CSE4097.1

Introduction to Deep Learning

Convolutional Neural Network Homework

Enes Torluoğlu - 150121002

Mohamad Nael Ayoubi - 150120997

1. Purpose

We aim to design and train a CNN model that predicts the digit and its rotation by the given image. While we tried to keep our model as small and efficient as possible we tried to achieve best results. We used MNIST dataset with some modifications to fit our task, and torch library to write the neural network.

2. Dataset

Dataset: Modified version of MNIST from torchvision.datasets

Modifications:

Only selected images of digits; 1, 2, 3, 4, 5, 7

Each image is rotated by [0, 90, 180, 270] degrees

Each sample on dataset includes:

- Image: image of the digit
- Digit Label: Categorical label of the digits(Digits are mapped to 0-5)
- Rotation Label: Categorical label of the rotation(Rotations are mapped to 0-3)

3. Neural Network Architecture

1.Input:

- 1x28x28 (image)

2.conv1 Layer:

- 8x28x28 Output
- 3x3 Filters
- 8 Filters
- Stride 1
- Padding 1

3.pool Layer:

- 8x14x14 Output
- 2x2 Pooling
- Stride 2

4.conv2 Layer

- 16x14x14 Output
- 3x3 Filters
- 16 Filters
- Stride 1
- Padding 1

5.pool Layer:

- 16x7x7 Output
- 2x2 Pooling
- Stride 2

6.Flattening:

- 784 Output

7. Two Head Output Fully Connected Layers:

1.fc_digit:

- Outputs predicted digit

2.fc_rotation layer:

