Laboratory Work
Application of pretrained model

Objective: the purpose of this laboratory work is to study the application of pretrained convolutional neural network models for image classification and to analyze transfer learning strategies using quantitative evaluation metrics.

The submission includes: ipynb and the report file. The report must include step by step descriptions, experimental results, metric values, plots, and brief explanations. The report must follow the provided template.

Task 1. Dataset selection and preparation

Select a small open image classification dataset suitable for transfer learning. The dataset must be publicly available and compatible with PyTorch. The dataset must contain at least three classes. The total number of images used must be at least 500. Split the dataset into training and test sets using a fixed random seed, with 70 percent of the data used for training and 30 percent for testing. In the report, specify the dataset name and source, number of classes, total number of images, image resolution, color format, and class balance. Resize all images to the input size required by the pretrained model. Apply ImageNet normalization. Print the shape of one training batch.

Task 2. Pretrained model loading

Load a pretrained CNN model from torchvision.models. Print the model architecture and the total number of trainable parameters. Identify the feature extractor part and the classification head.

Task 3. Transfer learning with frozen backbone

Freeze all convolutional layers of the pretrained model. Replace the final classification layer to match the number of classes in the dataset. Train the model for exactly 8 epochs using CrossEntropyLoss and the Adam optimizer with learning rate. Track training and test loss and accuracy for each epoch. Plot loss and accuracy curves. Report the final test accuracy.

Task 4. Fine tuning

Unfreeze the last convolutional block of the pretrained model while keeping all earlier layers frozen. Train the model for exactly 5 epochs using Adam with learning rate 0.0001. Track training and test loss and accuracy. Plot loss and accuracy curves. Report the final test accuracy and compare it with the frozen backbone model.

Task 5. Metrics based evaluation

For both transfer learning strategies, compute accuracy, precision, recall, and macro averaged F1 score on the test set. Present the results in a single comparison table. Explain briefly why macro F1 is used for this task.

Task 6. Error analysis

For the better performing model, compute and visualize the confusion matrix. Display exactly six test images: three correctly classified and three misclassified. For each image, show the true label and the predicted label. Write a short explanation of which classes are confused most often and possible reasons for these errors.

Task 7. Comparison with training from scratch
Implement a small CNN model trained from scratch with fewer parameters than pre-trained model. Train this model for exactly 8 epochs using the same dataset split and optimizer settings as in Task 3. Evaluate the model using the same metrics. Compare its performance with the pretrained model and explain the observed difference.

Task 8. Feature representation analysis
Select one test image and extract feature maps from one early convolutional layer and one deeper convolutional layer of the pretrained model. Visualize exactly four feature maps from each layer.

Task 9. Model efficiency analysis
Compare the pretrained model and the model trained from scratch in terms of number of trainable parameters, training time per epoch, and final test accuracy. Summarize the comparison in a table and explain the trade off between model complexity and performance.

Task 10. Final conclusions
Summarize the experimental results.