Saugat Malla

Project 5

Project Report

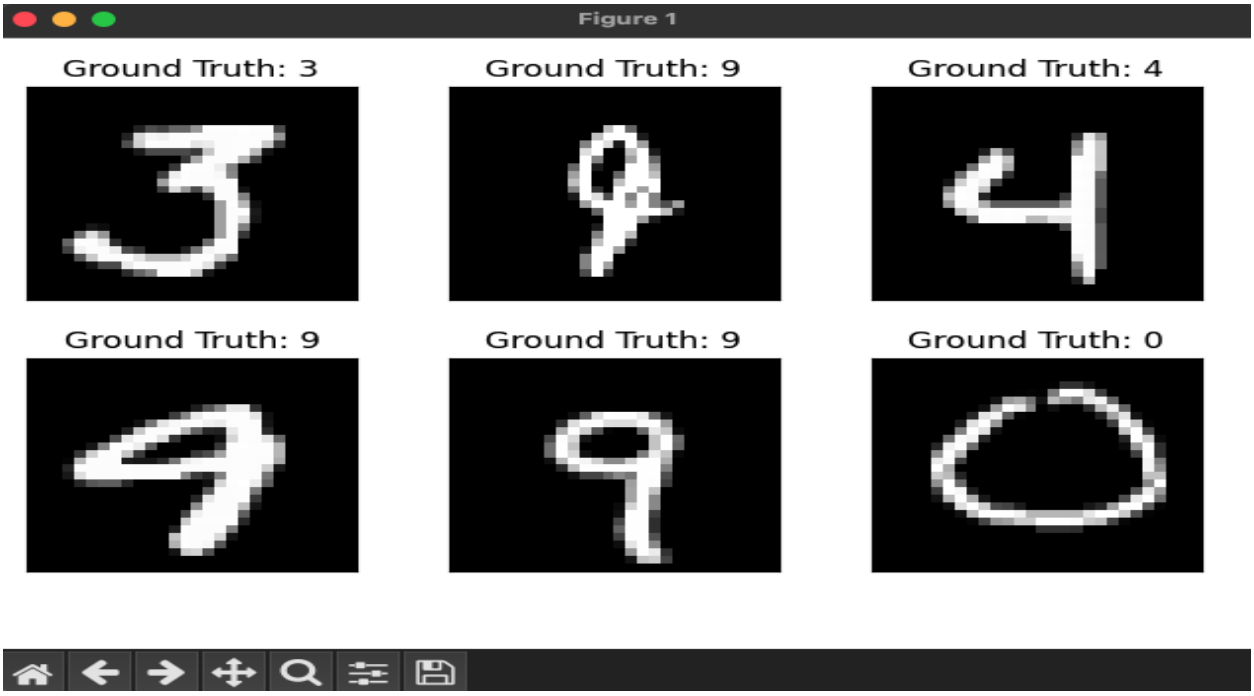
1. Description

The project involves building and training a neural network for digit recognition using the MNIST dataset. It begins with data acquisition and visualization, followed by network construction with specific layers. Training involves multiple epochs, with evaluation after each epoch. The trained network is saved for later use and tested on both the MNIST test set and handwritten digits. Analysis includes examining network structure, visualizing filters, and evaluating performance on Greek letters. Additionally, experimentation explores network architecture variations to optimize performance and training time. Overall, the project offers practical experience in deep learning model development, training, analysis, and experimentation.

