香港中文大學
The Chinese University of Hong Kong

CMSC 5707 Advanced Topics in AI

Assignment 2b
Neural network programming

Student Name: **YANG Jingfan**

Student ID: **1155177962**

Instructor: **Dr. WONG Kin Hong**

3 Nov 2022

Contents

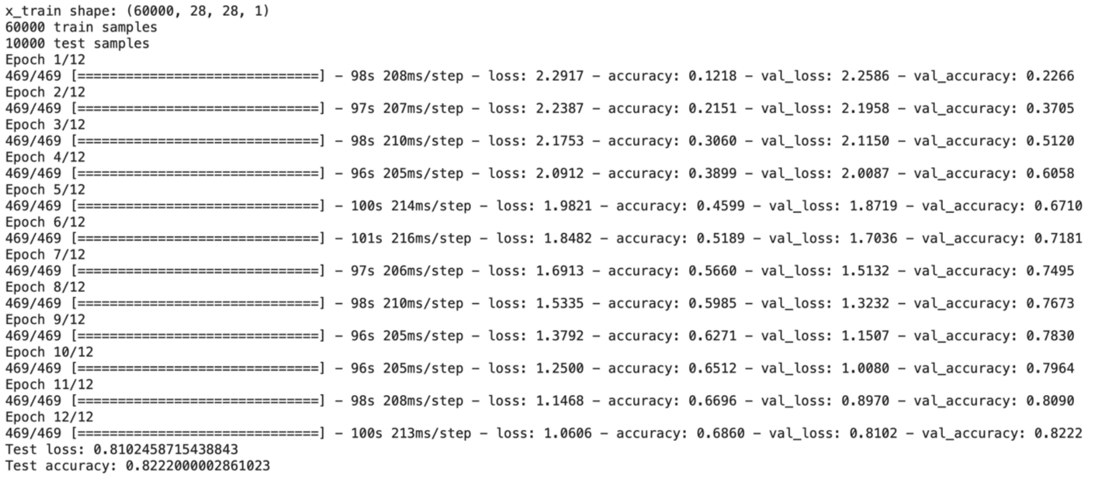
1 Introduction

This report provides a modified CNN model with better character recognition performance on the MNIST dataset. It also introduces the main methods used in modifying this model and gives a discussion about factors that affect recognition accuracy.

