Lab 10 - Introduction to Deep Learning for Vision (CNNs & Pretrained Models)

Tigist Wondimneh
GSR/5506/17

Github- Link

Objective

- Understand the fundamentals of deep learning for computer vision.
- Learn to perform image classification using pretrained CNNs (ResNet, MobileNet).
- Apply predictions to real-world images.
- Fine-tune models and visualize intermediate features.

1. PyTorch ResNet-18 Inference

Screenshot of Prediction:

```
Predicted class index: 281
Predicted class label: tabby
```

Interpretation:

- ResNet-18 correctly identifies the main object in the image.
- Output index corresponds to ImageNet class label.

2. TensorFlow MobileNetV2 Inference

Screenshot of Prediction:

```
Top 3 Predictions (TensorFlow MobileNetV2):
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
35363/35363 ━━━━━━━━━━━━━━━━━━━━ 0s 6us/step
('n02123159', 'tiger_cat', np.float32(0.491762))
('n02123045', 'tabby', np.float32(0.30150068))
('n02124075', 'Egyptian_cat', np.float32(0.10086851))
```

Interpretation:

- MobileNetV2 is lightweight and provides quick predictions.
- Top-3 probabilities indicate model confidence.

3. Custom Image Input with OpenCV

Screenshot:

Interpretation:

- OpenCV can feed images directly into CNNs for prediction.
- Predicted labels are consistent with previous results.

4. Suggested Exercises

4.1 Run predictions on your own image dataset

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step
baby.jpeg -> Predicted: ski_mask (21.20%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step
bird.jpeg -> Predicted: indigo_bunting (61.01%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 173ms/step
cat.jpeg -> Predicted: tiger_cat (49.18%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 144ms/step
cow.jpeg -> Predicted: ox (76.93%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 146ms/step
dog.jpeg -> Predicted: golden_retriever (67.51%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 151ms/step
frog.jpeg -> Predicted: tailed_frog (67.86%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 235ms/step
lion.jpeg -> Predicted: lion (88.88%)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step
zebra.jpeg -> Predicted: zebra (94.40%)
```

Interpretation:

- Predictions demonstrate model performance on multiple images.
- Useful for batch inference scenarios.

4.2 Compare ResNet vs MobileNet predictions

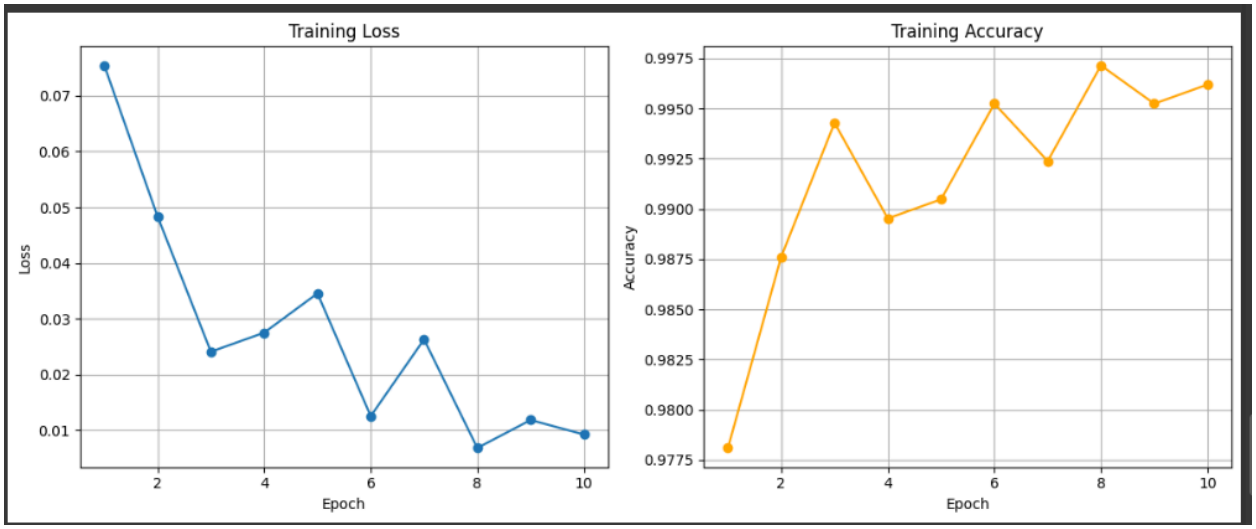
```
ResNet Prediction: golden retriever

1/1 ━━━━━━━━━━━━━━━━━━━━ 5s 5s/step
MobileNet Prediction: golden_retriever
```

Interpretation:

- Both models produce consistent predictions.
- Differences may occur in more ambiguous images due to architecture differences.

4.3 Fine-tune a pretrained model on a small custom dataset

Screenshot:

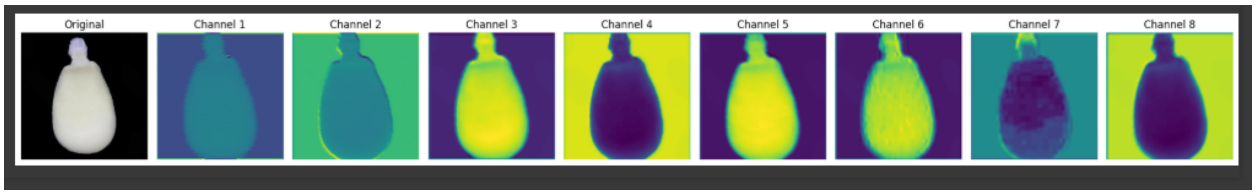


Interpretation:

- Fine-tuning adapts pretrained models to new classes.
- Even small datasets benefit from transfer learning.

4.4 Visualize activation maps / intermediate features

Screenshot:



Interpretation:

- Early layers capture edges, textures, and patterns.
- Visualization helps understand what the CNN “sees”.

5. Summary

- CNNs automatically extract features from images.
- Pretrained models like ResNet and MobileNet simplify deployment.
- PyTorch and TensorFlow provide versatile frameworks for vision tasks.
- OpenCV integration allows real-time or custom image inputs.
- Fine-tuning and activation visualization provide deeper understanding of model behavior.