Laboratory Work 4
Application of Pretrained Convolutional Neural Networks

Sekenova Balym 23B031433

February 23, 2026

Contents

1 Introduction

The objective of this laboratory work is to investigate the application of pretrained convolutional neural networks for multi-class image classification and to analyze different transfer learning strategies using quantitative evaluation metrics.

Modern deep learning models are typically trained on large-scale datasets such as ImageNet, which contains over one million labeled images. Transfer learning enables leveraging these pretrained representations for new tasks with limited data, significantly improving performance and training efficiency.

In this work, three approaches are compared:

- Transfer learning with a frozen convolutional backbone
- Fine-tuning of high-level convolutional layers
- Training a small convolutional neural network from scratch

The Oxford 102 Flowers dataset was used as a fine-grained classification benchmark to evaluate these strategies.

2 Task 1: Dataset Selection and Preparation

2.1 Dataset Description

Dataset name: Oxford 102 Flowers
Source: Visual Geometry Group, University of Oxford
Number of classes: 102
Total number of images: 8189
Image format: RGB
Resized resolution: 224×224

The dataset contains fine-grained flower categories with subtle inter-class differences.

2.2 Data Preprocessing

Images were resized to 224×224 to match the input size required by pretrained CNN models trained on ImageNet.

ImageNet normalization was applied:

$$\text{mean} = [0.485, 0.456, 0.406]$$

$$\text{std} = [0.229, 0.224, 0.225]$$

The dataset was split into 70% training and 30% testing using a fixed random seed.

3 Task 2: Pretrained Model Loading

The ResNet18 architecture pretrained on ImageNet was used.
Total number of parameters: 11,689,512
Trainable parameters (fine-tuned model): 11,228,838

The model consists of:

- Feature extractor (convolutional backbone)
- Fully connected classification head

4 Task 3: Transfer Learning with Frozen Backbone

All convolutional layers were frozen. Only the final fully connected layer was trained for 8 epochs using:

- Loss function: CrossEntropyLoss
- Optimizer: Adam (learning rate = 0.001)

Final test accuracy (Frozen Backbone): **0.8252** (82.52%)

4.1 Training and Test Curves

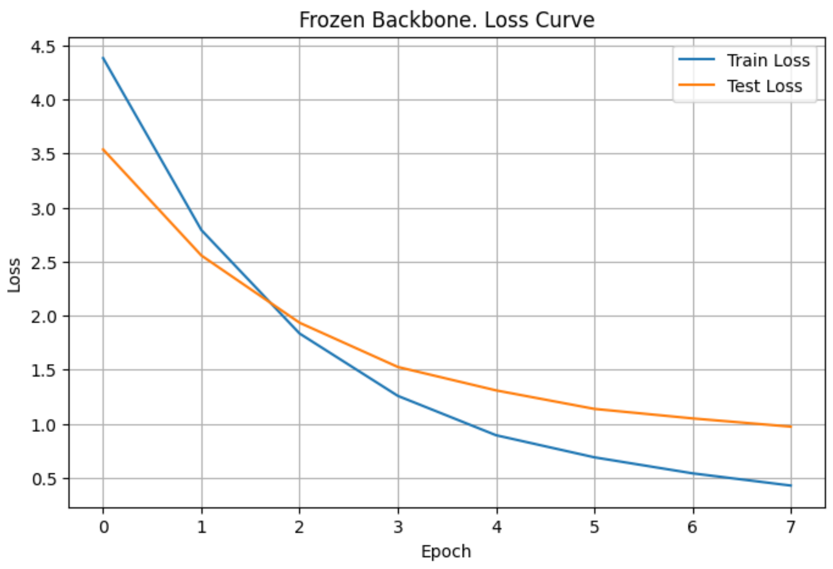


Figure 1: Training and Test Loss (Frozen Backbone)