- Outputs predicted rotation

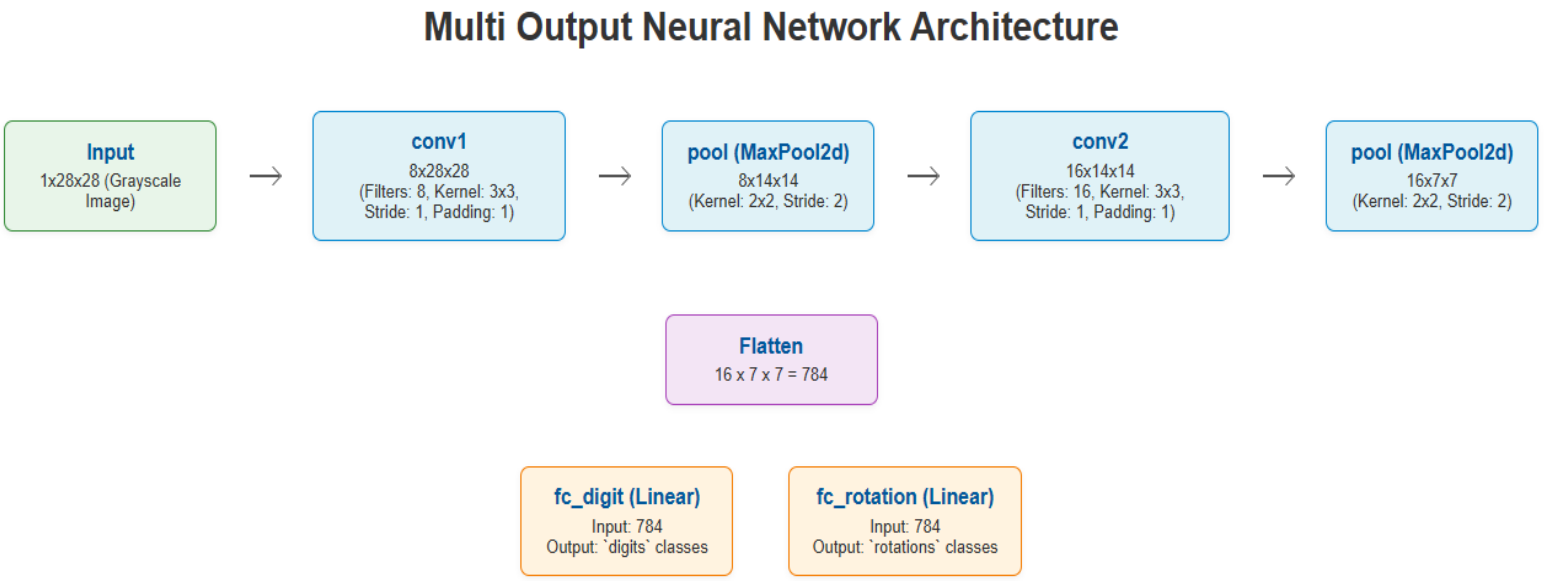

Figure 1: NeuralNetwork Architecture

4. Training Process

For training process we split our dataset into training and test sets with an 80/20 ratio. To ensure more balanced and representative splits, we used StratifiedShuffleSplit method. Thanks to this method we maintained the original proportion of each digits and rotations within both the training and test sets. Here are some values for the dataset:

Full Dataset Size: 145436

Train Set Distribution:

Total instances: 116348

Digits:

1 (label 0): 21572 instances (18.5%)

2 (label 1): 19064 instances (16.4%)

3 (label 2): 19620 instances (16.9%)

4 (label 3): 18696 instances (16.1%)

5 (label 4): 17348 instances (14.9%)

7 (label 5): 20048 instances (17.2%)

Rotations:

0° (label 0): 29087 instances (25.0%)

90° (label 1): 29087 instances (25.0%)

180° (label 2): 29087 instances (25.0%)

270° (label 3): 29087 instances (25.0%)

