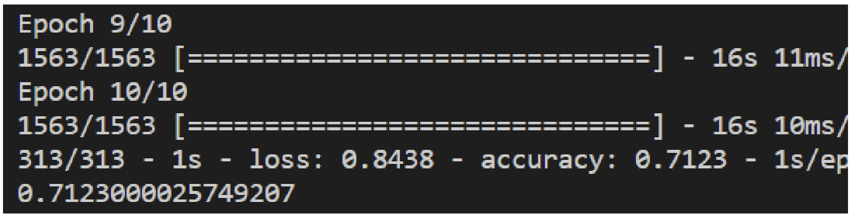
Deep Learning in Biomedical Image Analysis

Homework#3 學號：109033209 姓名：陳俊丞

1. Please execute this code and report the final accuracy of test data.



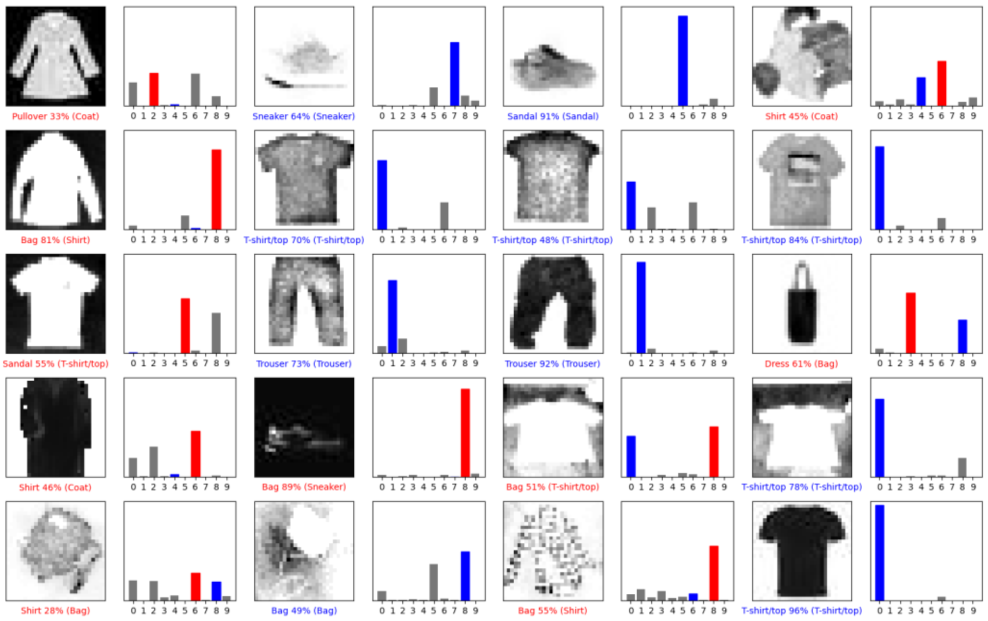
```
Epoch 9/10
1563/1563 [==============================] - 16s 11ms/
Epoch 10/10
1563/1563 [==============================] - 16s 10ms/
313/313 - 1s - loss: 0.8438 - accuracy: 0.7123 - 1s/ep
0.7123000025749207
```

The accuracy is about 71.23%

2. Compare the accuracy of clothes identification using CNN versus FCNN. Use your own fashion data and compare the accuracy of classification results of these two neural network models.