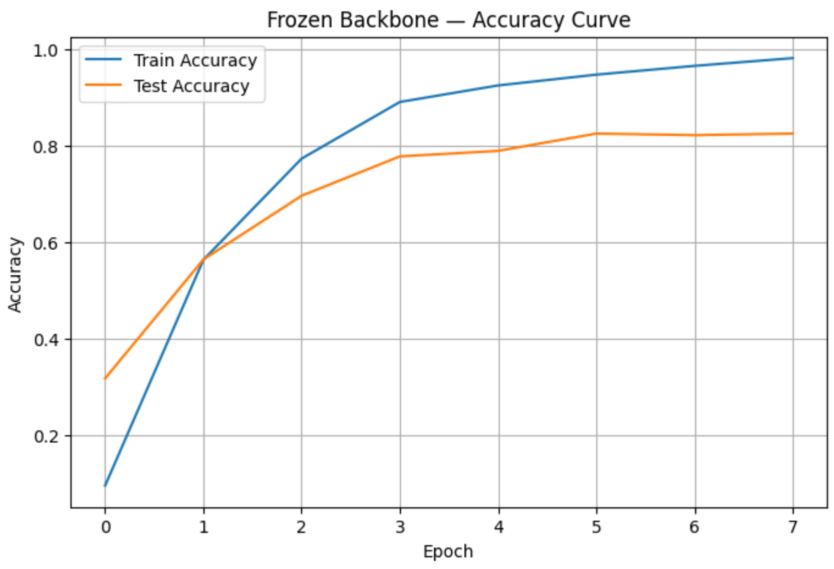


Figure 2: Training and Test Accuracy (Frozen Backbone)

The model achieved strong performance using only the pretrained feature extractor without updating convolutional layers.

5 Task 4: Fine-Tuning

In this experiment, the last convolutional block (layer4) was unfrozen while earlier layers remained frozen.

Training configuration:

- Epochs: 5
- Optimizer: Adam (learning rate = 0.0001)
- Loss: CrossEntropyLoss

Final test accuracy (Fine-Tuning): **0.8954** (89.54%)

Fine-tuning improved performance by approximately 7% compared to the frozen backbone strategy.

5.1 Training and Test Curves

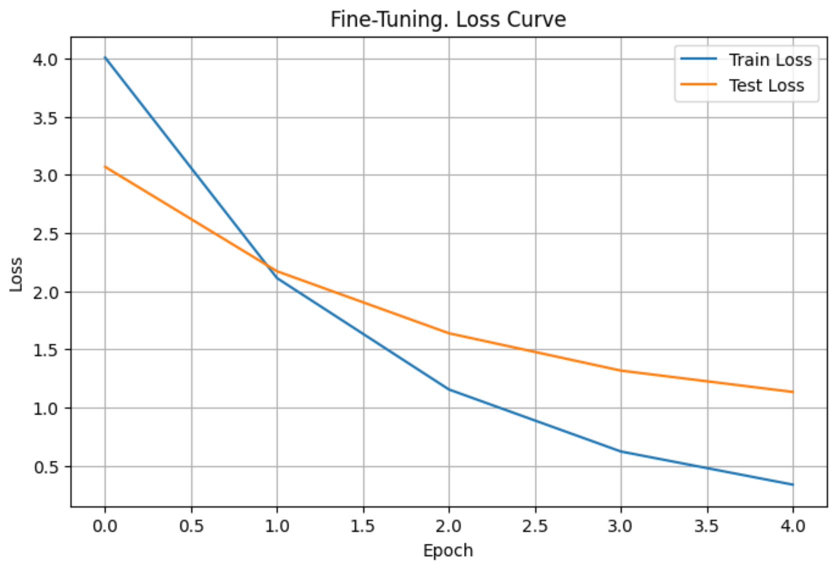


Figure 3: Training and Test Loss (Fine-Tuning)

