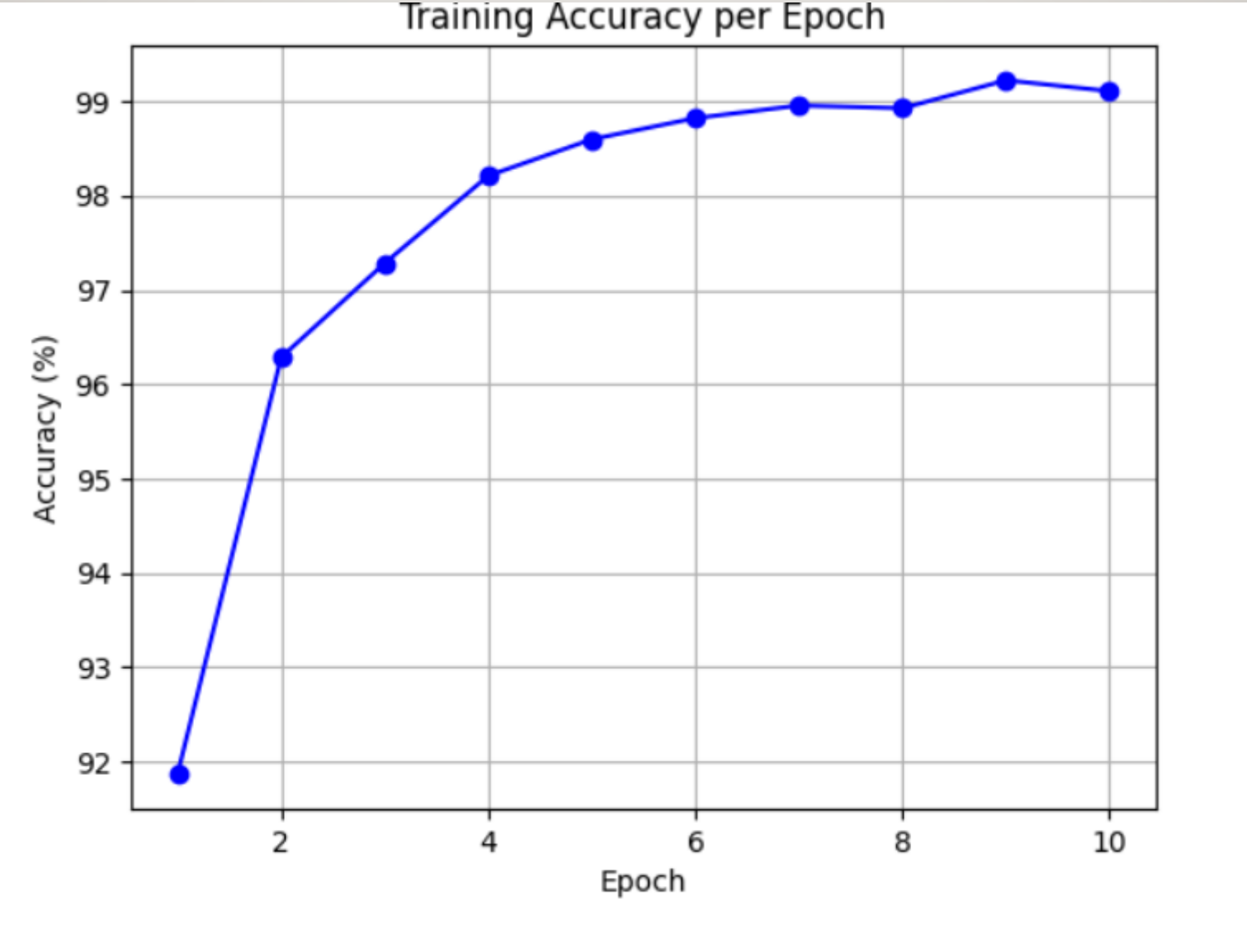
•cane (dog) misclassified as gatto (cat) — 21 times.