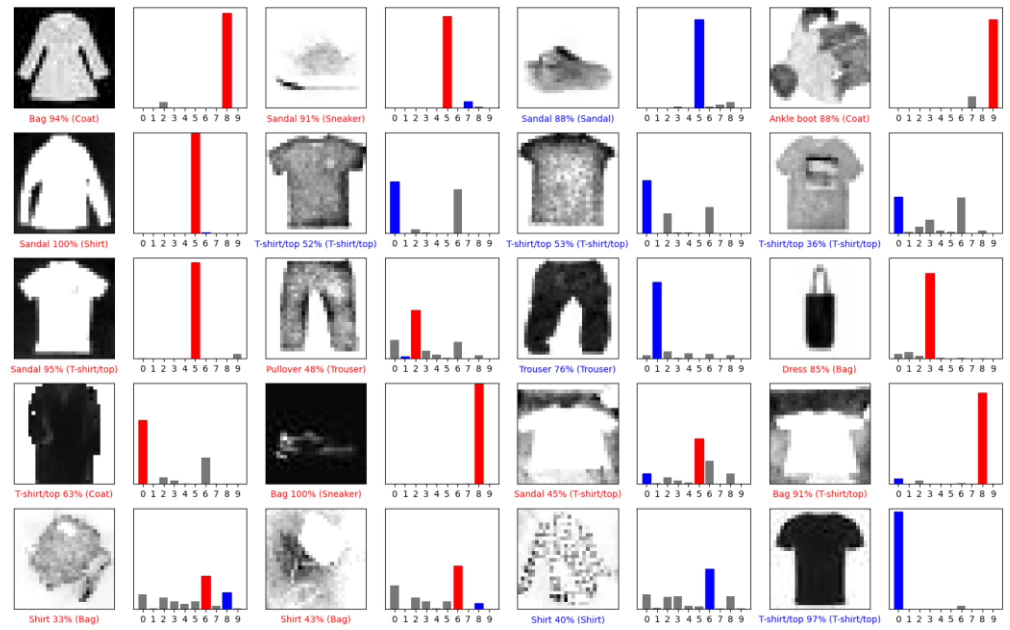
Clothes with CNN function

Accuracy = 50%(10/20)

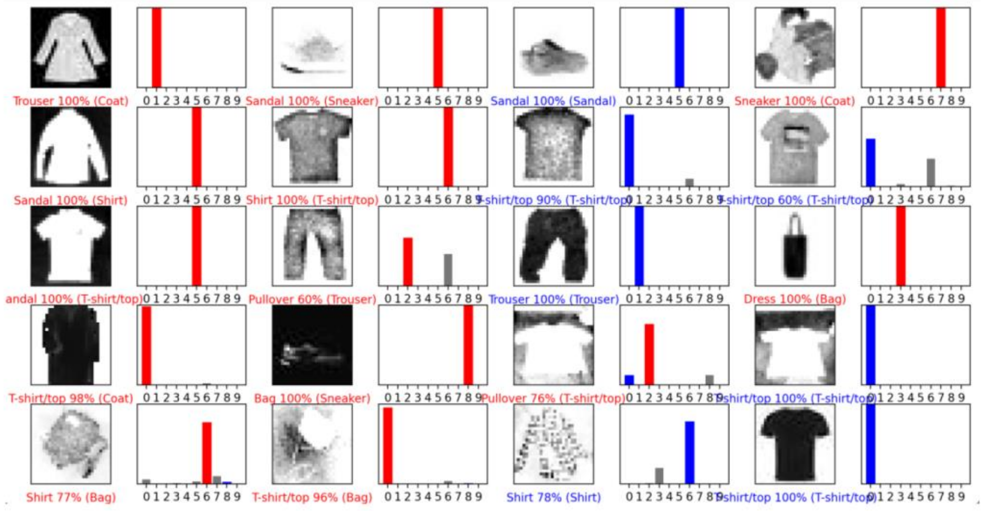


Clothes with FCNN function

Accuracy = 35%(7/20)



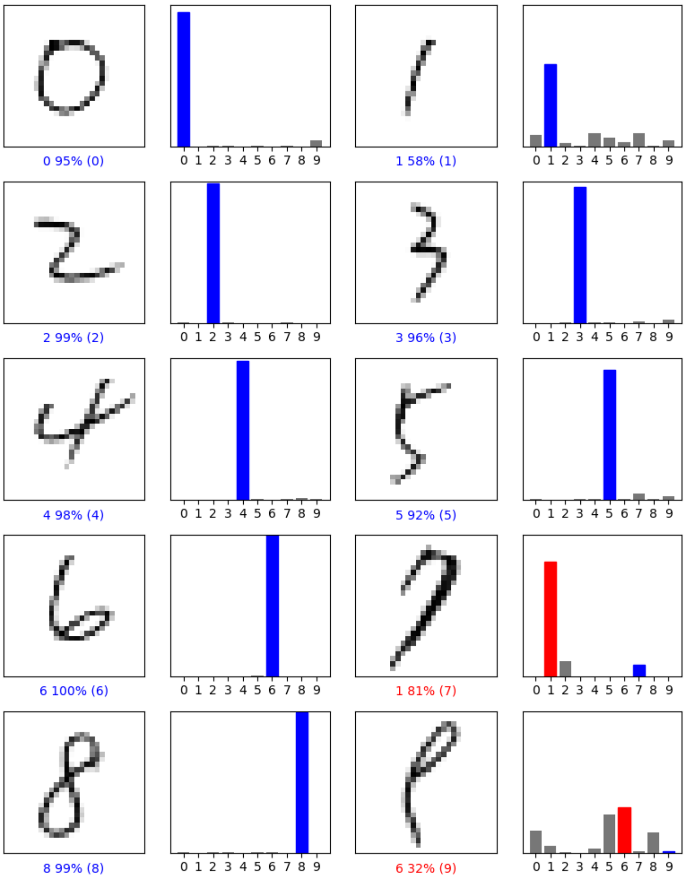
上次作業的學習模型(HW2)

