Deep Learning Exercise Report

Isla Kim: shining04@csu.fullerton.edu
- 1. Simple deep learning exercise using CNN

Joanna A Menghamal: jmenghamal@csu.fullerton.edu
- 2. Text Extraction from images

Swayam Shree:sswayam@csu.fullerton.edu
- 3. Image Classification
- 4. Object Detection

1. Simple deep learning exercise using Convolutional Neural Networks (CNN)

(a) Testing the CNN program for digit recognition

: This program performs digit recognition using both the CPU and GPU. It separately displays the classification accuracy and training time for CPU and GPU executions. Google Colab was used to run the program on the GPU.

`L05_digit_CNN.py` result:

```
100%|██████████| 9.91M/9.91M [00:00<00:00, 16.0MB/s]
100%|██████████| 28.9k/28.9k [00:00<00:00, 484kB/s]
100%|██████████| 1.65M/1.65M [00:00<00:00, 4.49MB/s]
100%|██████████| 4.54k/4.54k [00:00<00:00, 7.91MB/s]
=== Running on CPU ===
[cpu] Epoch 1, Loss: 0.3091
[cpu] Epoch 2, Loss: 0.0828
[cpu] Epoch 3, Loss: 0.0622
[cpu] Epoch 4, Loss: 0.0496
[cpu] Epoch 5, Loss: 0.0415
[cpu] Epoch 6, Loss: 0.0341
[cpu] Epoch 7, Loss: 0.0303
[cpu] Epoch 8, Loss: 0.0279
[cpu] Epoch 9, Loss: 0.0228
[cpu] Epoch 10, Loss: 0.0214

Device: cpu
GPU Model: N/A
Test Accuracy: 98.82%
Training Time: 157.43 seconds
```

```
=== Running on GPU ===
[cuda] Epoch 1, Loss: 0.2887
[cuda] Epoch 2, Loss: 0.0831
[cuda] Epoch 3, Loss: 0.0600
[cuda] Epoch 4, Loss: 0.0500
[cuda] Epoch 5, Loss: 0.0414
[cuda] Epoch 6, Loss: 0.0355
[cuda] Epoch 7, Loss: 0.0302
[cuda] Epoch 8, Loss: 0.0270
[cuda] Epoch 9, Loss: 0.0229
[cuda] Epoch 10, Loss: 0.0198

Device: cuda
GPU Model: Tesla T4
Test Accuracy: 98.74%
Training Time: 118.25 seconds
```

(b) Testing the CNN program for digit recognition

: This program is an enhanced version of program (a), with additional layers and adjusted parameters to improve test accuracy. The model is trained over 20 epochs. As a result, an accuracy of 99.27% was achieved.

```
/Documents/GitHub/DL-project/1-DL-Exercise/improved_digit_cnn.py
Using device: cpu
Epoch 1/20
Train Loss: 0.3005 | Train Accuracy: 90.47%
Test Accuracy: 97.75% | Time: 99.24s

Epoch 2/20
Train Loss: 0.1339 | Train Accuracy: 95.93%
Test Accuracy: 98.23% | Time: 100.09s

Epoch 3/20
Train Loss: 0.1074 | Train Accuracy: 96.74%
Test Accuracy: 98.35% | Time: 817.16s

Epoch 4/20
Train Loss: 0.0943 | Train Accuracy: 97.25%
Test Accuracy: 98.42% | Time: 838.78s

Epoch 5/20
Train Loss: 0.0866 | Train Accuracy: 97.47%
Test Accuracy: 98.71% | Time: 94.09s

Epoch 6/20
Train Loss: 0.0801 | Train Accuracy: 97.60%
Test Accuracy: 98.88% | Time: 93.44s

Epoch 7/20
Train Loss: 0.0755 | Train Accuracy: 97.79%
Test Accuracy: 98.85% | Time: 93.69s

Epoch 8/20
Train Loss: 0.0729 | Train Accuracy: 97.84%
Test Accuracy: 99.00% | Time: 93.33s

Epoch 9/20
Train Loss: 0.0690 | Train Accuracy: 97.94%
Test Accuracy: 98.88% | Time: 92.89s

Epoch 10/20
Train Loss: 0.0619 | Train Accuracy: 98.14%
Test Accuracy: 98.81% | Time: 93.39s
```

```
Epoch 11/20
Train Loss: 0.0598 | Train Accuracy: 98.23%
Test Accuracy: 98.97% | Time: 93.04s

Epoch 12/20
Train Loss: 0.0578 | Train Accuracy: 98.36%
Test Accuracy: 99.17% | Time: 93.06s

Epoch 13/20
Train Loss: 0.0587 | Train Accuracy: 98.28%
Test Accuracy: 99.18% | Time: 92.97s

Epoch 14/20
Train Loss: 0.0512 | Train Accuracy: 98.50%
Test Accuracy: 99.16% | Time: 93.41s

Epoch 15/20
Train Loss: 0.0511 | Train Accuracy: 98.50%
Test Accuracy: 99.20% | Time: 93.45s

Epoch 16/20
Train Loss: 0.0490 | Train Accuracy: 98.57%
Test Accuracy: 99.14% | Time: 93.48s

Epoch 17/20
Train Loss: 0.0505 | Train Accuracy: 98.50%
Test Accuracy: 99.21% | Time: 93.42s

Epoch 18/20
Train Loss: 0.0471 | Train Accuracy: 98.64%
Test Accuracy: 99.29% | Time: 93.31s

Epoch 19/20
Train Loss: 0.0471 | Train Accuracy: 98.64%
Test Accuracy: 99.25% | Time: 93.24s

Epoch 20/20
Train Loss: 0.0434 | Train Accuracy: 98.67%
Test Accuracy: 99.27% | Time: 93.01s
```