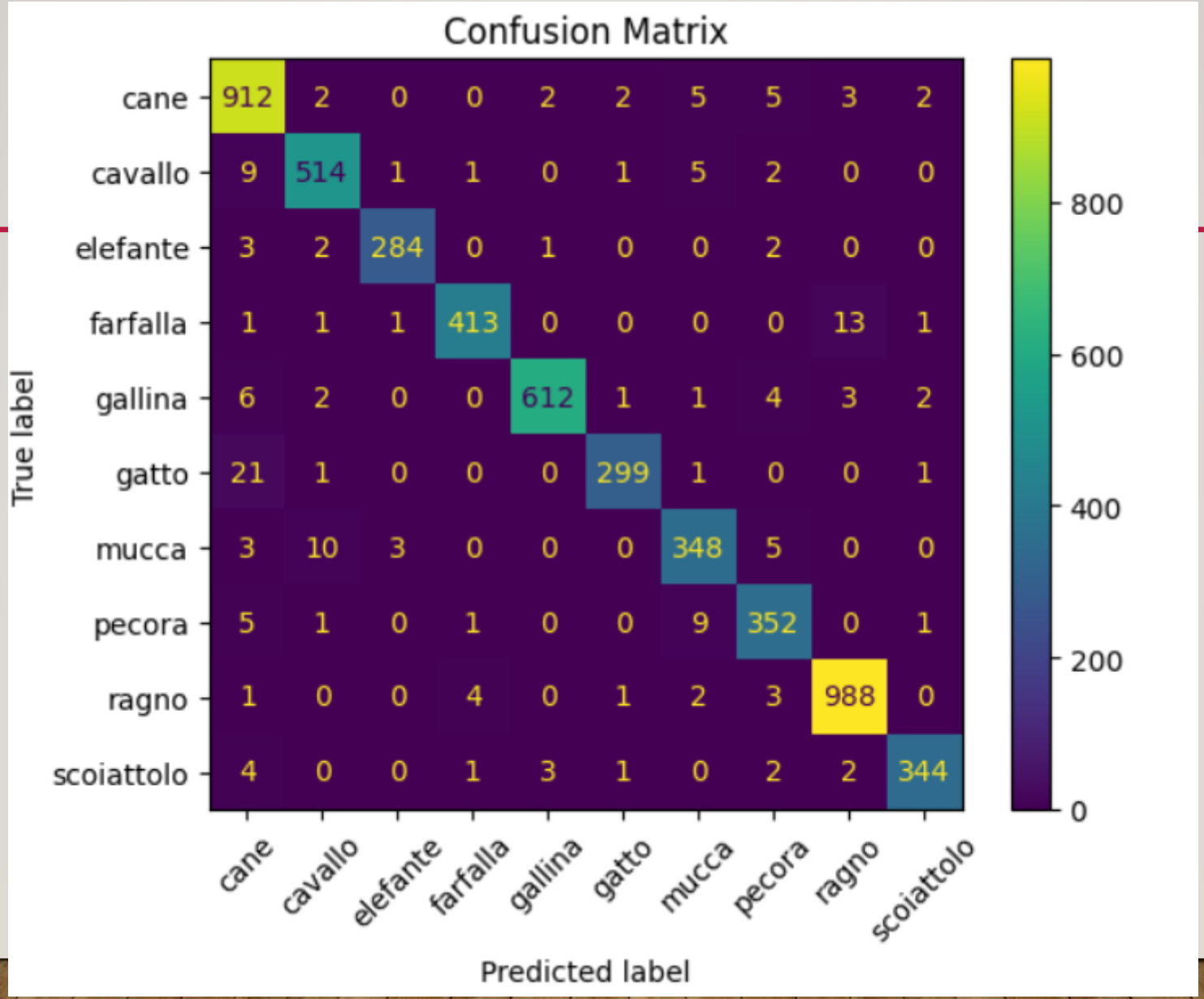
•gatto (cat) misclassified as cane (dog) — 6 times.

•farfalla (butterfly) misclassified as ragno (spider) — 13 times.

•mucca (cow) confused with pecora (sheep), and vice versa.

•scoiattolo (squirrel) misclassified as farfalla and gallina.


Training Accuracy per Epoch
Accuracy (%)
99
98
97
96
95
94
93
92
2
4
6
8
10
Epoch


Confusion Matrix
cane 912 2 0 0 2 2 5 5 3 2
cavallo 9 514 1 1 0 1 5 2 0 0
elefante 3 2 284 0 1 0 0 2 0 0
farfalla 1 1 1 413 0 0 0 0 13 1
gallina 6 2 0 0 612 1 1 4 3 2
gatto 21 1 0 0 0 299 1 0 0 1
mucca 3 10 3 0 0 0 348 5 0 0
pecora 5 1 0 1 0 0 9 352 0 1
ragno 1 0 0 4 0 1 2 3 988 0
scoiattolo 4 0 0 1 3 1 0 2 2 344
True label
cane cavallo elefante farfalla gallina gatto mucca pecora ragno scoiattolo
Predicted label
800
600
400
200
0

- model classifies very well on every class except one, and ragno (spider) and cane (dog) are the top true positives.
- Dog vs Cat (cane vs gatto) and Cow vs Horse (mucca vs cavallo) are the most incorrectly classified pairs, which is consistent with usual misclassification patterns in animal data sets.
- Accuracy can be adjusted to some degree for gatto, farfalla, and mucca.

VISUALIZATION

P: cavallo
A: cavallo

P: cane
A: cane

P: ragno
A: ragno

P: scoiattolo
A: scoiattolo

SAVING AND DEPLOYMENT

- To utilize the trained model again for future use:
- • Weights of the model are saved to a file.
- • The model can be reloadable during inference without retraining.
- It enables:
- • Testing new images in the future.
- • Utilization in real-world applications such as animal recognition programs or educational games.
- For deployment, a simple interface can be designed for the purpose of uploading pictures and showing the predicted labels. This enables the model to be used outside the notebook.