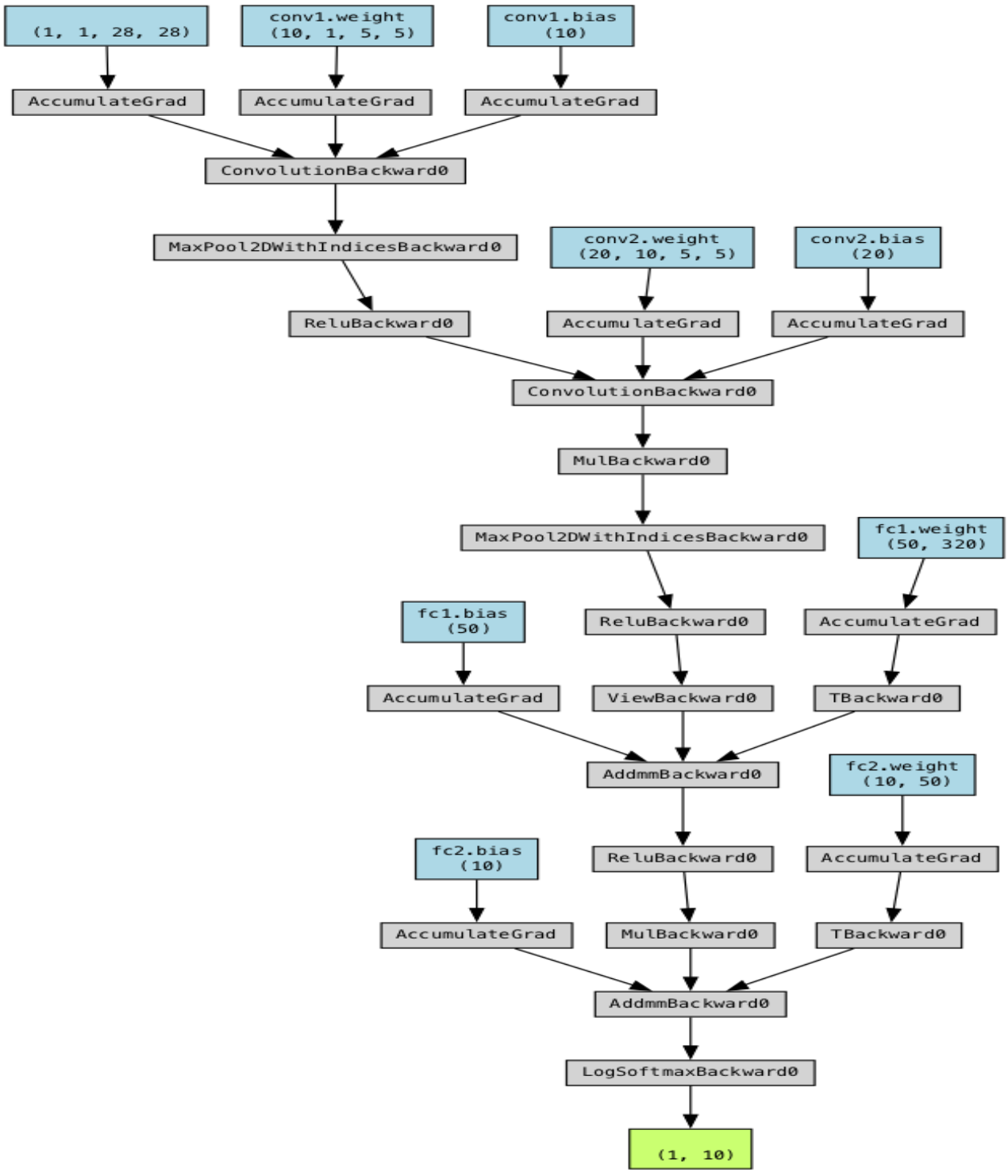
2. Images

a. Task 1

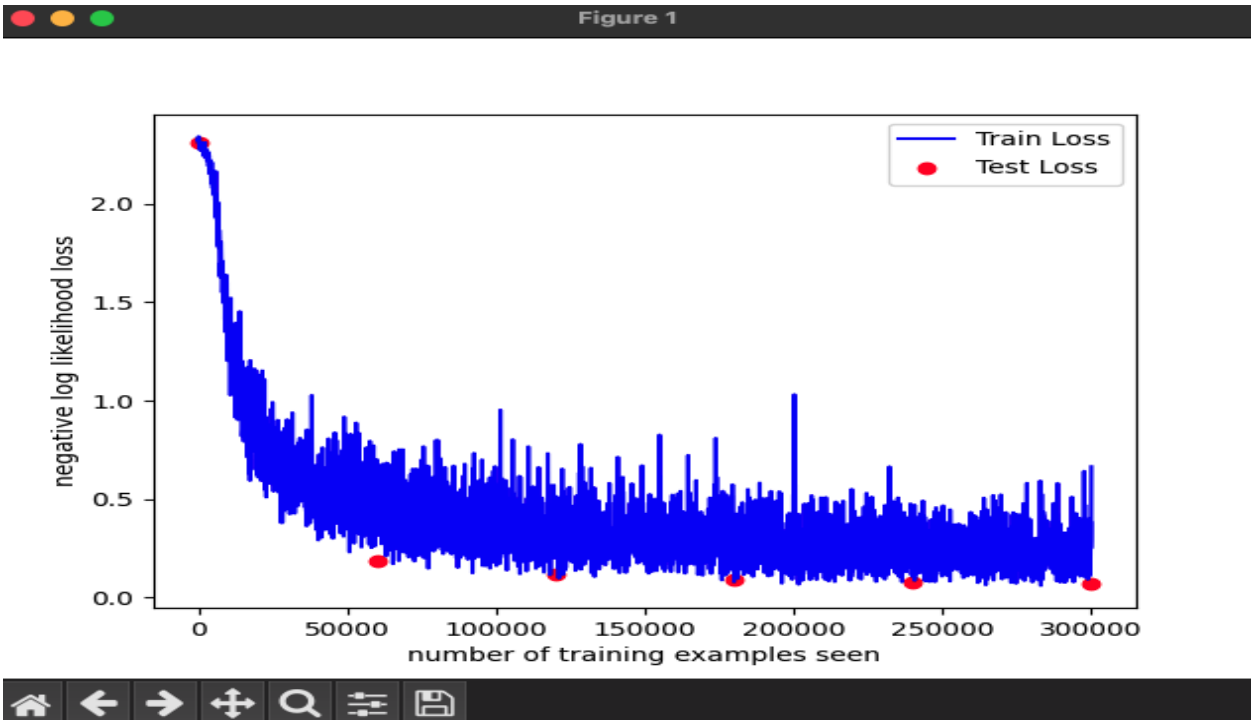
i. Image 1: Plot for six example digits