2 Experiments

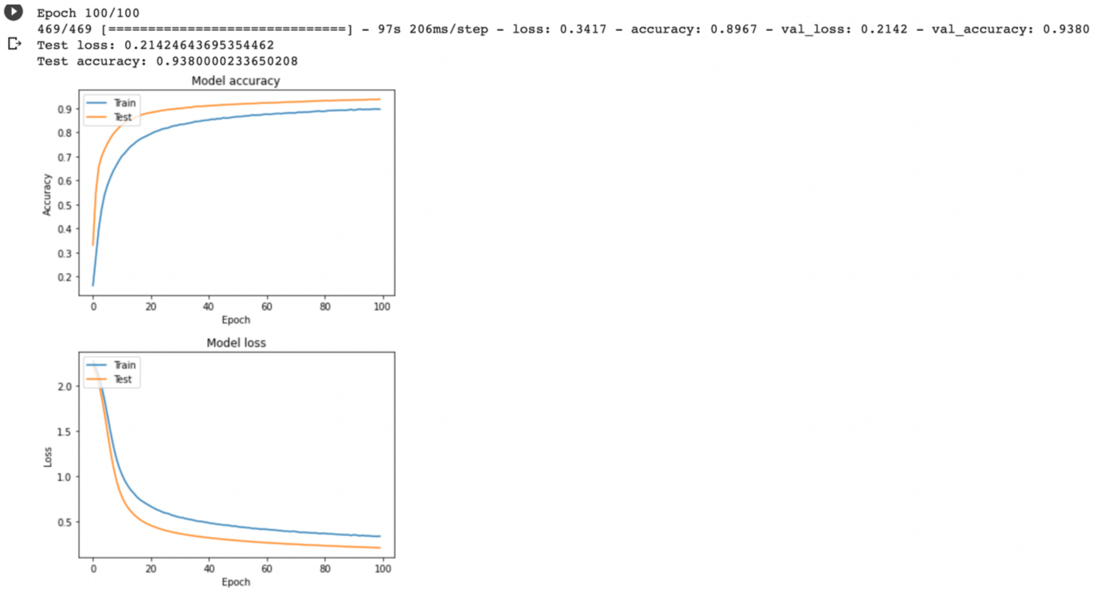
2.1 Effect of epoch on performance

An epoch means training the neural network with all the training data for one cycle. An epoch uses all the data exactly once. A forward pass and a backward pass together are counted as one pass. In this experiment, make sure all other factors do not change and only change the number of epochs, the accuracy of recognition will first increase and then begin to decrease. Figure 1 and figure 2 show the result, the number of epochs respectively equals 12 and 100.



```
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Epoch 1/12
469/469 [==============================] - 98s 208ms/step - loss: 2.2917 - accuracy: 0.1218 - val_loss: 2.2586 - val_accuracy: 0.2266
Epoch 2/12
469/469 [==============================] - 97s 207ms/step - loss: 2.2387 - accuracy: 0.2151 - val_loss: 2.1958 - val_accuracy: 0.3705
Epoch 3/12
469/469 [==============================] - 98s 210ms/step - loss: 2.1753 - accuracy: 0.3060 - val_loss: 2.1150 - val_accuracy: 0.5120
Epoch 4/12
469/469 [==============================] - 96s 205ms/step - loss: 2.0912 - accuracy: 0.3899 - val_loss: 2.0087 - val_accuracy: 0.6058
Epoch 5/12
469/469 [==============================] - 100s 214ms/step - loss: 1.9821 - accuracy: 0.4599 - val_loss: 1.8719 - val_accuracy: 0.6710
Epoch 6/12
469/469 [==============================] - 101s 216ms/step - loss: 1.8482 - accuracy: 0.5189 - val_loss: 1.7036 - val_accuracy: 0.7181
Epoch 7/12
469/469 [==============================] - 97s 206ms/step - loss: 1.6913 - accuracy: 0.5660 - val_loss: 1.5132 - val_accuracy: 0.7495
Epoch 8/12
469/469 [==============================] - 98s 210ms/step - loss: 1.5335 - accuracy: 0.5985 - val_loss: 1.3232 - val_accuracy: 0.7673
Epoch 9/12
469/469 [==============================] - 96s 205ms/step - loss: 1.3792 - accuracy: 0.6271 - val_loss: 1.1507 - val_accuracy: 0.7830
Epoch 10/12
469/469 [==============================] - 96s 205ms/step - loss: 1.2500 - accuracy: 0.6512 - val_loss: 1.0080 - val_accuracy: 0.7964
Epoch 11/12
469/469 [==============================] - 98s 208ms/step - loss: 1.1468 - accuracy: 0.6696 - val_loss: 0.8970 - val_accuracy: 0.8090
Epoch 12/12
469/469 [==============================] - 100s 213ms/step - loss: 1.0606 - accuracy: 0.6860 - val_loss: 0.8102 - val_accuracy: 0.8222
Test loss: 0.8102458715438843
Test accuracy: 0.8222000002861023
```