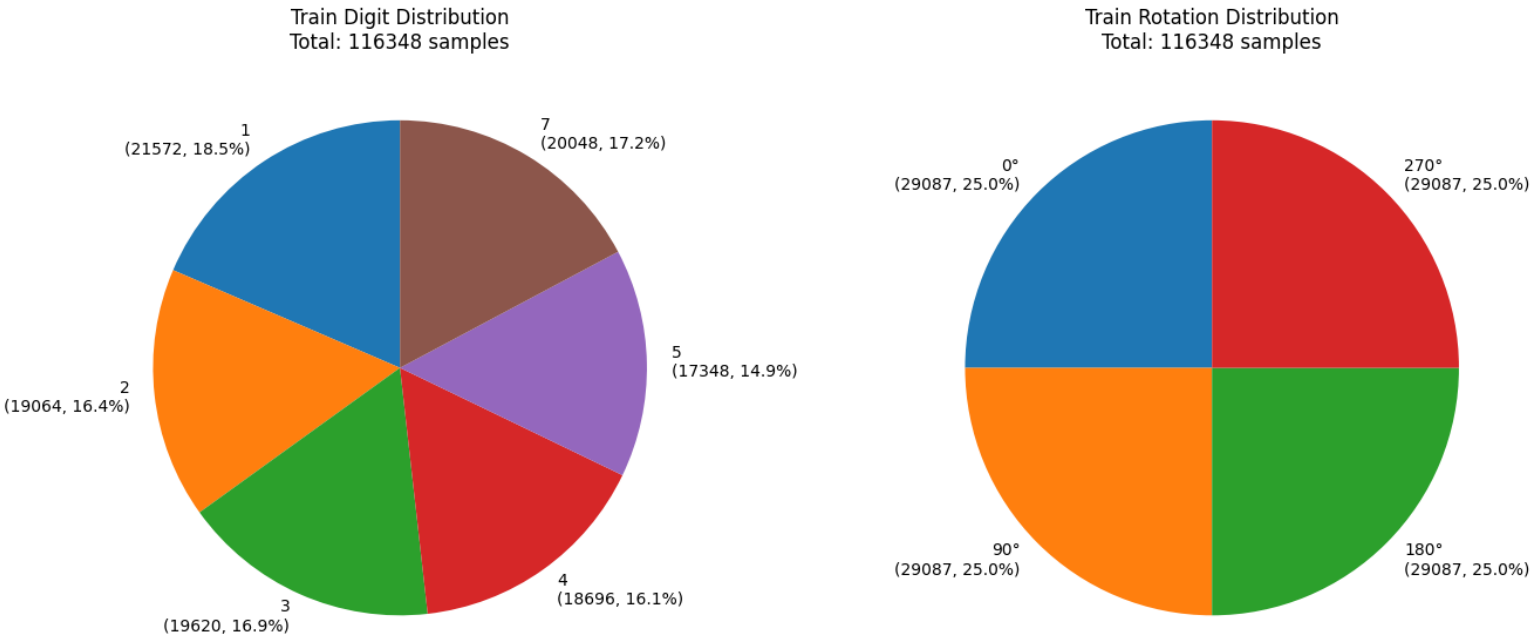

Figure 2: Train Data Distribution Charts

Test Set Distribution:

Total instances: 29088

Digits:

1 (label 0): 5396 instances (18.6%)

2 (label 1): 4768 instances (16.4%)

3 (label 2): 4904 instances (16.9%)

4 (label 3): 4672 instances (16.1%)

5 (label 4): 4336 instances (14.9%)

7 (label 5): 5012 instances (17.2%)

Rotations:

0° (label 0): 7272 instances (25.0%)

90° (label 1): 7272 instances (25.0%)

180° (label 2): 7272 instances (25.0%)

270° (label 3): 7272 instances (25.0%)