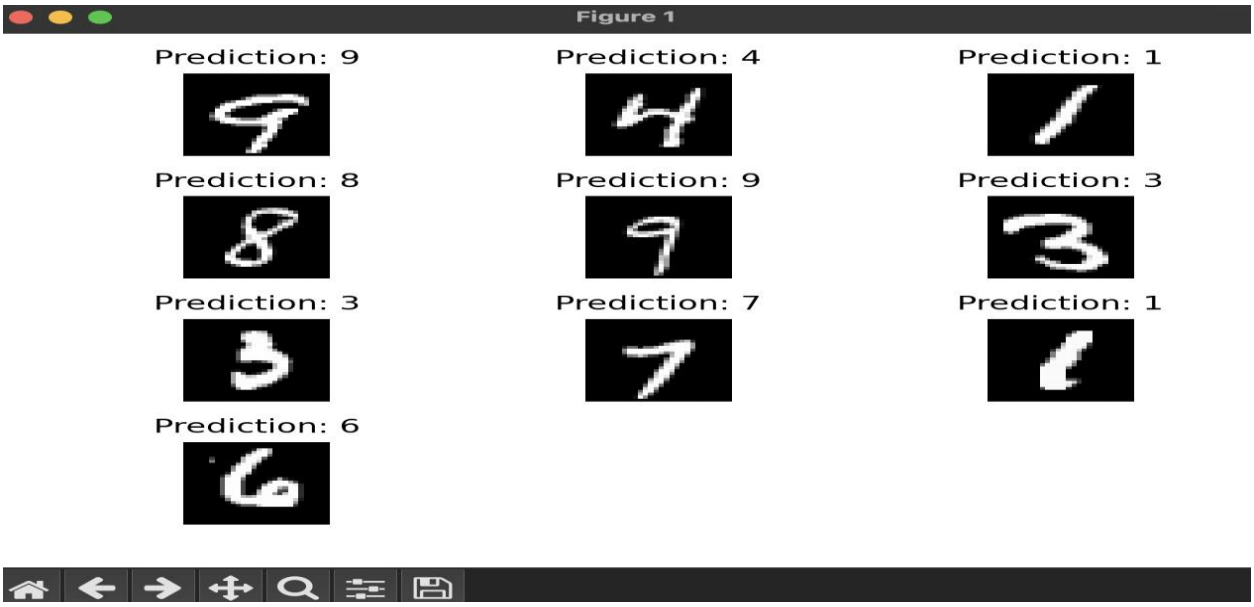
ii. Image 2: Diagram of the network



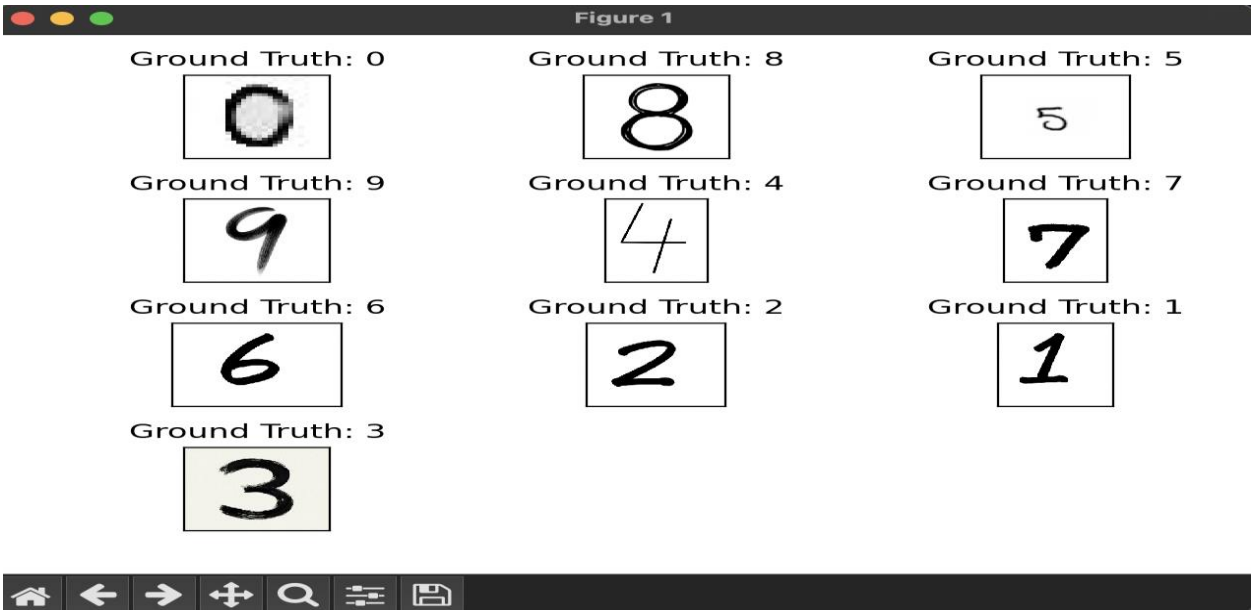
iii. Image 3: Accuracy score of training and test