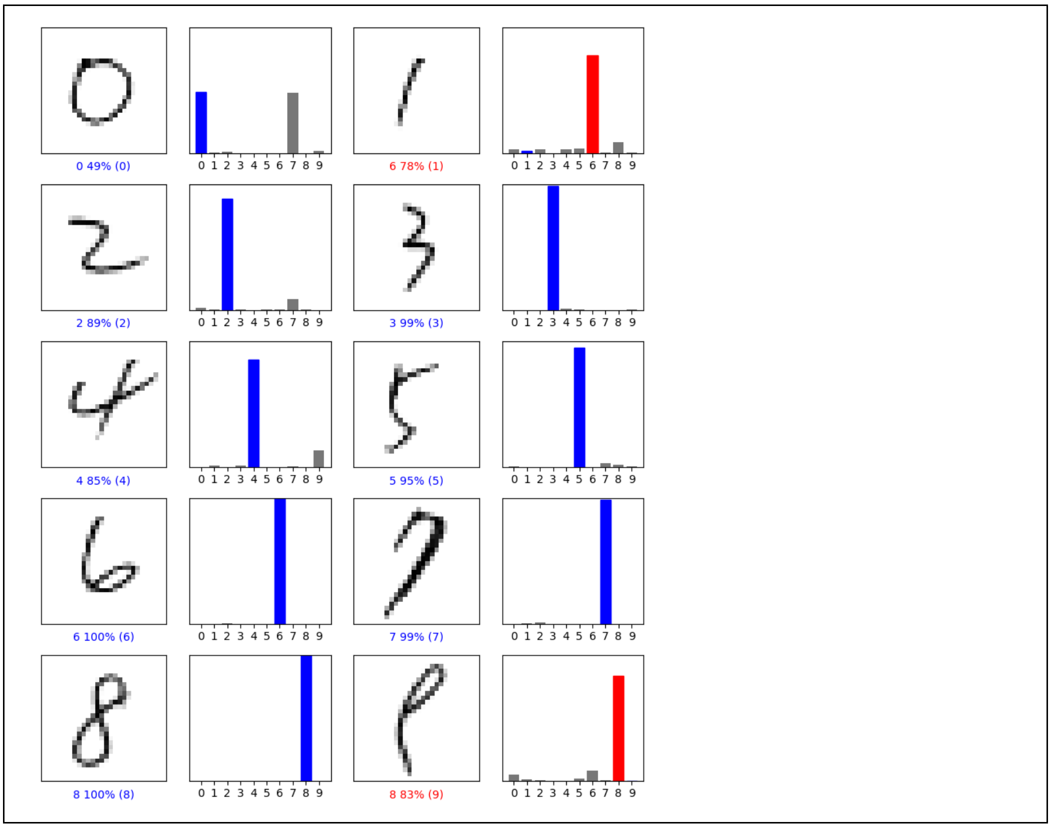


相比於上次的學習模型，這次透過 CNN 與 FCNN 學習模型作測試在準確率方面有很好的提升，當然也包括將較為不明顯的圖案辨識正確，即便答錯的測試中也有提高正確答案機率的情況發生，可見 CNN 以及 FCNN 對深度學習有很大的幫助，而我也發現 CNN 的相對弱勢是在於圖案不能夠過大或過小，否則在辨識上會出現明顯的錯誤，即使是相同的圖案作放大縮小都會影響識別。

不過關於照片也有機會是因為衣服皺摺的程度不一，導致在拍攝的成果尚無法完整的呈現，另外也有可能是因為衣物的顏色與背景的顏色太過相近，以至於在調整畫素時會發生衣物與背景的交界不夠明顯，所以在學習上會有判斷的誤差，最後就是衣物的選擇上，雖然在建立模型是使用了 60,000 筆測資，但衣物的種類、樣式、紋路，甚至是擺放方向可能還是無法完整掌握，所以在識別上會有學習的落差。

3. Compare the accuracy of digits identification using CNN versus FCNN. Use your own fashion data and compare the accuracy of classification results of these two neural network models.