Figure 1 performance of epoch equals to 12

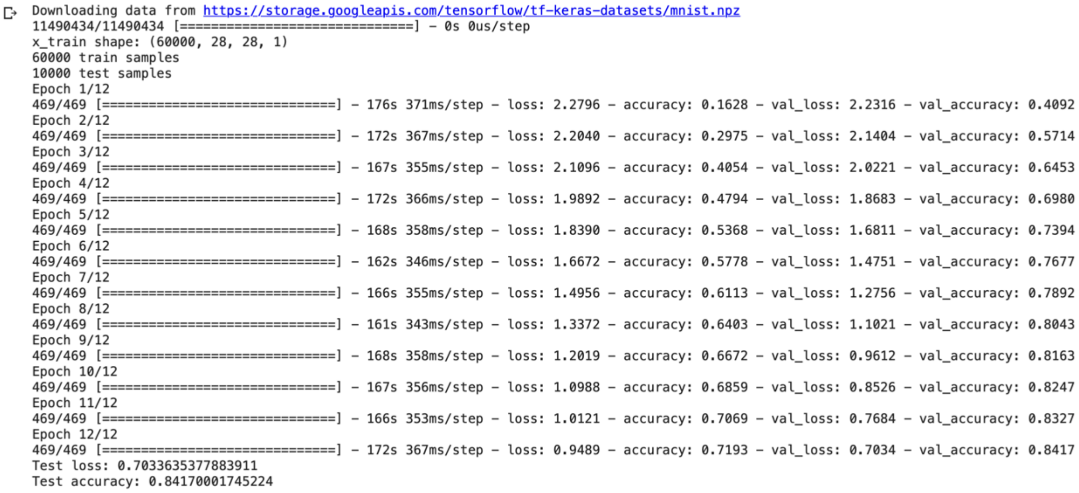


```
Epoch 100/100
469/469 [==============================] - 97s 206ms/step - loss: 0.3417 - accuracy: 0.8967 - val_loss: 0.2142 - val_accuracy: 0.9380
Test loss: 0.21424643695354462
Test accuracy: 0.9380000233650208
```