iv. Image 4: Predicted values of 10 images

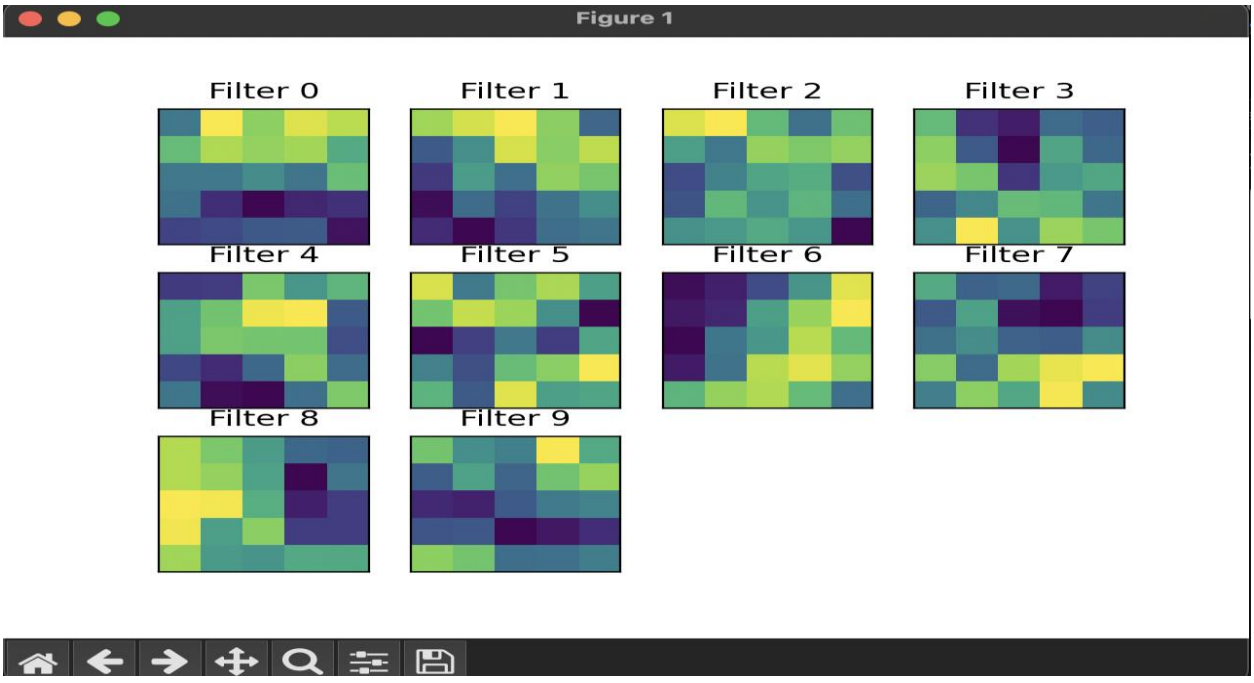


v. Image 5: Prediction on new images



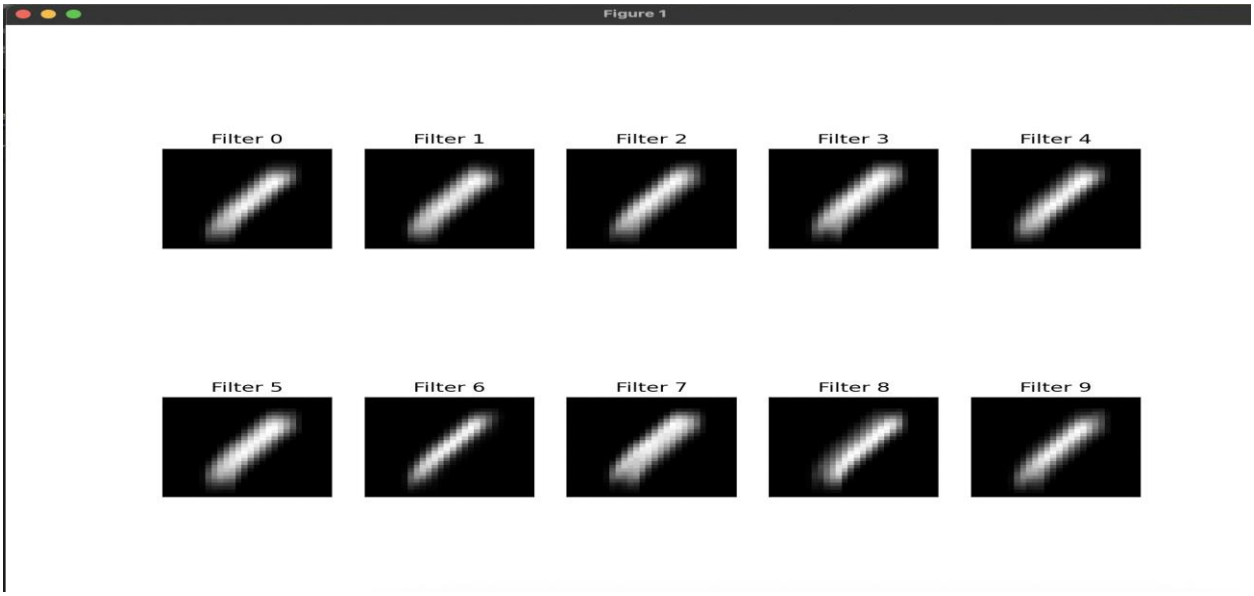
b. Task 2

i. Image 6: Filters visualized



ii. Image 7: Effects of filters

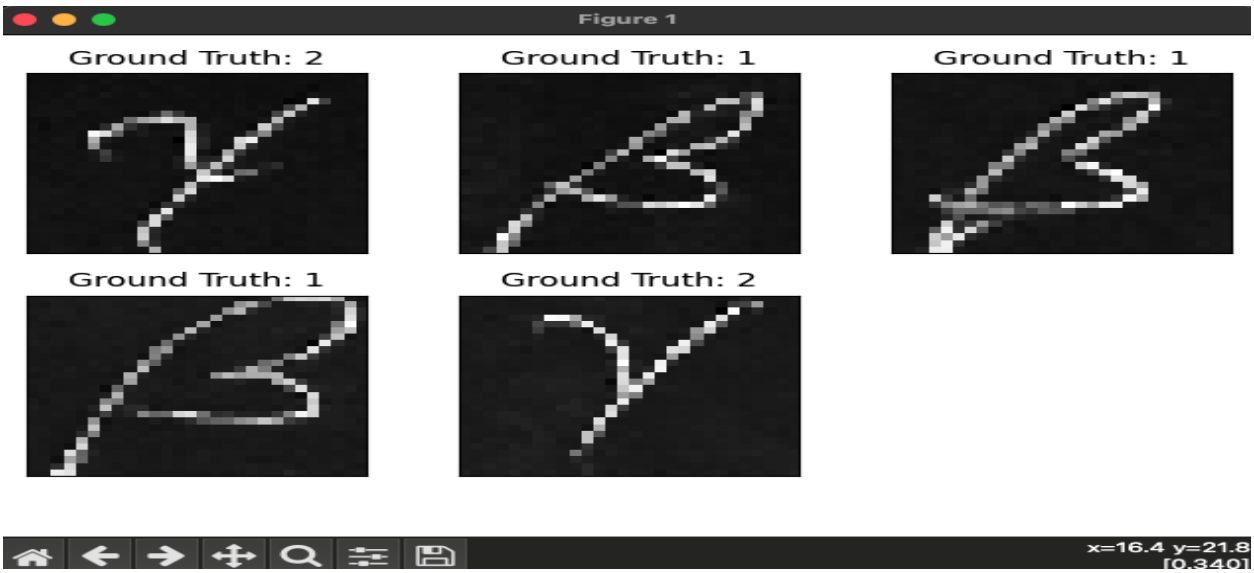
Note: Filters have varying effects on the image as seen in the figure below