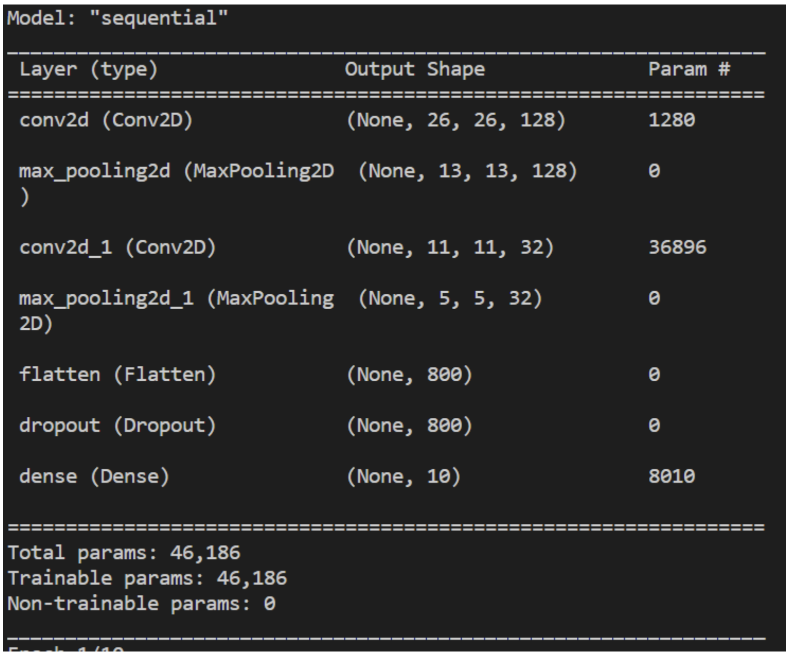
Digits with CNN function Accuracy = 80%(8/10)

Digits with FCNN function Accuracy = 80%(8/10)



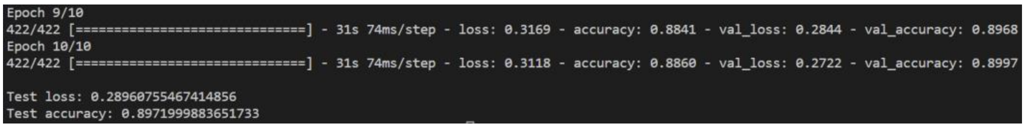
除了衣物，這次的作業也包括手寫數字辨別，儘管數字的形狀相對簡單，但有可能隨著寫手獨特的手寫習慣而與原先大眾所認識的數字形狀有落差，學習模型就會變得難以識別，例如我所寫入的數字9，人眼能夠識別此數字為9，儘管形狀像5、圓弧的形狀像6，對機器辨識就會加深難度，除次之外，我發現CNN將7認成1，不過FCNN並沒有認錯，反而將1認成6，所以雖然CNN以及FCNN都能夠提高圖案辨識，但還是在個別的識別下有誤差。

4. Additional information on model and HDF5 file usage

Clothes CNN model structure(Accuracy = 89.72%)