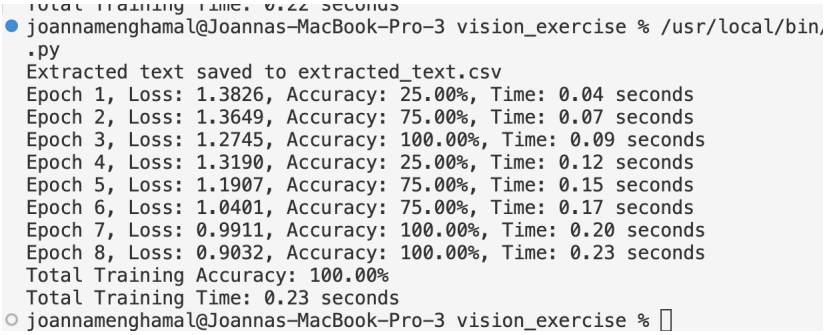
(c) Handwritten digit prediction

: This program uses a previously trained model to recognize handwritten images of the digits 3, 4, and 5 as digital numbers and outputs the results. It also saves the trained model. Since the MNIST image dataset consists of white digits on a black background, color inversion is applied to match the format of the training and testing data. The images are adjusted using transforms.

```
.venvcsuftitan@MacBook-Pro-145 DL-project % /Users/csuftitan/Documents/GitHub/DL-project/.venv/bin/python /Users/csuftitan
/Documents/GitHub/DL-project/1-DL-Exercise/predict_my_digits.py
Prediction for /Users/csuftitan/Documents/GitHub/DL-project/1-DL-Exercise/handwritten_digits/digit3.png: 3
Prediction for /Users/csuftitan/Documents/GitHub/DL-project/1-DL-Exercise/handwritten_digits/digit4.png: 4
Prediction for /Users/csuftitan/Documents/GitHub/DL-project/1-DL-Exercise/handwritten_digits/digit5.png: 5
```

