

```
In [1]: from data_loader import load_mnist, show_image
import os
current_dir = os.getcwd()
project_root = os.path.dirname(current_dir)
data_dir = os.path.join(project_root, 'data')
train_images, train_labels = load_mnist(data_dir, kind='train')
test_images, test_labels = load_mnist(data_dir, kind='t10k')
show_image(train_images, train_labels, test_images, test_labels)
```

```
+-----+-----+-----+
| Dataset | Images Shape | Labels Shape |
+-----+-----+-----+
| Train   | (60000, 784) | (60000,)      |
| Test    | (10000, 784) | (10000,)      |
+-----+-----+-----+
```

Train Images (First 30)



```
In [2]: from cnn_model import CNN
model = CNN()
model.print_model()
```

| Layer (type) | Output Shape | Param # |
|----------------|--------------|---------|
| Conv2D | (6, 28, 28) | 156 |
| BatchNorm | (6, 28, 28) | 12 |
| ReLU | (6, 28, 28) | 0 |
| MaxPool2D | (6, 14, 14) | 0 |
| Conv2D | (16, 10, 10) | 2416 |
| BatchNorm | (16, 10, 10) | 32 |
| ReLU | (16, 10, 10) | 0 |
| MaxPool2D | (16, 5, 5) | 0 |
| FullyConnected | (120,) | 48120 |
| ReLU | (120,) | 0 |
| FullyConnected | (84,) | 10164 |
| ReLU | (84,) | 0 |
| FullyConnected | (10,) | 850 |

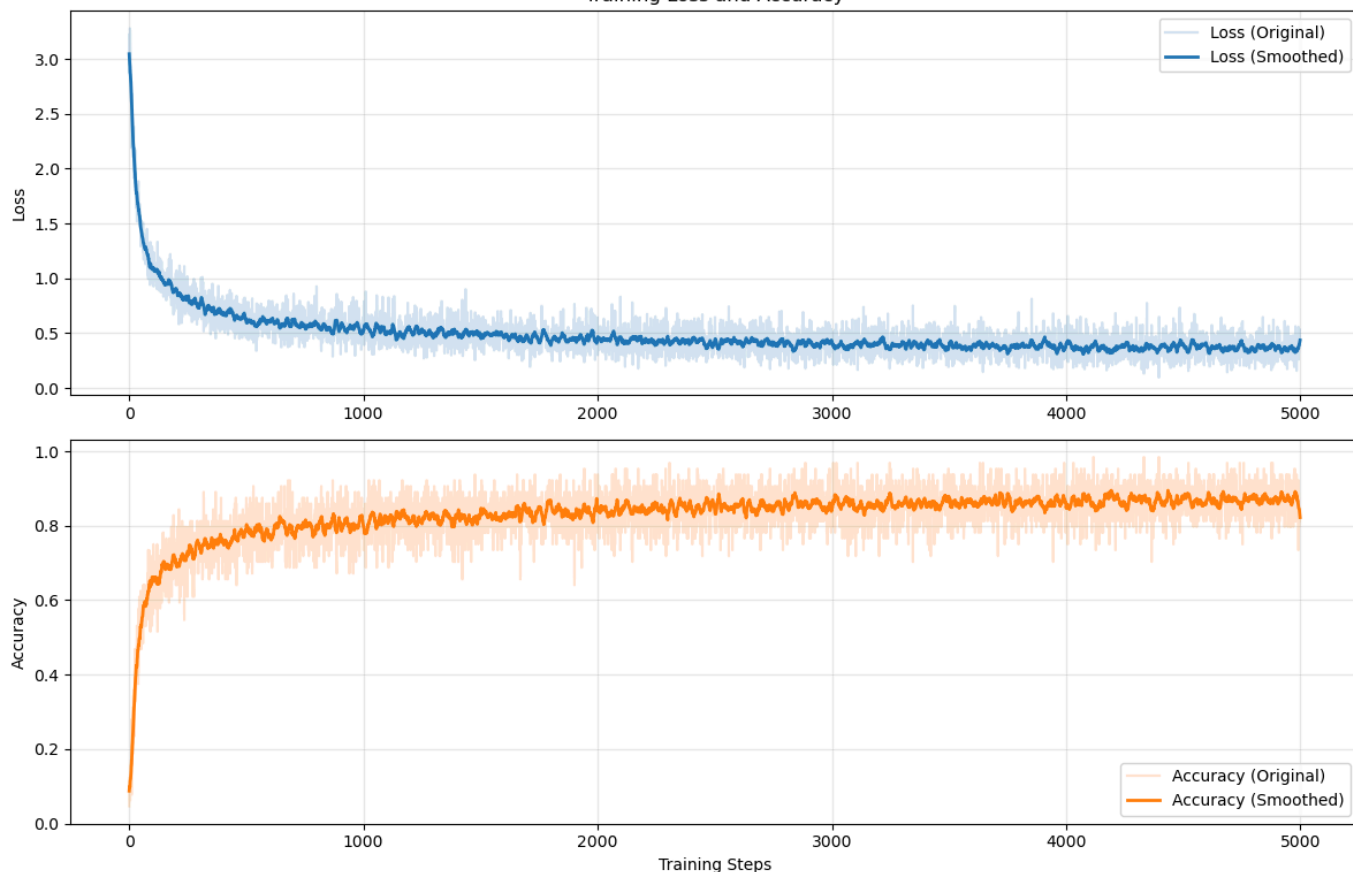
=====
Total params: 61750

```
In [ ]: from train import train
max_steps = 5000
batch_size = 64
learning_rate = 0.0005
train(train_images, train_labels, model, max_steps, batch_size, learning_rate)
```