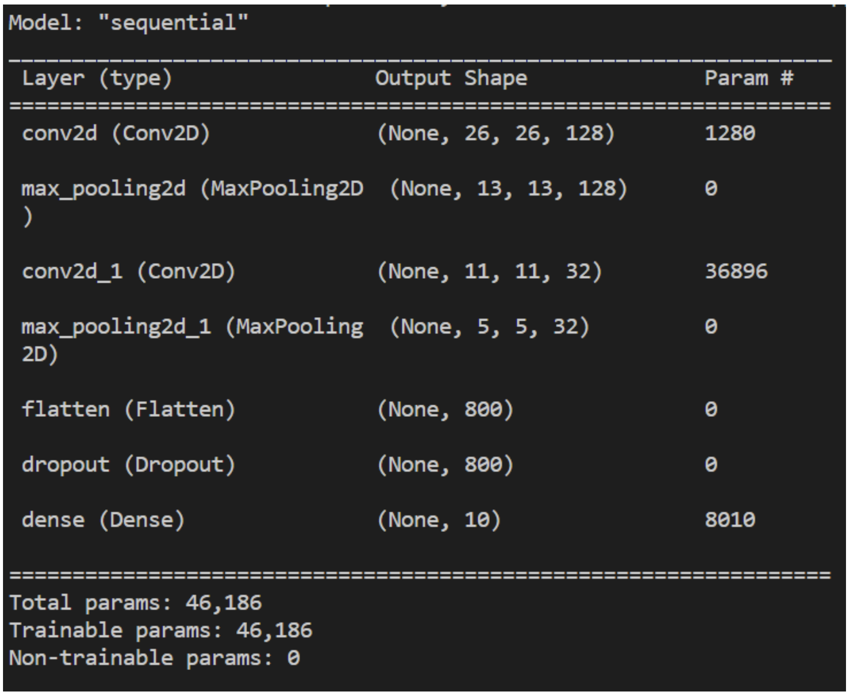


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 128)       1280

 max_pooling2d (MaxPooling2D  (None, 13, 13, 128)      0
 )

 conv2d_1 (Conv2D)           (None, 11, 11, 32)        36896

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 32)         0
 2D)

 flatten (Flatten)           (None, 800)               0

 dropout (Dropout)           (None, 800)               0

 dense (Dense)               (None, 10)                8010

=================================================================
Total params: 46,186
Trainable params: 46,186
Non-trainable params: 0
_________________________________________________________________
```

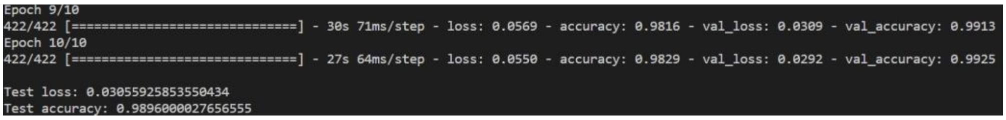


```
Epoch 9/10
422/422 [==============================] - 31s 74ms/step - loss: 0.3169 - accuracy: 0.8841 - val_loss: 0.2844 - val_accuracy: 0.8968
Epoch 10/10
422/422 [==============================] - 31s 74ms/step - loss: 0.3118 - accuracy: 0.8860 - val_loss: 0.2722 - val_accuracy: 0.8997

Test loss: 0.28960755467414856
Test accuracy: 0.8971999883651733
```

Digits CNN model structure(Accuracy = 98.96%)

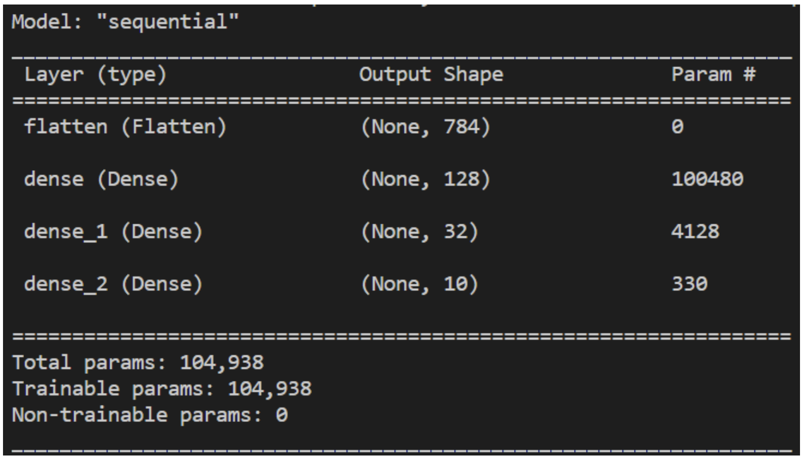


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 128)       1280

 max_pooling2d (MaxPooling2D  (None, 13, 13, 128)      0
 )

 conv2d_1 (Conv2D)           (None, 11, 11, 32)        36896

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 32)         0
 2D)

 flatten (Flatten)           (None, 800)               0

 dropout (Dropout)           (None, 800)               0

 dense (Dense)               (None, 10)                8010

=================================================================
Total params: 46,186
Trainable params: 46,186
Non-trainable params: 0
```