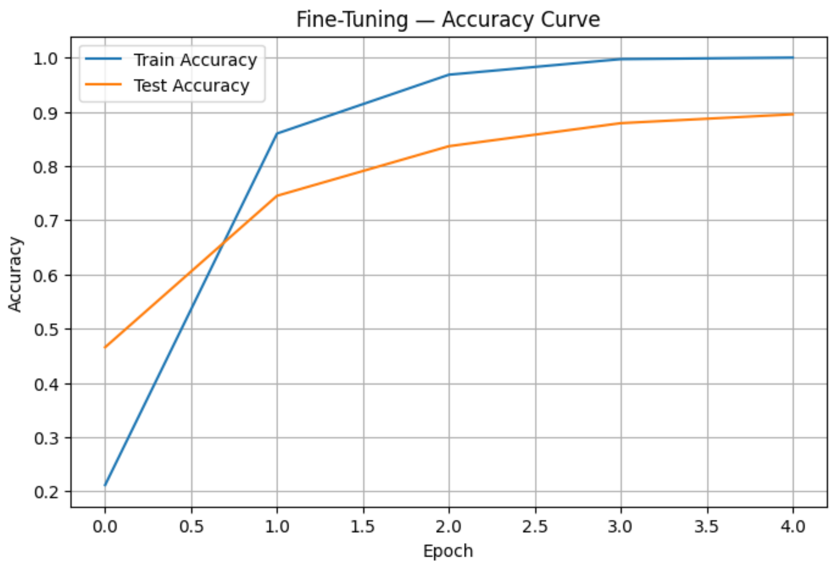


Figure 4: Training and Test Accuracy (Fine-Tuning)

6 Task 5: Metrics-Based Evaluation

Both strategies were evaluated using accuracy, precision, recall and macro-averaged F1 score.

Model	Accuracy	Precision	Recall	Macro F1
Frozen Backbone	0.8252	0.8454	0.8373	0.8207
Fine-Tuning	0.8954	0.9040	0.9056	0.8940

Table 1: Comparison of Transfer Learning Strategies

Macro F1 was used because the dataset contains 102 classes. Macro-averaging assigns equal importance to each class, preventing dominant classes from biasing the evaluation.

7 Task 6: Error Analysis

7.1 Confusion Matrix

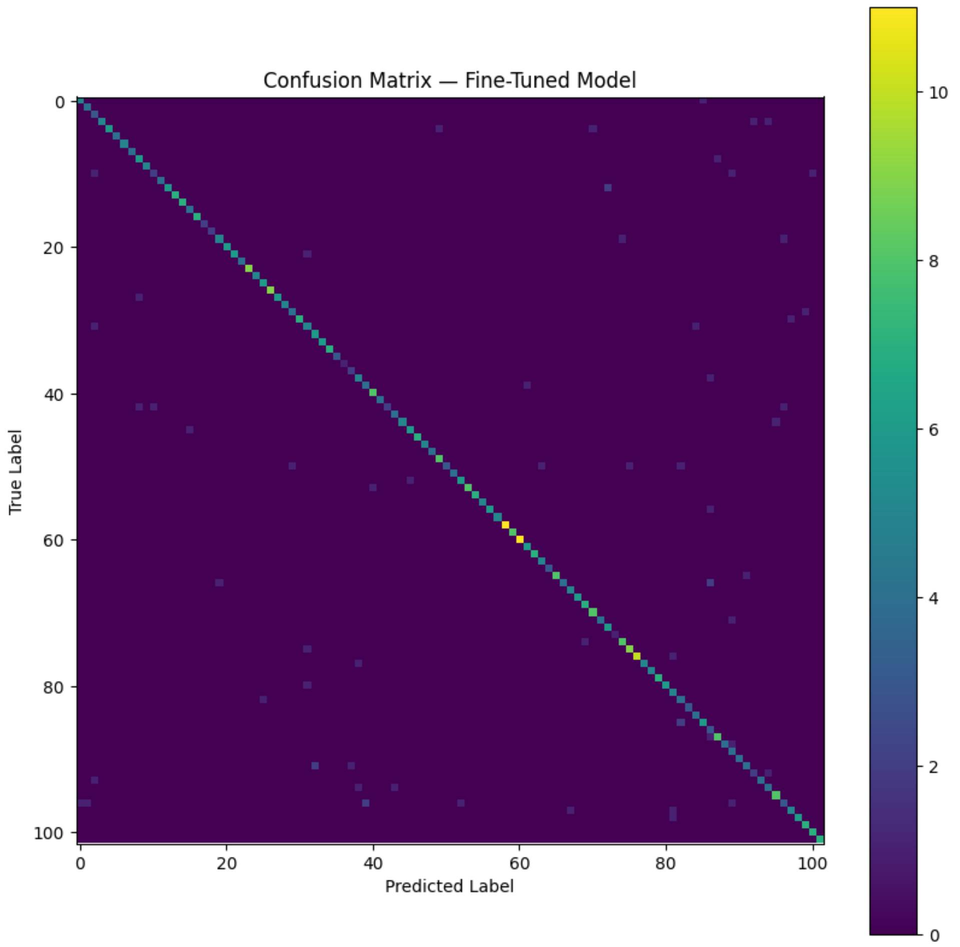


Figure 5: Confusion Matrix (Fine-Tuned Model)

The confusion matrix indicates that most classification errors occur between visually similar flower species. Since the Oxford 102 Flowers dataset represents a fine-grained classification problem, inter-class differences are often subtle, involving small variations in petal shape, color gradients, or texture patterns.