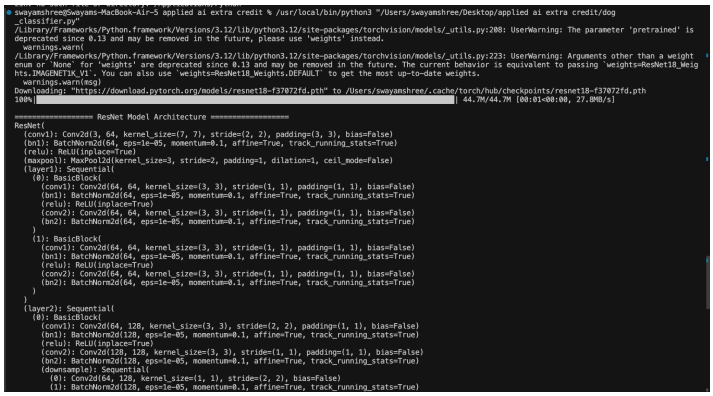
2.Text Extraction

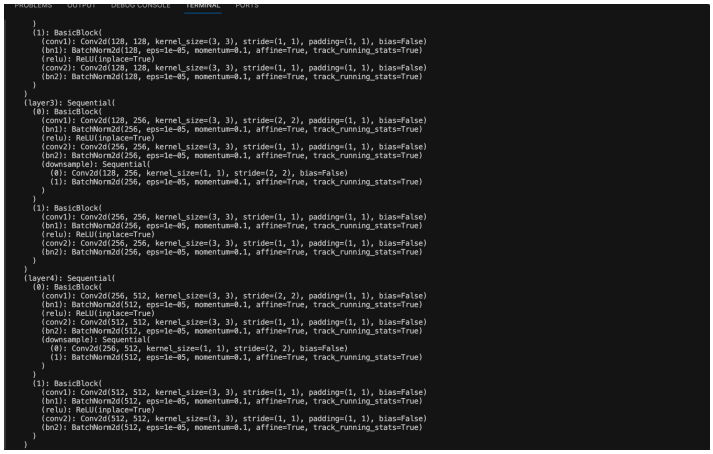
“text_extraction.py” loads and preprocesses the dataset “L05_DL_VISION_Receipts.zip” and preprocess the data, and train a Convolutional Neural Network (CNN) model to classify the receipts based on the extracted text. The project utilizes Optical Character Recognition (OCR) to read text from receipt images, stores the extracted text in a CSV format, and then uses a CNN to classify the receipts, evaluating the model's performance over multiple epochs. 8 epochs were used.



3. Image classification

“dog_classifier.py” loads and preprocesses the dataset “L05_DL_Vision_Dogs.zip” and classifies the dog breeds using a pretrained CNN model “ResNet”

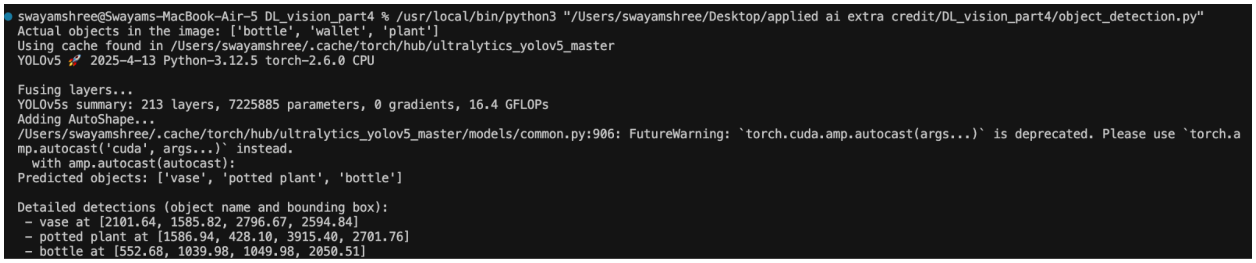





```
Total parameters in model:        11689512
Trainable parameters in model:    11689512

===== Starting Training for 20 epoch(s) =====
Epoch [1/20], Loss: 1.8631
Epoch [2/20], Loss: 1.7970
Epoch [3/20], Loss: 1.5415
Epoch [4/20], Loss: 1.4718
Epoch [5/20], Loss: 1.3207
Epoch [6/20], Loss: 1.1302
Epoch [7/20], Loss: 0.9511
Epoch [8/20], Loss: 0.9492
Epoch [9/20], Loss: 0.7141
Epoch [10/20], Loss: 0.5635
Epoch [11/20], Loss: 0.5935
Epoch [12/20], Loss: 0.6139
Epoch [13/20], Loss: 0.5941
Epoch [14/20], Loss: 0.3990
Epoch [15/20], Loss: 0.4130
Epoch [16/20], Loss: 0.4139
Epoch [17/20], Loss: 0.2524
Epoch [18/20], Loss: 0.3416
Epoch [19/20], Loss: 0.3439
Epoch [20/20], Loss: 0.3067
===== Training complete. Time elapsed: 268.27 seconds =====

Validation Accuracy: 85.71%
Total classification time (train + val): 268.27 seconds

Additional model components not discussed in detail might include:
 - conv1: Conv2d
 - bn1: BatchNorm2d
 - relu: ReLU
 - maxpool: MaxPool2d
 - layer1: Sequential
 - layer2: Sequential
 - layer3: Sequential
 - layer4: Sequential
 - avgpool: AdaptiveAvgPool2d
 - fc: Linear
```

4. Object detection

object_detection.py to detects all objects in the image using the pretrained “YOLO” object detection model. The program prints a list of object names (labels) actually in the image, a list of predicted object names, and their corresponding locations (coordinates).



The image contained plants, a wallet, and a bottle. The model accurately identifies the plants and the bottle. However, it includes a vase. This could be the pot in which the plants are stored, or the model could be identifying my wallet as a vase.