Figure 2 performance of epoch equals to 100

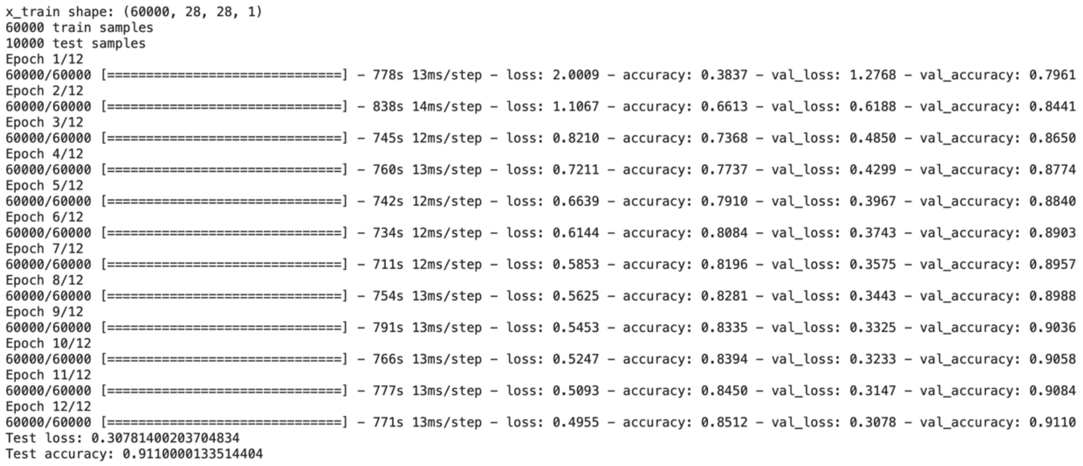
2.2 Effect of batch size on performance

Batch size is the number of images used to train a single forward and backward pass in a character recognition program. Setting batch size too high can make the network take too long time to train; however, if it is too low, it will make the network bounce back and forth without achieving acceptable performance. Figure 3 and figure 4 show the result, the number of batch sizes respectively equals 128 and 1.



```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Epoch 1/12
469/469 [==============================] - 176s 371ms/step - loss: 2.2796 - accuracy: 0.1628 - val_loss: 2.2316 - val_accuracy: 0.4092
Epoch 2/12
469/469 [==============================] - 172s 367ms/step - loss: 2.2040 - accuracy: 0.2975 - val_loss: 2.1404 - val_accuracy: 0.5714
Epoch 3/12
469/469 [==============================] - 167s 355ms/step - loss: 2.1096 - accuracy: 0.4054 - val_loss: 2.0221 - val_accuracy: 0.6453
Epoch 4/12
469/469 [==============================] - 172s 366ms/step - loss: 1.9892 - accuracy: 0.4794 - val_loss: 1.8683 - val_accuracy: 0.6980
Epoch 5/12
469/469 [==============================] - 168s 358ms/step - loss: 1.8390 - accuracy: 0.5368 - val_loss: 1.6811 - val_accuracy: 0.7394
Epoch 6/12
469/469 [==============================] - 162s 346ms/step - loss: 1.6672 - accuracy: 0.5778 - val_loss: 1.4751 - val_accuracy: 0.7677
Epoch 7/12
469/469 [==============================] - 166s 355ms/step - loss: 1.4956 - accuracy: 0.6113 - val_loss: 1.2756 - val_accuracy: 0.7892
Epoch 8/12
469/469 [==============================] - 161s 343ms/step - loss: 1.3372 - accuracy: 0.6403 - val_loss: 1.1021 - val_accuracy: 0.8043
Epoch 9/12
469/469 [==============================] - 168s 358ms/step - loss: 1.2019 - accuracy: 0.6672 - val_loss: 0.9612 - val_accuracy: 0.8163
Epoch 10/12
469/469 [==============================] - 167s 356ms/step - loss: 1.0988 - accuracy: 0.6859 - val_loss: 0.8526 - val_accuracy: 0.8247
Epoch 11/12
469/469 [==============================] - 166s 353ms/step - loss: 1.0121 - accuracy: 0.7069 - val_loss: 0.7684 - val_accuracy: 0.8327
Epoch 12/12
469/469 [==============================] - 172s 367ms/step - loss: 0.9489 - accuracy: 0.7193 - val_loss: 0.7034 - val_accuracy: 0.8417
Test loss: 0.7033635377883911
Test accuracy: 0.84170001745224
```