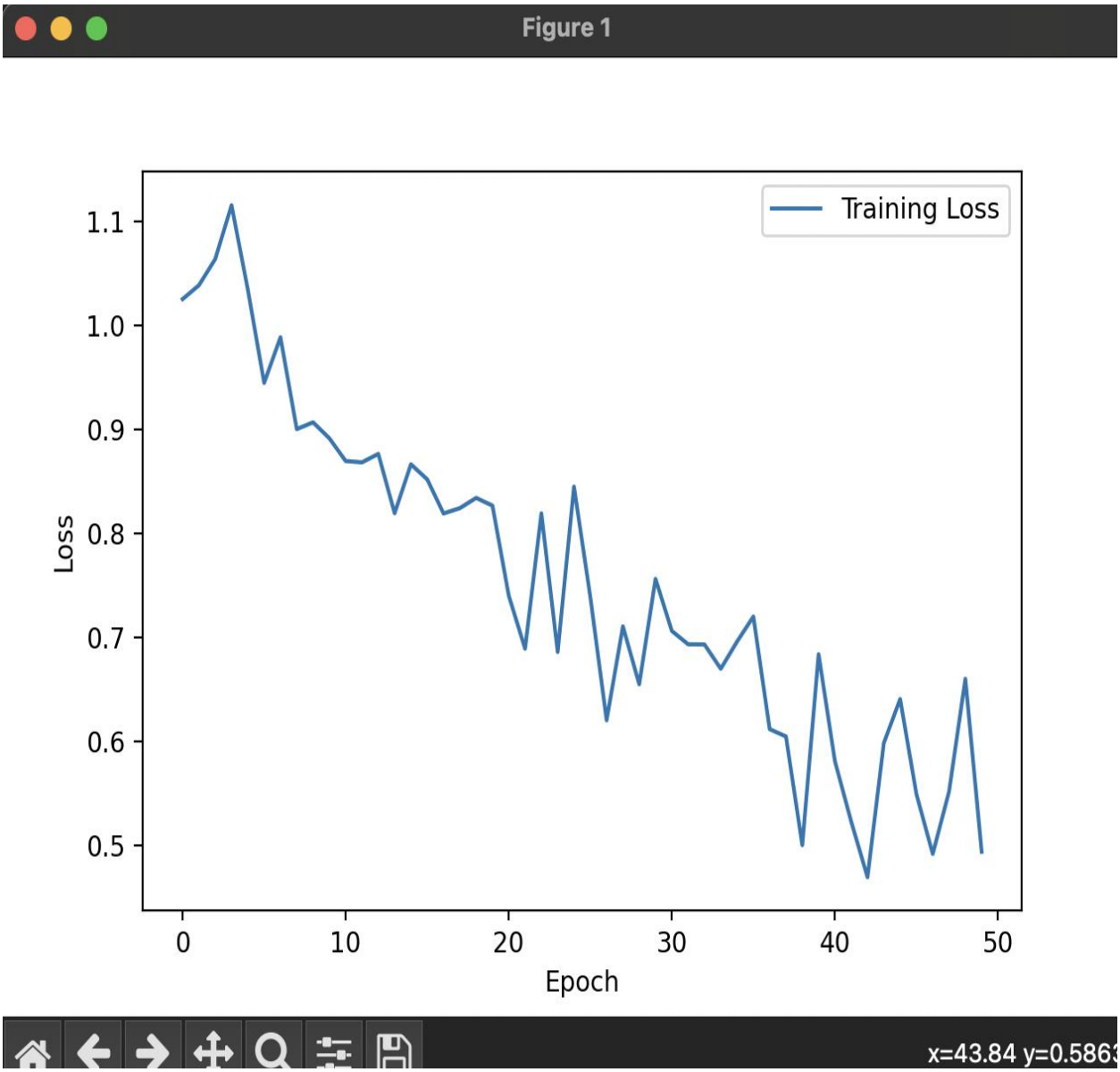
c. Task 3

Note: After around 50 epochs we get 97% accuracy

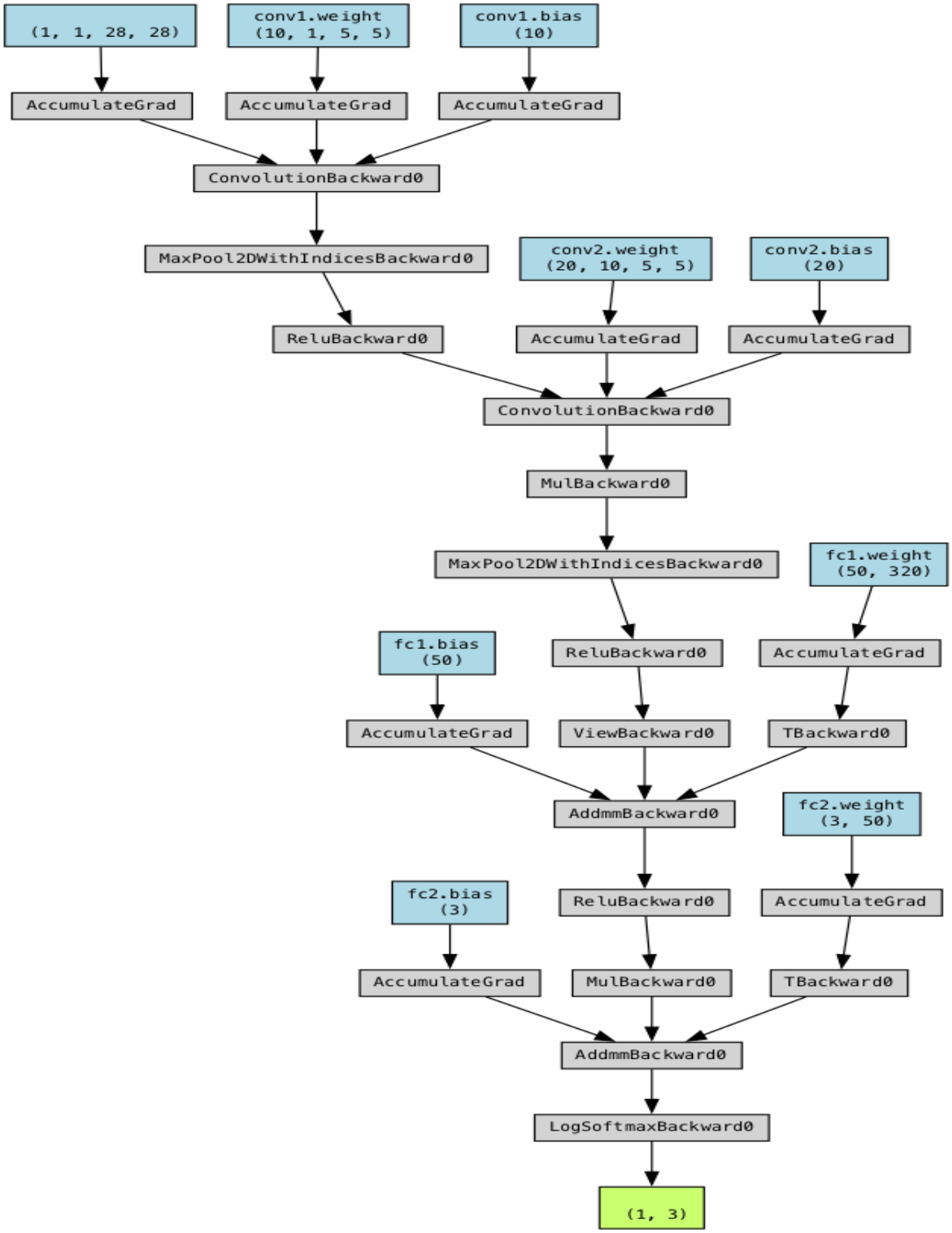
i. Image 8: Original images in dataset



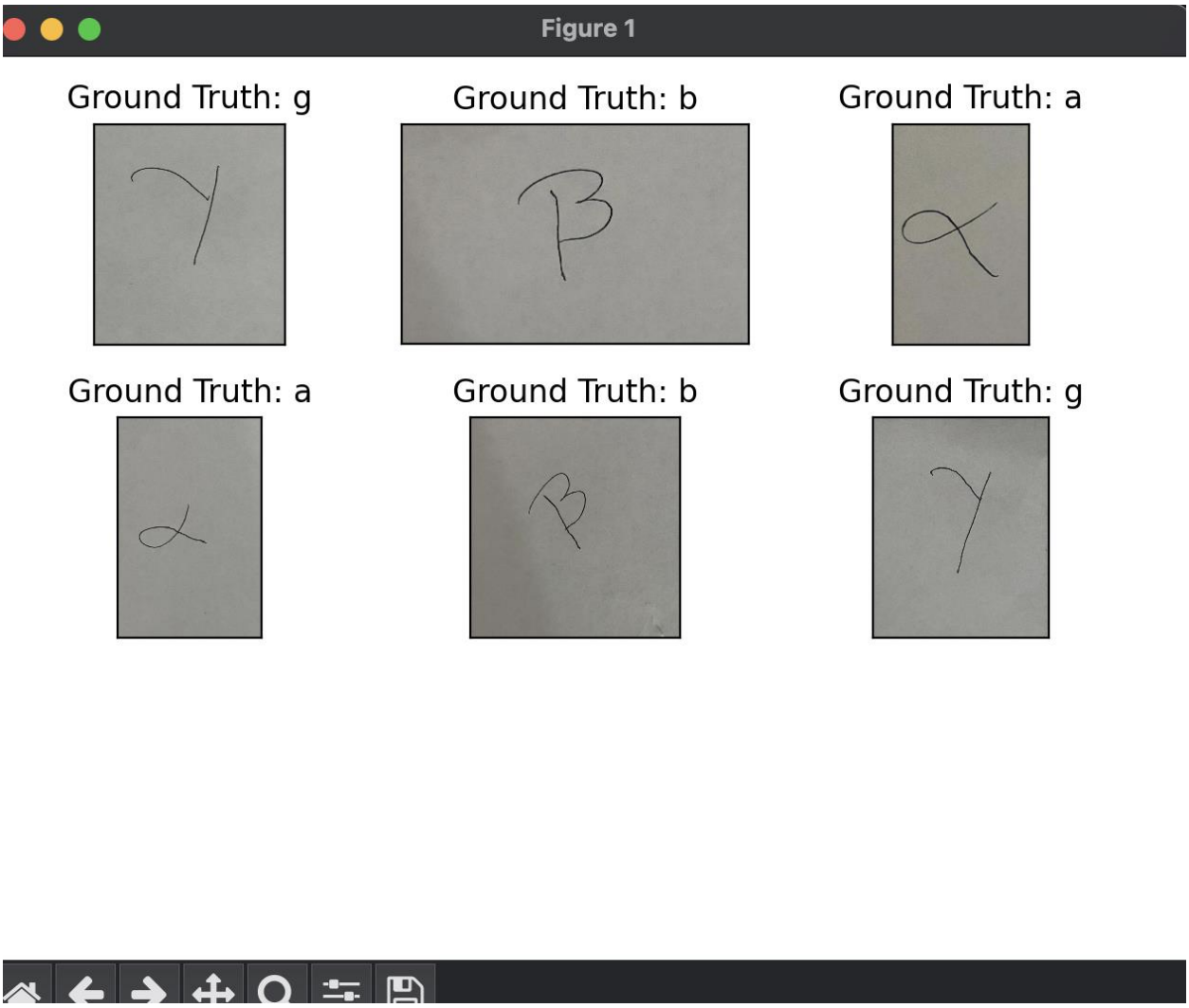
ii. Image 9: Training error



iii. Image 10: Modified Network



iv. Image 11: Result on additional data



d. Task 4

i. **Original Network**:

- **Dataset**: MNIST Fashion dataset consisting of grayscale images of fashion items.
  - **Network Architecture**:
    - Two convolutional layers with adjustable kernel sizes.
    - Max pooling layers.
    - Fully connected layers with customizable number of neurons.
    - Dropout layers with adjustable dropout rates.