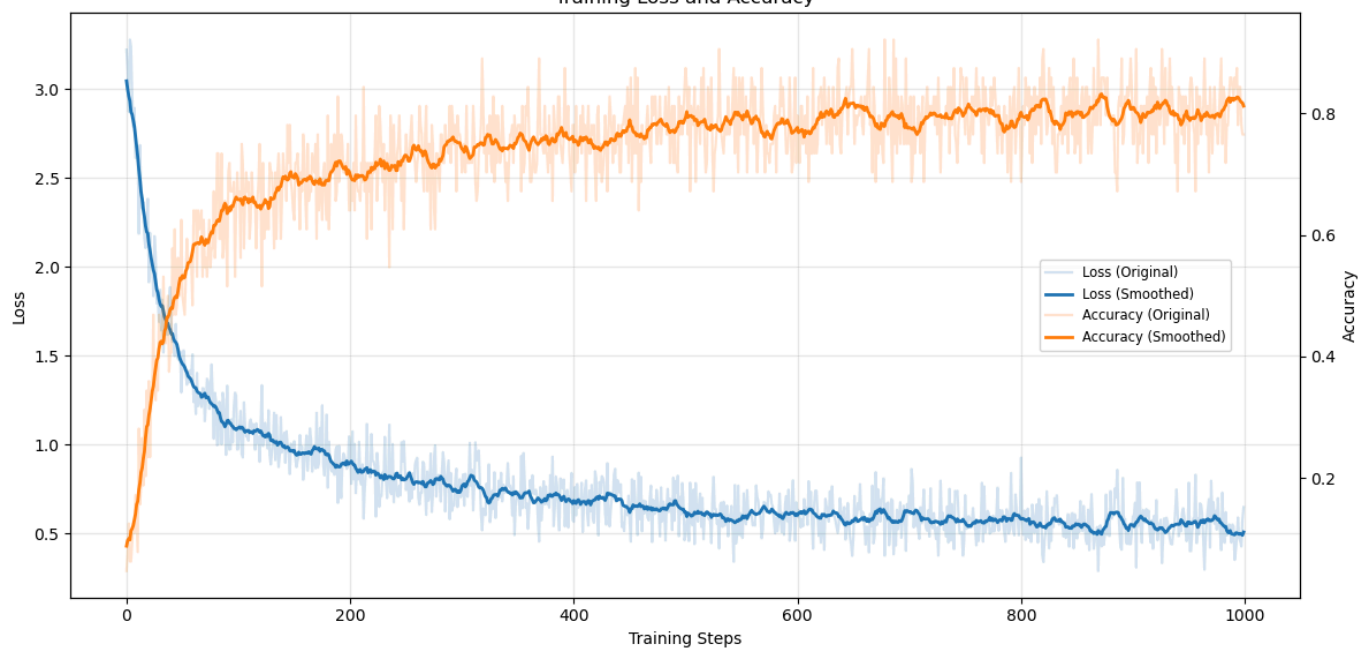
```
In [3]: from plot import *
log_dir = os.path.join(current_dir, 'logs')
losses = np.load(os.path.join(log_dir, 'losses.npy'))
accuracies = np.load(os.path.join(log_dir, 'accuracies.npy'))
plot_loss_accuracy1(losses=losses, accuracies=accuracies)
plot_loss_accuracy2(losses=losses, accuracies=accuracies)
losses = losses[0:1000]
accuracies = accuracies[0:1000]
plot_loss_accuracy1(losses=losses, accuracies=accuracies)
plot_loss_accuracy2(losses=losses, accuracies=accuracies)
```