Figure 3 performance of batch size equals to 128

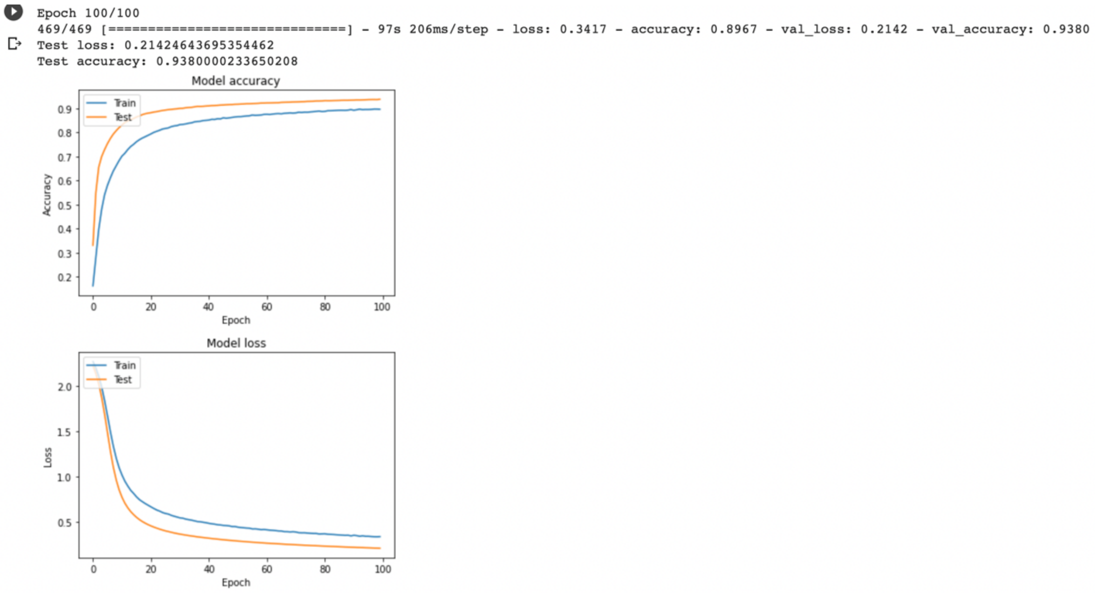


```
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Epoch 1/12
60000/60000 [==============================] - 778s 13ms/step - loss: 2.0009 - accuracy: 0.3837 - val_loss: 1.2768 - val_accuracy: 0.7961
Epoch 2/12
60000/60000 [==============================] - 838s 14ms/step - loss: 1.1067 - accuracy: 0.6613 - val_loss: 0.6188 - val_accuracy: 0.8441
Epoch 3/12
60000/60000 [==============================] - 745s 12ms/step - loss: 0.8210 - accuracy: 0.7368 - val_loss: 0.4850 - val_accuracy: 0.8650
Epoch 4/12
60000/60000 [==============================] - 760s 13ms/step - loss: 0.7211 - accuracy: 0.7737 - val_loss: 0.4299 - val_accuracy: 0.8774
Epoch 5/12
60000/60000 [==============================] - 742s 12ms/step - loss: 0.6639 - accuracy: 0.7910 - val_loss: 0.3967 - val_accuracy: 0.8840
Epoch 6/12
60000/60000 [==============================] - 734s 12ms/step - loss: 0.6144 - accuracy: 0.8084 - val_loss: 0.3743 - val_accuracy: 0.8903
Epoch 7/12
60000/60000 [==============================] - 711s 12ms/step - loss: 0.5853 - accuracy: 0.8196 - val_loss: 0.3575 - val_accuracy: 0.8957
Epoch 8/12
60000/60000 [==============================] - 754s 13ms/step - loss: 0.5625 - accuracy: 0.8281 - val_loss: 0.3443 - val_accuracy: 0.8988
Epoch 9/12
60000/60000 [==============================] - 791s 13ms/step - loss: 0.5453 - accuracy: 0.8335 - val_loss: 0.3325 - val_accuracy: 0.9036
Epoch 10/12
60000/60000 [==============================] - 766s 13ms/step - loss: 0.5247 - accuracy: 0.8394 - val_loss: 0.3233 - val_accuracy: 0.9058
Epoch 11/12
60000/60000 [==============================] - 777s 13ms/step - loss: 0.5093 - accuracy: 0.8450 - val_loss: 0.3147 - val_accuracy: 0.9084
Epoch 12/12
60000/60000 [==============================] - 771s 13ms/step - loss: 0.4955 - accuracy: 0.8512 - val_loss: 0.3078 - val_accuracy: 0.9110
Test loss: 0.30781400203704834
Test accuracy: 0.9110000133514404
```

Figure 4 performance of batch size equals to 1

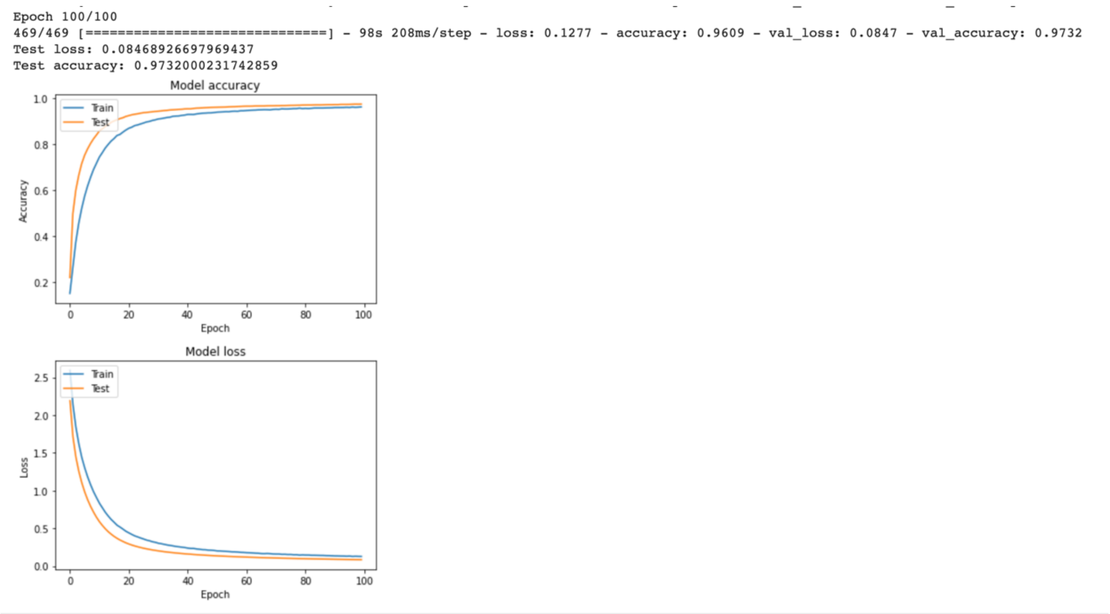
2.3 Effect of the number of layers on performance