Misclassifications primarily arise between classes with similar color distributions and structural characteristics. This confirms that even deep pretrained models face challenges when discriminating between highly similar categories.

7.2 Correct and Misclassified Examples

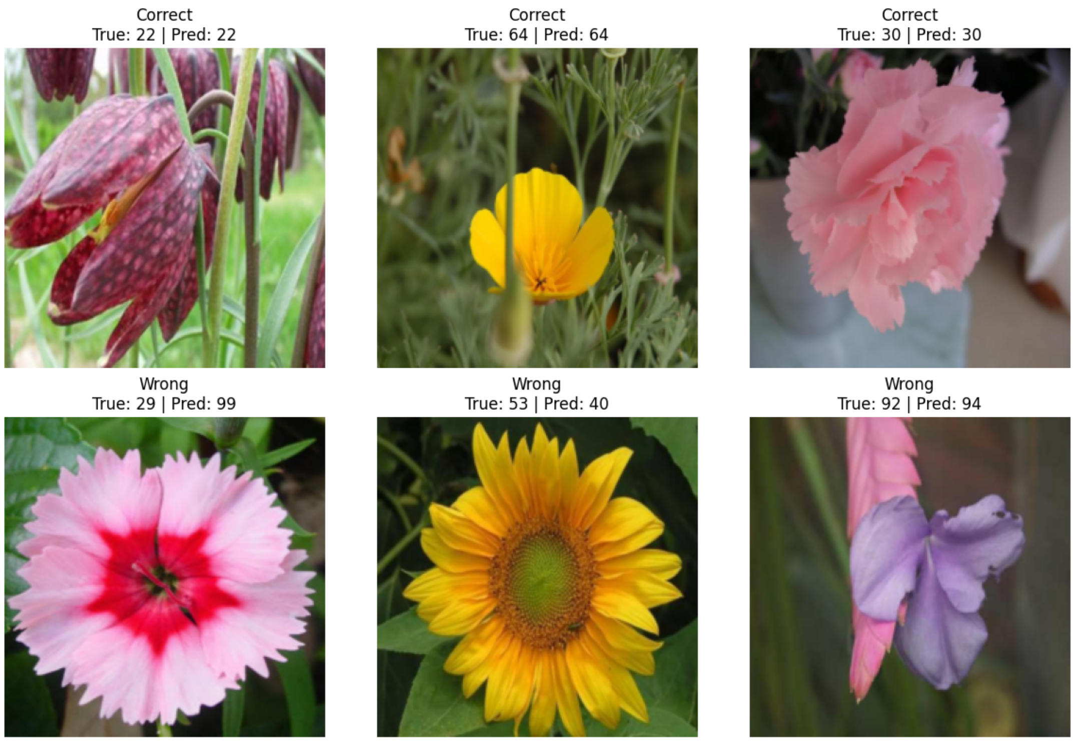


Figure 6: Three Correct and Three Misclassified Images

Misclassifications primarily occur between fine-grained classes with subtle visual differences.

8 Task 7: Training from Scratch

A small CNN model with 152,486 parameters was implemented and trained for 8 epochs using the same optimizer and dataset split.

Final test accuracy (Scratch Model): **0.1650** (16.50%)