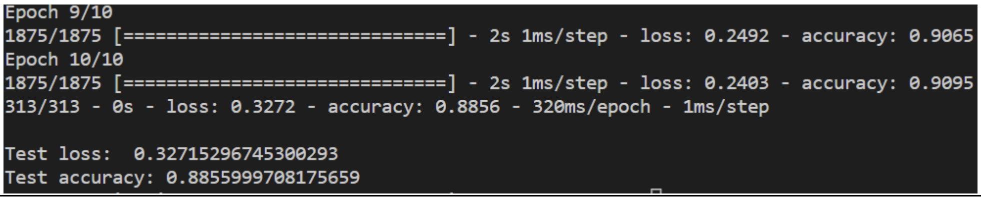


```
Epoch 9/10
422/422 [==============================] - 30s 71ms/step - loss: 0.0569 - accuracy: 0.9816 - val_loss: 0.0309 - val_accuracy: 0.9913
Epoch 10/10
422/422 [==============================] - 27s 64ms/step - loss: 0.0550 - accuracy: 0.9829 - val_loss: 0.0292 - val_accuracy: 0.9925

Test loss: 0.03055925853550434
Test accuracy: 0.9896000027656555
```

Clothes FCNN model structure(Accuracy = 88.56%)

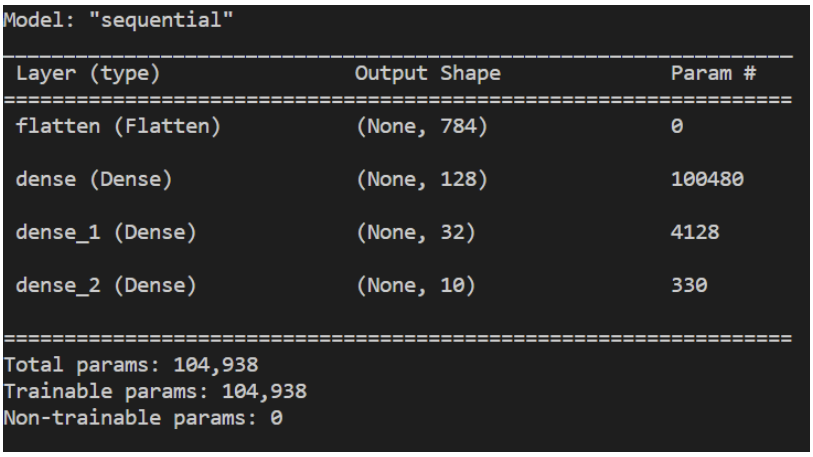


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 32)                4128

 dense_2 (Dense)             (None, 10)                330

=================================================================
Total params: 104,938
Trainable params: 104,938
Non-trainable params: 0
_________________________________________________________________
```

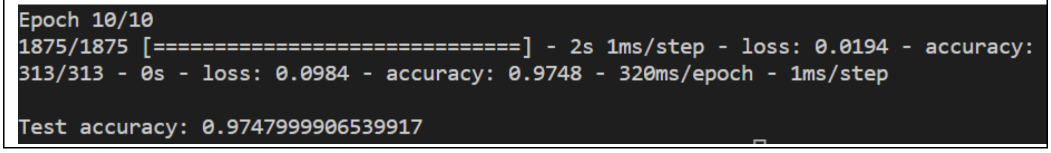


```
Epoch 9/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2492 - accuracy: 0.9065
Epoch 10/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2403 - accuracy: 0.9095
313/313 - 0s - loss: 0.3272 - accuracy: 0.8856 - 320ms/epoch - 1ms/step

Test loss:  0.32715296745300293
Test accuracy: 0.8855999708175659
```

Digits FCNN model structure(Accuracy = 97.48%)



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 32)                4128

 dense_2 (Dense)             (None, 10)                330

=================================================================
Total params: 104,938
Trainable params: 104,938
Non-trainable params: 0
```



```
Epoch 10/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.0194 - accuracy:
313/313 - 0s - loss: 0.0984 - accuracy: 0.9748 - 320ms/epoch - 1ms/step

Test accuracy: 0.9747999906539917
```

上面的圖片分別是我 FCNN 以及 CNN 使用的模型架構，並且都套用在 mnist_digit 數據集以及 mnist_fashion 數據集，並且將各自的模型以及訓練過程的參數透過 model.save 來儲存，分別為 digit_fcnn.h5, digit_cnn.h5, cloth_fcnn.h5, cloth_cnn.h5。之後運用 keras.models.load_model 來將訓練好的模型輸入，並且運用之前處理好的照片來驗證。