By adding more layers and increasing the number of filters, a deeper and denser network allows the model to learn more complex features of the features. The results are shown respectively in figure 5 and figure 6.



```
Epoch 100/100
469/469 [==============================] - 97s 206ms/step - loss: 0.3417 - accuracy: 0.8967 - val_loss: 0.2142 - val_accuracy: 0.9380
Test loss: 0.21424643695354462
Test accuracy: 0.9380000233650208
```

Figure 5 original model

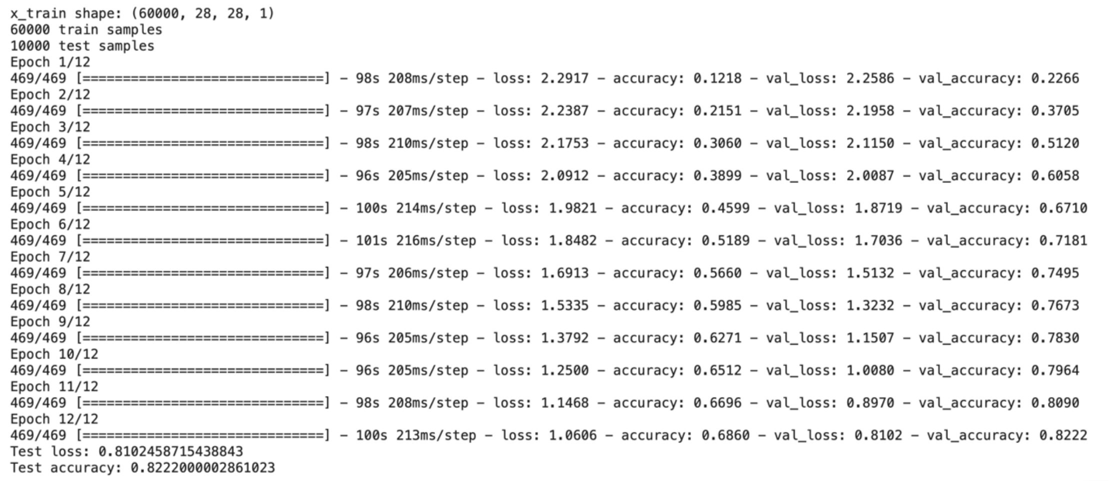


```
Epoch 100/100
469/469 [==============================] - 98s 208ms/step - loss: 0.1277 - accuracy: 0.9609 - val_loss: 0.0847 - val_accuracy: 0.9732
Test loss: 0.08468926697969437
Test accuracy: 0.9732000231742859
```