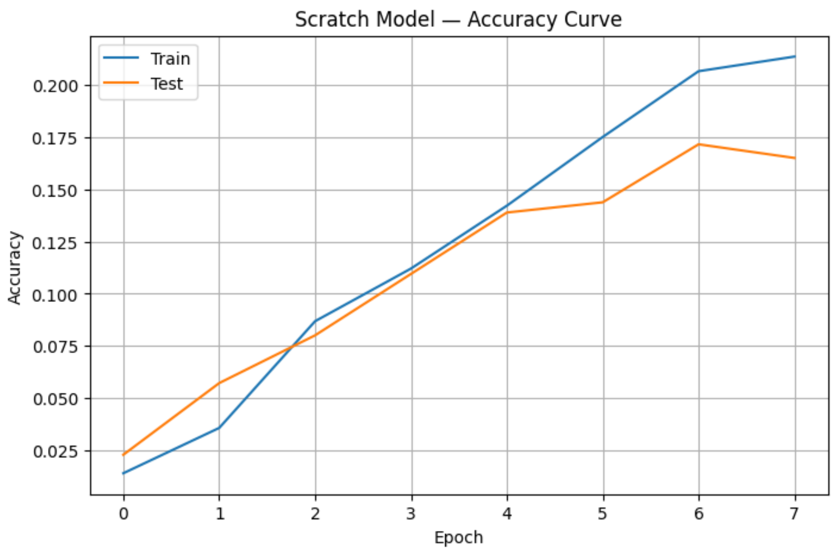


Figure 7: Accuracy Curve (Scratch Model)

The scratch model significantly underperformed compared to transfer learning approaches, demonstrating the importance of pretrained feature representations.

9 Task 8: Feature Representation Analysis

Feature maps were extracted from:

- Early convolutional layer (conv1)
- Deep convolutional block (layer4)

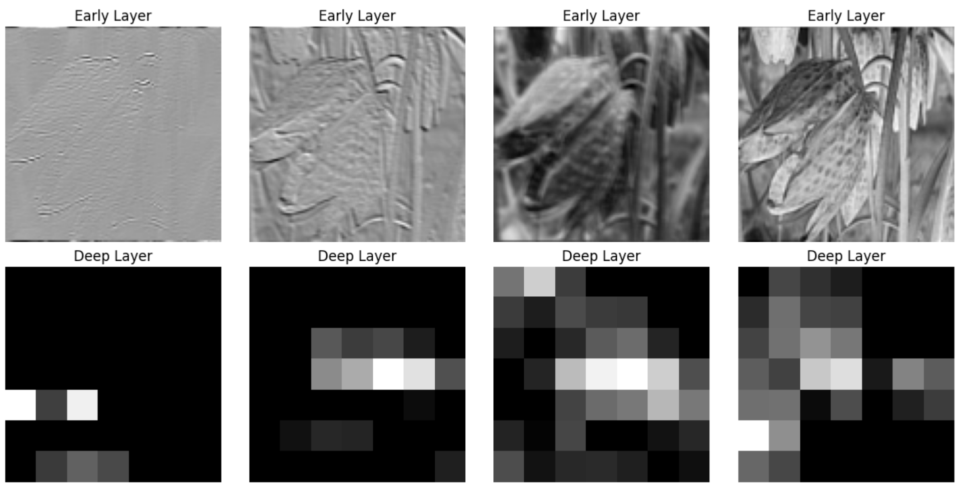


Figure 8: Feature Maps from Early and Deep Layers

Early layers captured low-level features such as edges and textures, while deeper layers represented higher-level semantic structures of flower shapes.

10 Task 9: Model Efficiency Analysis

Model	Trainable Params	Time per Epoch (s)	Test Accuracy
Fine-Tuned ResNet18	11,228,838	7.51	0.8954
Scratch CNN	152,486	8.02	0.1650

Table 2: Model Efficiency Comparison

Although the pretrained model contains significantly more parameters, it achieves substantially higher accuracy. This demonstrates the trade-off between model complexity and performance. Transfer learning enables superior generalization by leveraging knowledge learned from large-scale datasets such as ImageNet.

11 Task 10: Final Conclusions

This laboratory work demonstrated the practical effectiveness of transfer learning for fine-grained multi-class image classification.

The experimental results clearly show that:

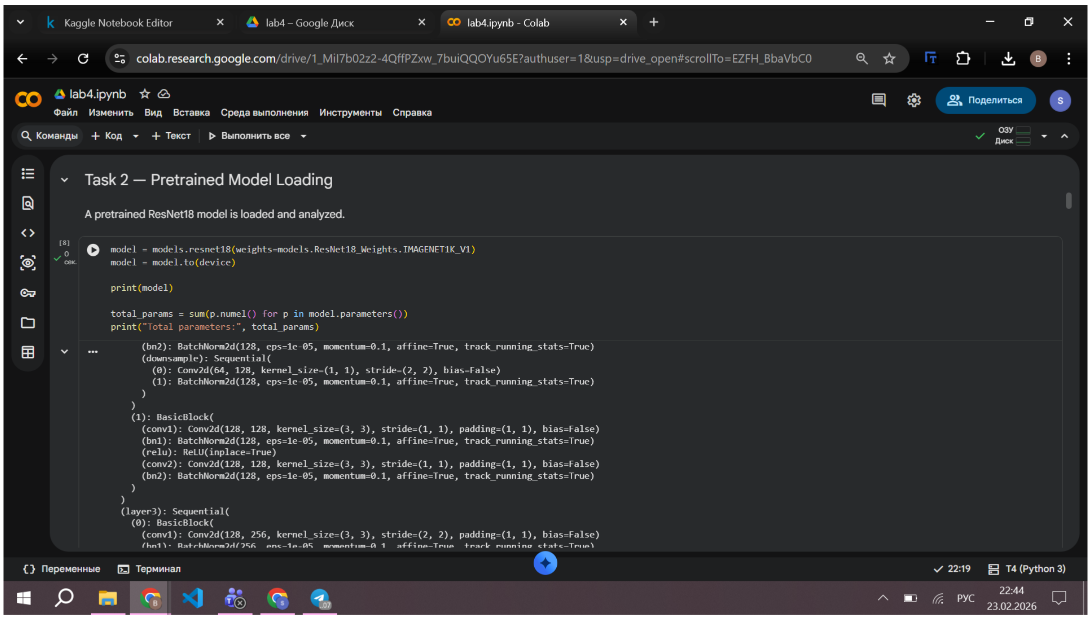
- Transfer learning significantly improves performance compared to training from scratch.
- Fine-tuning high-level convolutional layers further enhances classification accuracy.
- A small CNN trained from scratch fails to generalize effectively on a complex 102-class dataset.
- Pretrained models provide powerful hierarchical feature representations learned from large-scale datasets such as ImageNet.

Overall, the results confirm the superiority of transfer learning approaches for complex visual recognition tasks with limited training data.

A Appendix

A.1 Google Colab Execution Evidence

The following screenshot demonstrates the execution of the notebook in Google Colab, including model training outputs and metric calculations.



Task 2 — Pretrained Model Loading

A pretrained ResNet18 model is loaded and analyzed.

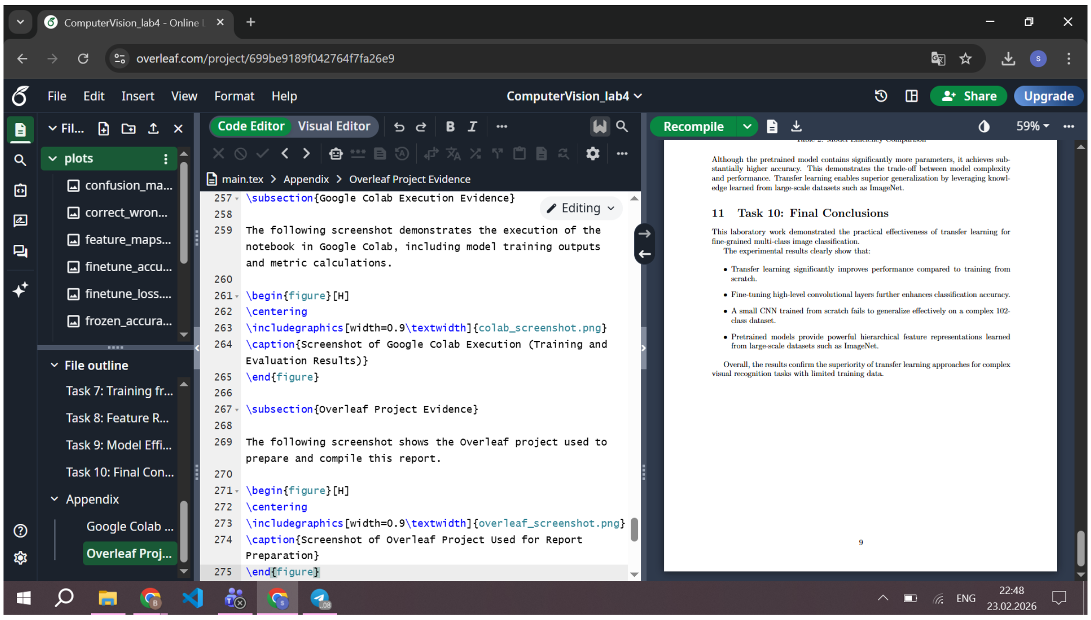
```
model = models.resnet18(weights=models.ResNet18_Weights.IMAGENET1K_V1)
model = model.to(device)

print(model)

total_params = sum(p.numel() for p in model.parameters())
print("Total parameters:", total_params)
```

```
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (downsample): Sequential(
          (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
      (1): BasicBlock(
        (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (layer3): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
```

