Using “normal” and “retrain” mode to improve accuracy of the MNIST model/

1. Write down the very first accuracy and the epoch you trained.

```
Train on 48000 samples, validate on 12000 samples
48000/48000 [==============================] - 1s 12us/sample - loss: 0.0907 - accuracy: 0.0900 - val_loss: 0.0905 - val_accura
cy: 0.1000
10000/10000 [==============================] - 1s 52us/sample - loss: 0.0905 - accuracy: 0.0903
60000/60000 [==============================] - 2s 36us/sample - loss: 0.0905 - accuracy: 0.0973
        [Info] Accuracy of training data = 9.7%
        [Info] Accuracy of testing data = 9.0%
```

↑預設使用 epoch=1、mode=’normal’得出的結果，正確率只有 9%，先試跑 30 次。

```
Epoch 30/30
48000/48000 [==============================] - 0s 8us/sample - loss: 0.0778 - accuracy: 0.5541 - val_loss: 0.0773 - val_accur
acy: 0.5630
10000/10000 [==============================] - 1s 56us/sample - loss: 0.0773 - accuracy: 0.5683
60000/60000 [==============================] - 2s 40us/sample - loss: 0.0775 - accuracy: 0.5595
        [Info] Accuracy of training data = 55.9%
        [Info] Accuracy of testing data = 56.8%
```

↑epochs=30，正確率來到 56.8%。

2. Write down the last accuracy you retrained and how many times did you retrain your model.

```
Epoch 30/30
48000/48000 [==============================] - 0s 7us/sample - loss: 0.0579 - accuracy: 0.6904 - val_loss: 0.0569 - val_accur
acy: 0.7068
10000/10000 [==============================] - 1s 51us/sample - loss: 0.0568 - accuracy: 0.7021
60000/60000 [==============================] - 2s 37us/sample - loss: 0.0575 - accuracy: 0.6953
        [Info] Accuracy of training data = 69.5%
        [Info] Accuracy of testing data = 70.2%
```

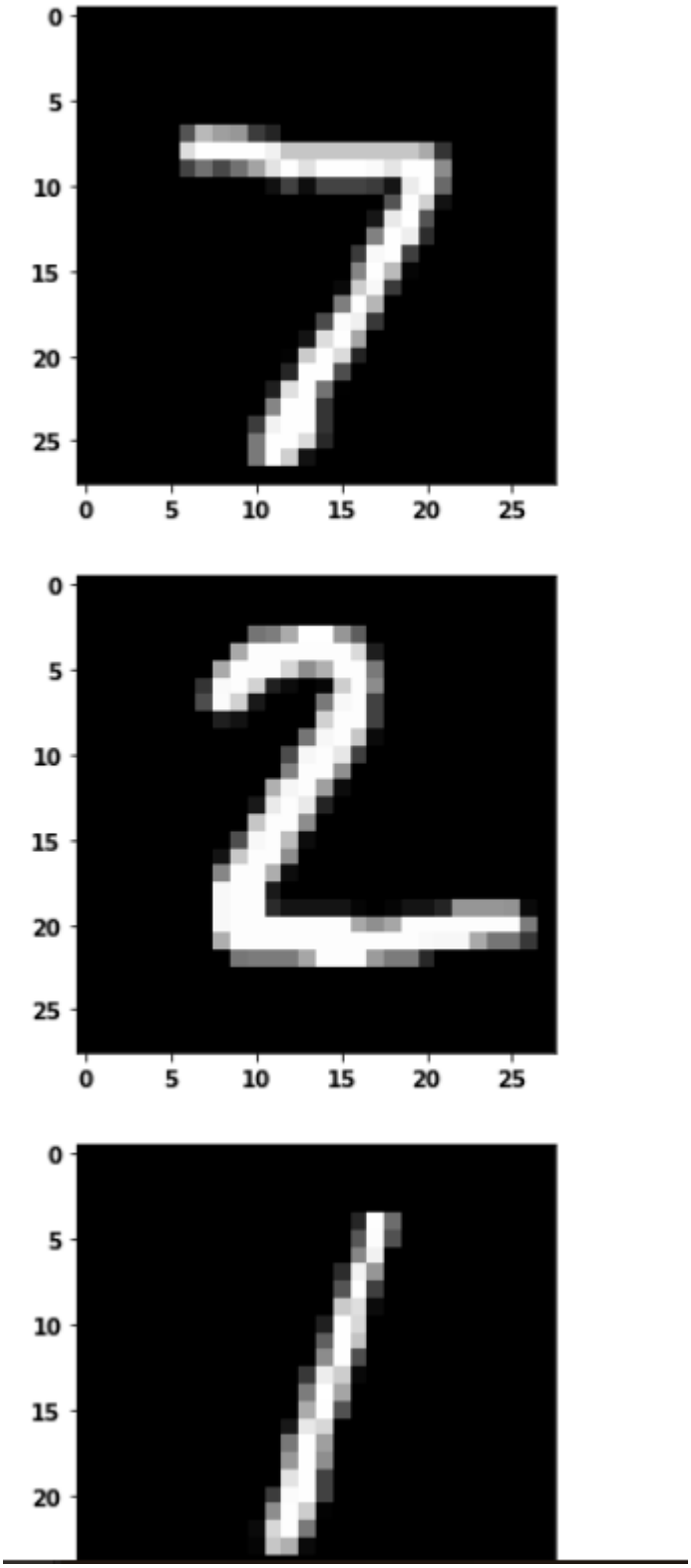
↑再使用 mode=’retrain’訓練 30 epochs，正確率來到 70.2%

```
Epoch 30/30
48000/48000 [==============================] - 0s 7us/sample - loss: 0.0417 - accuracy: 0.7935 - val_loss: 0.0402 - val_accurac
y: 0.8087
10000/10000 [==============================] - 1s 52us/sample - loss: 0.0403 - accuracy: 0.8080
60000/60000 [==============================] - 2s 38us/sample - loss: 0.0413 - accuracy: 0.7977
        [Info] Accuracy of training data = 79.8%
        [Info] Accuracy of testing data = 80.8%
```