Figure 6 modified model

3 Results

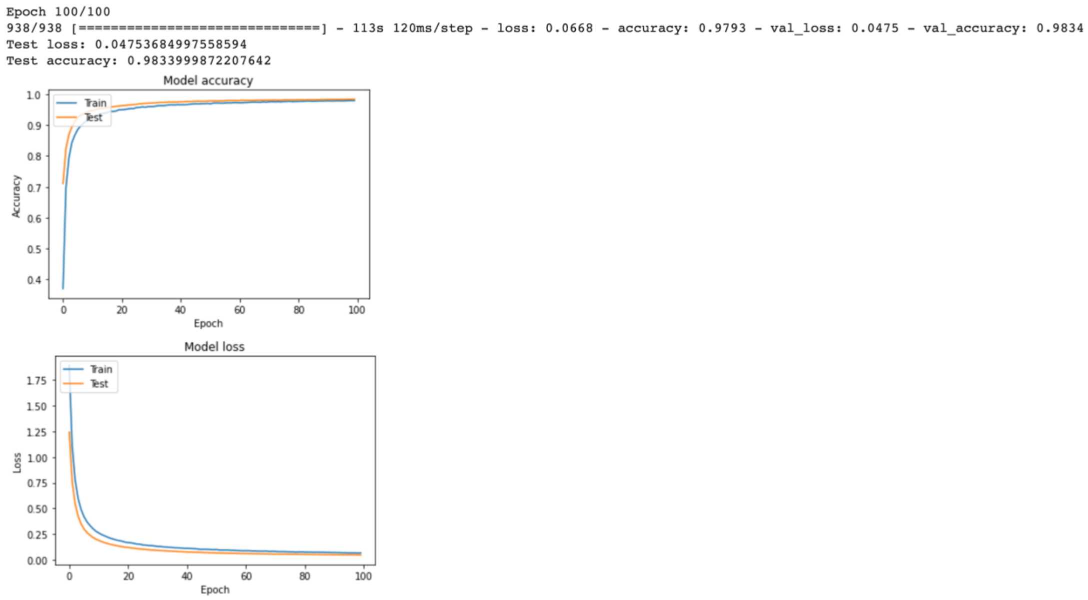
The result of the original CNN model is shown in figure 7.



```
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Epoch 1/12
469/469 [==============================] - 98s 208ms/step - loss: 2.2917 - accuracy: 0.1218 - val_loss: 2.2586 - val_accuracy: 0.2266
Epoch 2/12
469/469 [==============================] - 97s 207ms/step - loss: 2.2387 - accuracy: 0.2151 - val_loss: 2.1958 - val_accuracy: 0.3705
Epoch 3/12
469/469 [==============================] - 98s 210ms/step - loss: 2.1753 - accuracy: 0.3060 - val_loss: 2.1150 - val_accuracy: 0.5120
Epoch 4/12
469/469 [==============================] - 96s 205ms/step - loss: 2.0912 - accuracy: 0.3899 - val_loss: 2.0087 - val_accuracy: 0.6058
Epoch 5/12
469/469 [==============================] - 100s 214ms/step - loss: 1.9821 - accuracy: 0.4599 - val_loss: 1.8719 - val_accuracy: 0.6710
Epoch 6/12
469/469 [==============================] - 101s 216ms/step - loss: 1.8482 - accuracy: 0.5189 - val_loss: 1.7036 - val_accuracy: 0.7181
Epoch 7/12
469/469 [==============================] - 97s 206ms/step - loss: 1.6913 - accuracy: 0.5660 - val_loss: 1.5132 - val_accuracy: 0.7495
Epoch 8/12
469/469 [==============================] - 98s 210ms/step - loss: 1.5335 - accuracy: 0.5985 - val_loss: 1.3232 - val_accuracy: 0.7673
Epoch 9/12
469/469 [==============================] - 96s 205ms/step - loss: 1.3792 - accuracy: 0.6271 - val_loss: 1.1507 - val_accuracy: 0.7830
Epoch 10/12
469/469 [==============================] - 96s 205ms/step - loss: 1.2500 - accuracy: 0.6512 - val_loss: 1.0080 - val_accuracy: 0.7964
Epoch 11/12
469/469 [==============================] - 98s 208ms/step - loss: 1.1468 - accuracy: 0.6696 - val_loss: 0.8970 - val_accuracy: 0.8090
Epoch 12/12
469/469 [==============================] - 100s 213ms/step - loss: 1.0606 - accuracy: 0.6860 - val_loss: 0.8102 - val_accuracy: 0.8222
Test loss: 0.8102458715438843
Test accuracy: 0.8222000002861023
```