  - **Hyperparameters**:
    - Number of epochs: 5
    - Batch size: 64
    - Learning rate: 0.01
    - Momentum: 0.5

II. **Dimensions Explored**:

- **Kernel Size**: [2, 3, 5]
- **Number of Neurons**: [20, 120, 150]
- **Dropout Rate**: [0.2, 0.3,0.5]
- **Learning Rate**: [0.01, 0.001, 0.001]
- Regularization: [‘None’, ‘dropout’]

III. Plan:

- **Variations**: Created 162 variations by combining different values for kernel size, number of neurons, dropout rate, learning rate and regularization.
- **Training and Evaluation**:
  - Trained each variation for 5 epochs using stochastic gradient descent with momentum.
  - Evaluated performance on the test set to calculate accuracy.
  - Recorded training loss and accuracy for each variation.

IV. Hypothesis:

- Increased size of kernels, neurons and lower learning rate should give better results.

V. **Results**:

- **Variation Performance**:
  - Achieved test accuracies ranging from 84% to 88%.
  - Training losses varied between 0.2 and 0.5.
- **Impact of Dimensions**:
  - Kernel Size: Larger kernels tended to yield slightly higher accuracies but increased computational cost.
  - Number of Neurons: No significant impact observed on accuracy.

- Dropout Rate: Higher dropout rates generally led to better generalization.
- Learning Rate: Lower learning rates resulted in slower convergence but potentially higher final accuracies.