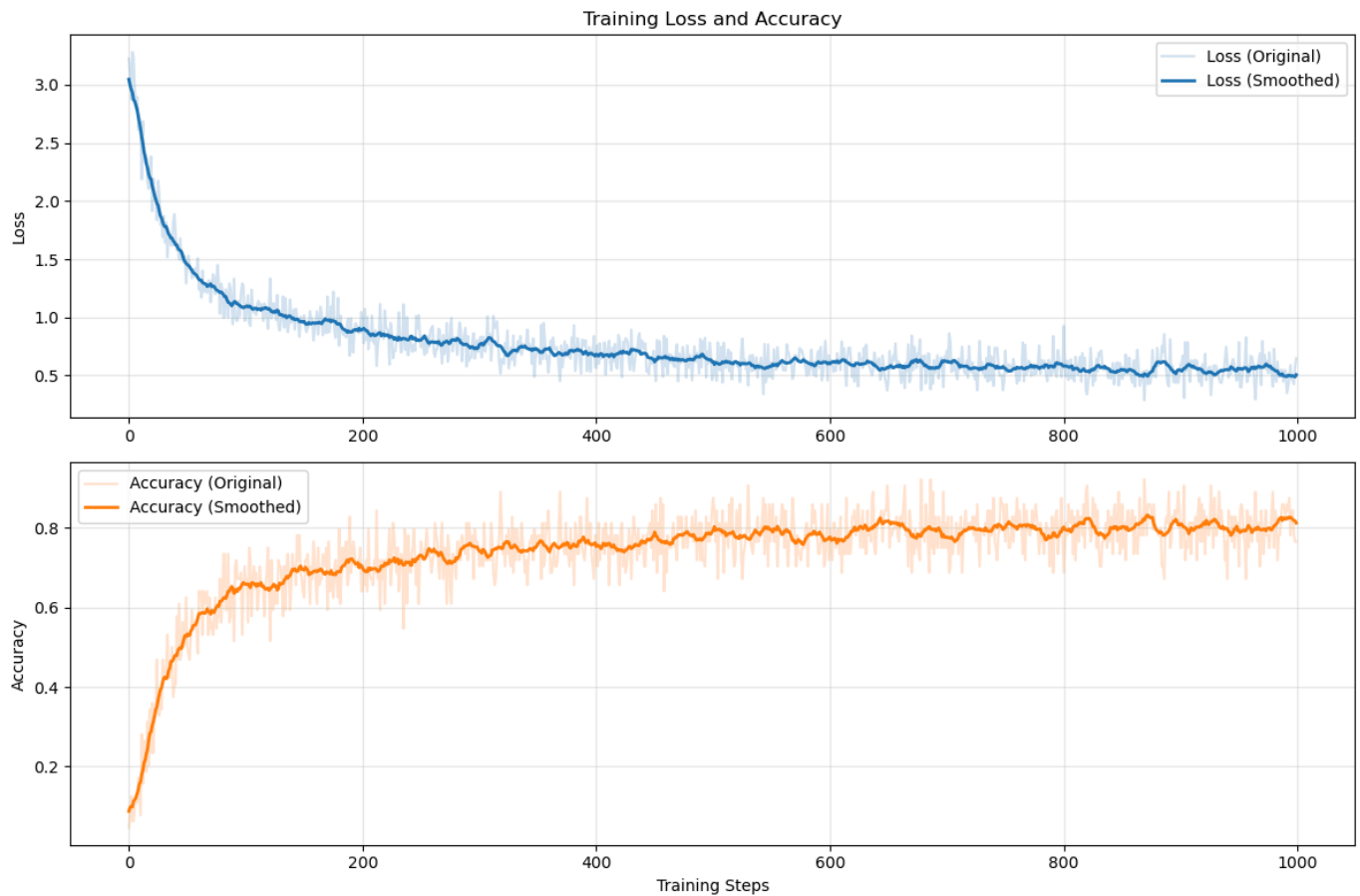


Training Loss and Accuracy



Training Loss and Accuracy





```
In [4]: from predict import predict
# test_mask = np.random.choice(len(test_images), 1000, replace=False)
# test_images = test_images[test_mask]
# test_labels = test_labels[test_mask]
predict(test_images=test_images, test_labels=test_labels)
```

model accuracy

=====

model_step1000.npz 0.8016

model_step2000.npz 0.8264

model_step3000.npz 0.8462

model_step4000.npz 0.8534

model_step5000.npz 0.8612