Figure 7 result of the original model

The modified CNN model was trained for 100 epochs and gives the following result shown in figure 8.



```
Epoch 100/100
938/938 [==============================] - 113s 120ms/step - loss: 0.0668 - accuracy: 0.9793 - val_loss: 0.0475 - val_accuracy: 0.9834
Test loss: 0.04753684997558594
Test accuracy: 0.9833999872207642
```

Figure 8 result of the modified model

By adding more layers, increasing epochs and other methods, the recognition rate was increased from 82.22% to 98.34%.

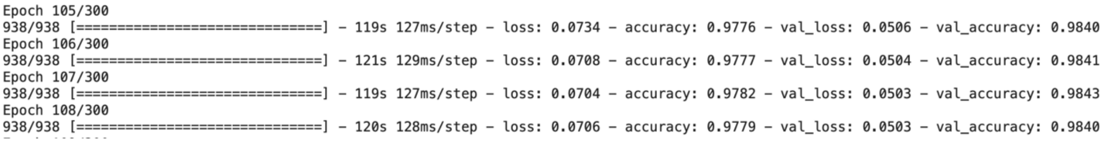
4 Discussion

There are 60000 training samples and 10000 test samples in the dataset. By modifying the original model, the accuracy of the recognition system raised from 82.22% to 98.34%.

The modified CNN model has 3 convolution layers, 3 max-pooling layers, and 2 dense layers. All the convolution layers and the first dense layer use the ReLU function as their activation function. The last dense layer uses the SoftMax function as its activation function to normalize the output. The kernel size of all the convolution layers is 3*3. Therefore, the first convolution layer has 3*3*32=288 weights and 32 biases, the second convolution layer has 3*3*48=432 layers and 48 biases, and the third convolution layer has 3*3*64 weights and 64 biases. Max-pooling layers have no bias. For fully connected layers, each element of the layer connects to all the elements in the next layer. Therefore, the first dense layer has 576*576=311766 weights and 576 biases, the second dense layer has 576*10=5760 weights and 10 biases.

In this modified CNN model, a dropout layer with a hyperparameter of 25% was added after the last max-pooling layer. By randomly dropping 25% of the neurons while training, the network generalizes better by simply not relying too much on any particular neurons to produce an output.

After more epochs, the accuracy rate and loss rate are drawn below. Through this figure 9, It can be predicted that after 107 epochs, the model starts to overfit the training set.



```
Epoch 105/300
938/938 [==============================] - 119s 127ms/step - loss: 0.0734 - accuracy: 0.9776 - val_loss: 0.0506 - val_accuracy: 0.9840
Epoch 106/300
938/938 [==============================] - 121s 129ms/step - loss: 0.0708 - accuracy: 0.9777 - val_loss: 0.0504 - val_accuracy: 0.9841
Epoch 107/300
938/938 [==============================] - 119s 127ms/step - loss: 0.0704 - accuracy: 0.9782 - val_loss: 0.0503 - val_accuracy: 0.9843
Epoch 108/300
938/938 [==============================] - 120s 128ms/step - loss: 0.0706 - accuracy: 0.9779 - val_loss: 0.0503 - val_accuracy: 0.9840
```

Figure 9 Overfitting

5 Conclusions

The accuracy of the character recognition system can be raised by changing the factors below.

- Increase the epochs — An epoch means training the neural network with all the training data for one cycle. By increasing the number of epochs, the model can be learned more times to increase its accuracy.
- Add more layers — The original model comprised two convolutional layers. In the new network, more convolutional layers were added.
- Deeper network — The number of filters in the convolutional layers and max-pooling layers was increased. The number of hidden in the dense layers was also increased to accommodate the larger input due to the increase in the volume of the convolutional layers. A deeper and denser network allows the model to learn more complex features of the features.