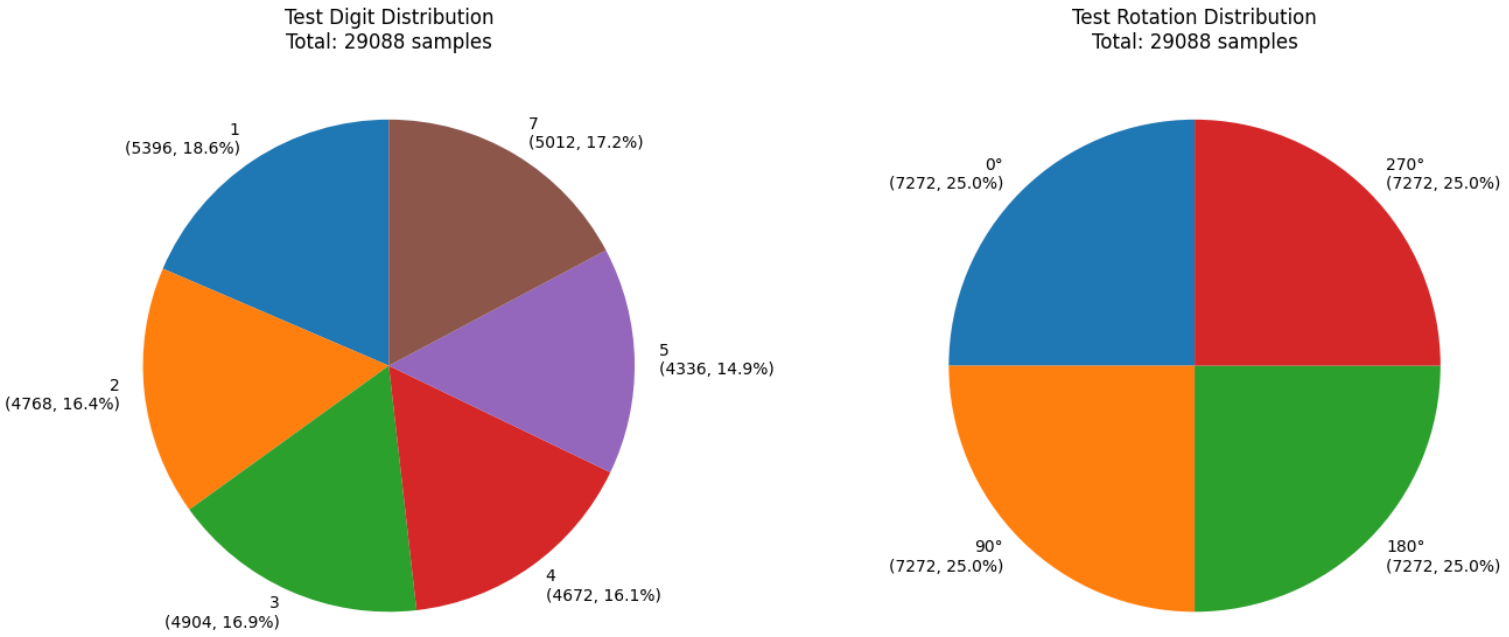

Figure 3: Train Data Distribution Charts

We trained our model for 20 epochs. Since we had 2 output heads, we had two losses, one for digits, one for rotations, then we combined into one loss to use for backpropogation. Also we used Adam method as optimizer. Here are some values for our losses through epochs:

Epoch 1/20 - Digit Loss: 0.2437, Rotation Loss: 0.0354, Total Loss: 0.2791

Epoch 2/20 - Digit Loss: 0.1367, Rotation Loss: 0.0019, Total Loss: 0.1386

Epoch 3/20 - Digit Loss: 0.0507, Rotation Loss: 0.0026, Total Loss: 0.0534

Epoch 4/20 - Digit Loss: 0.0819, Rotation Loss: 0.0540, Total Loss: 0.1359

Epoch 5/20 - Digit Loss: 0.1785, Rotation Loss: 0.0186, Total Loss: 0.1971

Epoch 6/20 - Digit Loss: 0.0928, Rotation Loss: 0.0316, Total Loss: 0.1244

Epoch 7/20 - Digit Loss: 0.0895, Rotation Loss: 0.0074, Total Loss: 0.0969

Epoch 8/20 - Digit Loss: 0.0514, Rotation Loss: 0.0247, Total Loss: 0.0761

Epoch 9/20 - Digit Loss: 0.0380, Rotation Loss: 0.0047, Total Loss: 0.0427

Epoch 10/20 - Digit Loss: 0.1909, Rotation Loss: 0.0009, Total Loss: 0.1917

Epoch 11/20 - Digit Loss: 0.2203, Rotation Loss: 0.0025, Total Loss: 0.2228

Epoch 12/20 - Digit Loss: 0.1973, Rotation Loss: 0.0004, Total Loss: 0.1976

Epoch 13/20 - Digit Loss: 0.0464, Rotation Loss: 0.0313, Total Loss: 0.0777

Epoch 14/20 - Digit Loss: 0.2496, Rotation Loss: 0.0169, Total Loss: 0.2664

Epoch 15/20 - Digit Loss: 0.0357, Rotation Loss: 0.0013, Total Loss: 0.0370

Epoch 16/20 - Digit Loss: 0.0887, Rotation Loss: 0.0049, Total Loss: 0.0936

Epoch 17/20 - Digit Loss: 0.0687, Rotation Loss: 0.0137, Total Loss: 0.0825

Epoch 18/20 - Digit Loss: 0.1646, Rotation Loss: 0.0004, Total Loss: 0.1650

Epoch 19/20 - Digit Loss: 0.2053, Rotation Loss: 0.0231, Total Loss: 0.2284

Epoch 20/20 - Digit Loss: 0.0956, Rotation Loss: 0.0016, Total Loss: 0.0972

5. Evaluation

Here are our evaluation results for the model. As you can see our model performs well in predicting rotations, with 99.52% accuracy even on test set. Meanwhile digit accuracy is at 96.68% on test set which is also solid accuracy.

--- Evaluating Model ---

--- Training Set Results ---

Digit Accuracy: 97.45%

Digit F1 Score (weighted): 0.9745

Digit F1 Score (macro): 0.9742

Rotation Accuracy: 99.76%

Rotation F1 Score (weighted): 0.9976

Rotation F1 Score (macro): 0.9976

Overall Accuracy: 97.26%

--- Test Set Results ---

Digit Accuracy: 96.68%

Digit F1 Score (weighted): 0.9668

Digit F1 Score (macro): 0.9664

Rotation Accuracy: 99.52%

Rotation F1 Score (weighted): 0.9952

Rotation F1 Score (macro): 0.9952

Overall Accuracy: 96.27%

Also here are some predictions from our model:

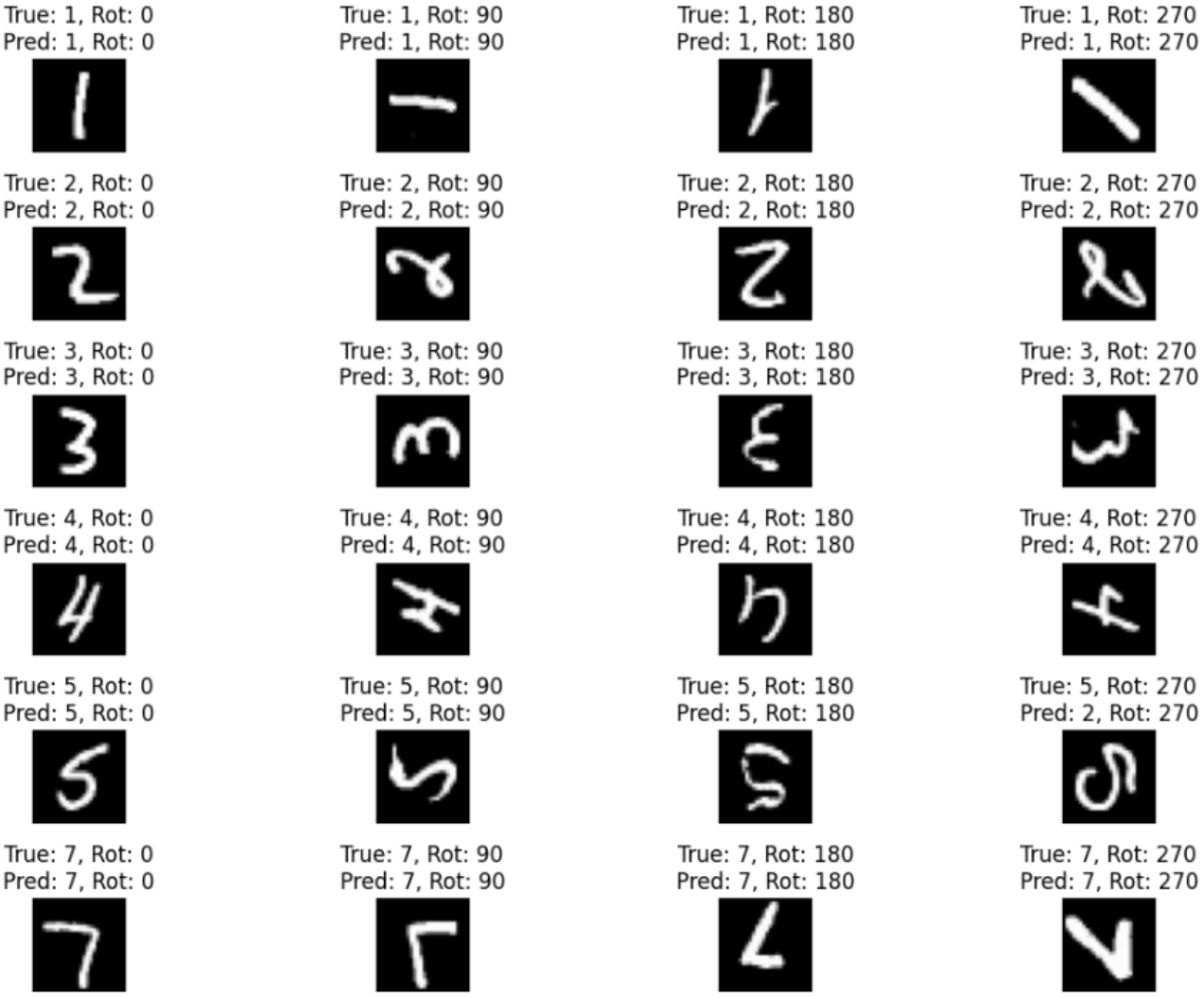

Figure 4: Prediction Samples