VI. Conclusion:

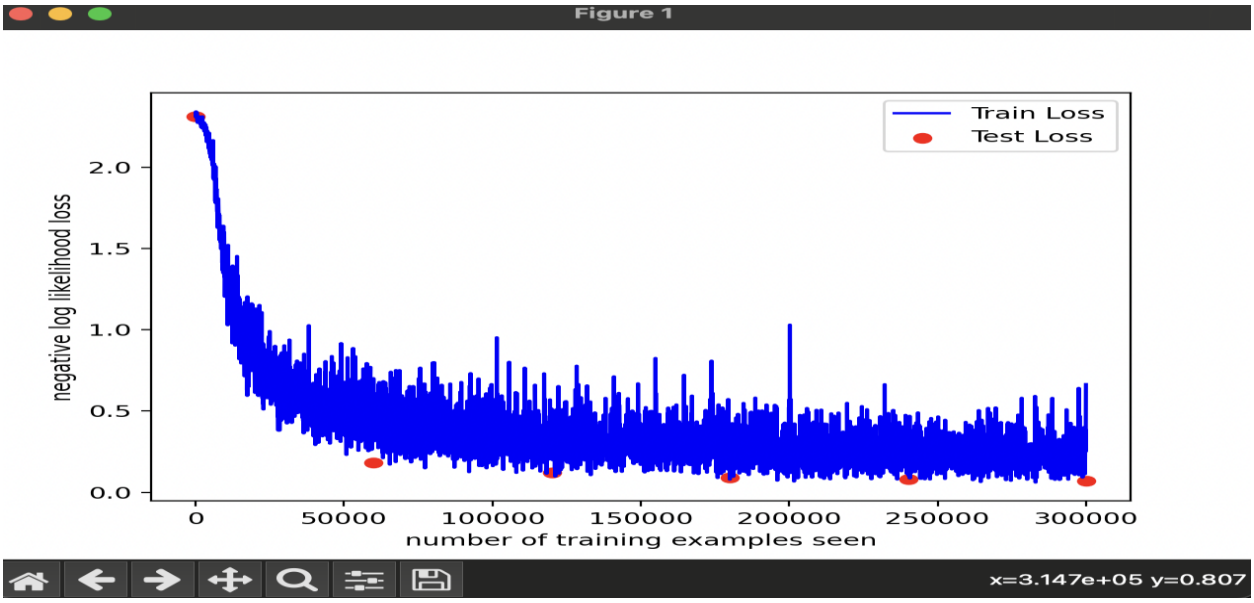
- **Optimal Configuration**: A kernel size of 5x5, dropout rate of 0.3, and learning rate of 0.01 yielded the best overall performance.
- **Trade-offs**: Balancing accuracy and computational cost are crucial, with larger kernel sizes offering marginal improvements at the expense of increased complexity.

VII. Images

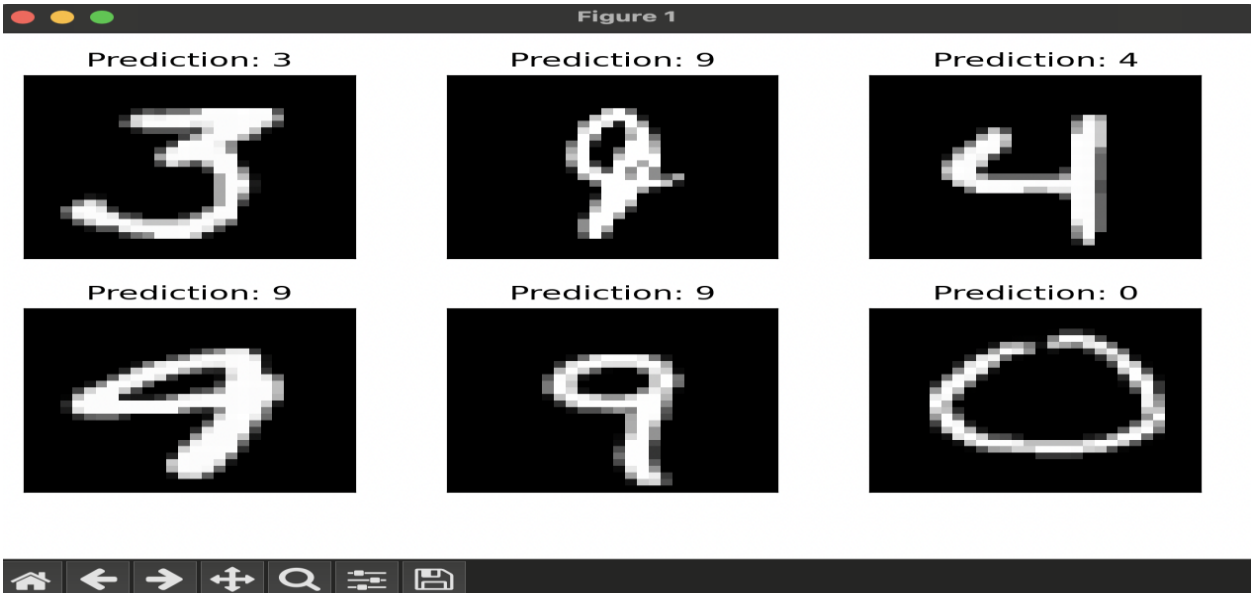
- Image: Result showing the best variation

```
Rank 1:
{'num_layers': 5, 'num_neurons': 120, 'dropout_rate': 0.3, 'learning_rate': 0.01, 'regularization': None}
```

3. Extensions
    a. Extension 1: I have used 2 additional dimensions in task 2
    b. Extension 2: Replaced first layer of Neural Network with Gabor filter
        i. Image 1: Training loss and accuracy plot



ii. Image 2: Predictions



iii. Image 3: Loss and Accuracy after 5 epochs

```
Test set: Avg. loss: 0.0690, Accuracy: 9764/10000 (98%)
```

4. Reflections

Building and training the digit recognition neural network using the MNIST dataset was an enriching experience, providing insights into deep learning methodologies and practical implementations. Working with PyTorch enabled hands-on exploration of network architectures and training procedures. Visualizing the dataset and designing the network architecture enhanced understanding of data preprocessing and model construction. Training and evaluating the network's performance on training and test sets illuminated the optimization process and model generalization. Exploring transfer learning for Greek letters expanded the project's scope, showcasing the adaptability of pre-trained models. Overall, this project deepened my understanding of deep learning principles, bolstered my skills in neural network development, and underscored the significance of image recognition in real-world scenarios.

5. Acknowledgements

Professor: Bruce Maxwell

TA’s: Zhizhou Gu, Poorna Chandra Vemula, Sasank Potluri