Figure 9: Screenshot of Google Colab Execution (Training and Evaluation Results)

A.2 Overleaf Project Evidence

The following screenshot shows the Overleaf project used to prepare and compile this report.



```
\subsection{Google Colab Execution Evidence}

The following screenshot demonstrates the execution of the notebook in Google Colab, including model training outputs and metric calculations.

\begin{figure}[H]
\centering
\includegraphics[width=0.9\textwidth]{colab_screenshot.png}
\caption{Screenshot of Google Colab Execution (Training and Evaluation Results)}
\end{figure}

\subsection{Overleaf Project Evidence}

The following screenshot shows the Overleaf project used to prepare and compile this report.

\begin{figure}[H]
\centering
\includegraphics[width=0.9\textwidth]{overleaf_screenshot.png}
\caption{Screenshot of Overleaf Project Used for Report Preparation}
\end{figure}
```

Although the pretrained model contains significantly more parameters, it achieves substantially higher accuracy. This demonstrates the trade-off between model complexity and performance. Transfer learning enables superior generalization by leveraging knowledge learned from large-scale datasets such as ImageNet.

11 Task 10: Final Conclusions

This laboratory work demonstrated the practical effectiveness of transfer learning for fine-grained multi-class image classification.

The experimental results clearly show that:

- Transfer learning significantly improves performance compared to training from scratch.
- Fine-tuning high-level convolutional layers further enhances classification accuracy.
- A small CNN trained from scratch fails to generalize effectively on a complex 102-class dataset.
- Pretrained models provide powerful hierarchical feature representations learned from large-scale datasets such as ImageNet.

Overall, the results confirm the superiority of transfer learning approaches for complex visual recognition tasks with limited training data.

9

Figure 10: Screenshot of Overleaf Project Used for Report Preparation

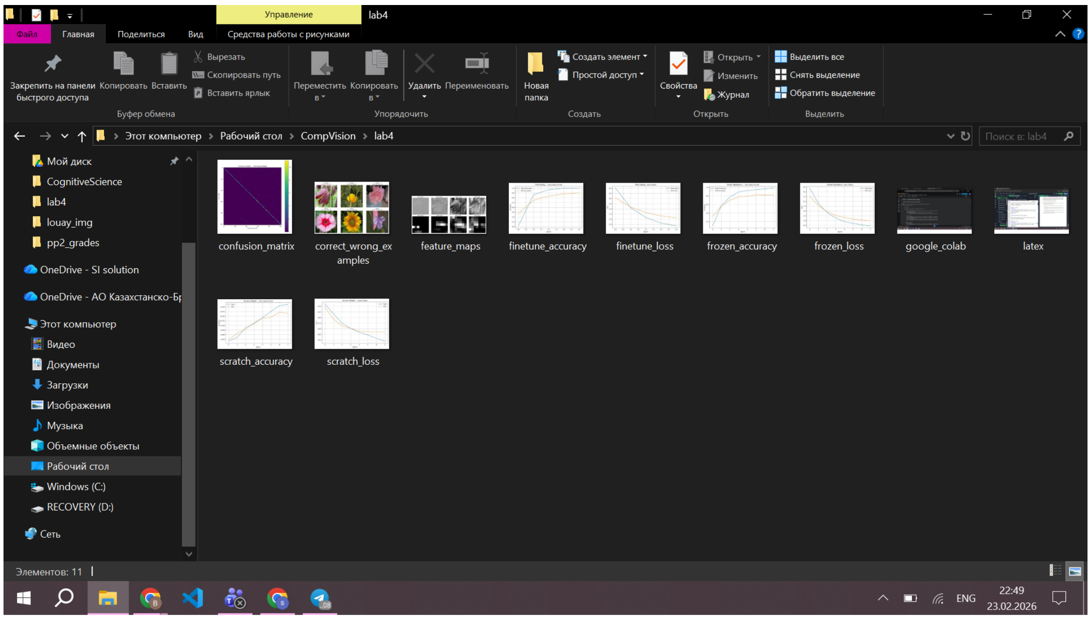


Figure 11: Screenshot of Local Storage for keeping photos