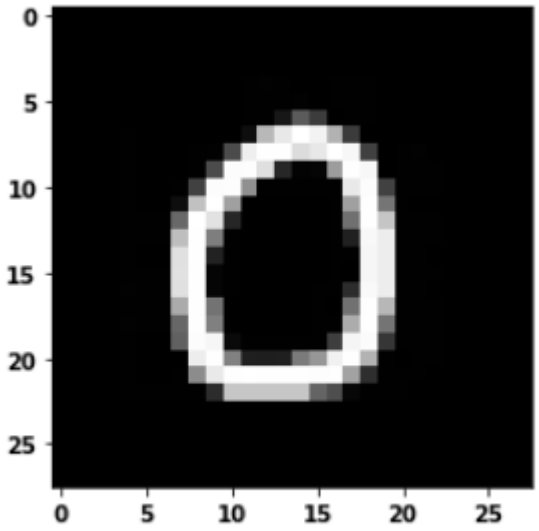
↑繼續用 mode=’retrain’訓練 30 epochs，總共 retrain 60 次正確率來到 80.8%

3. Write down the prediction performance of your MNIST model.

```
[7 2 1 0 4 1 7 4 6 7]
```

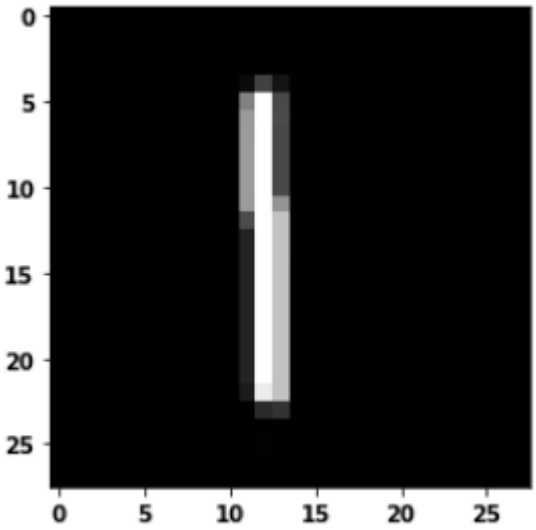


↑ 70.2%正確率可以讓前三張都對



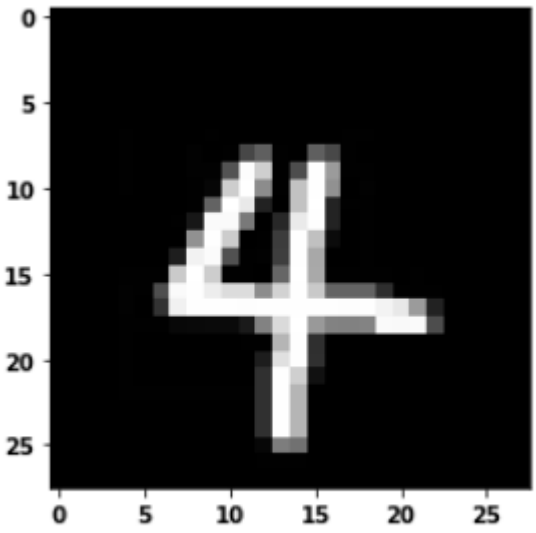

```
result:0
```

↑使用 converted_model.tflite 對助教圖片 0.jpg 做預測



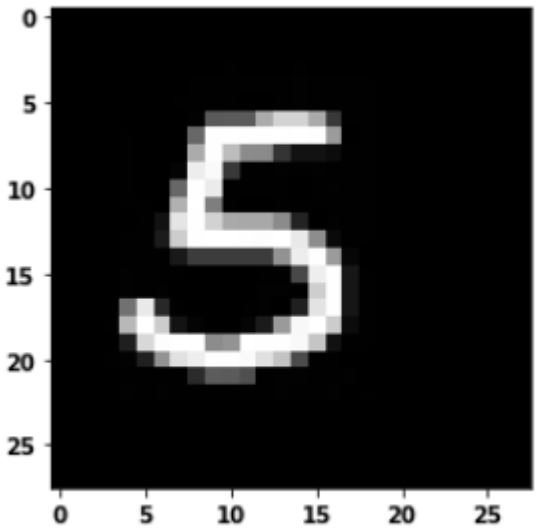

```
result:8
```

↑使用 converted_model.tflite 對助教圖片 1.jpg 做預測



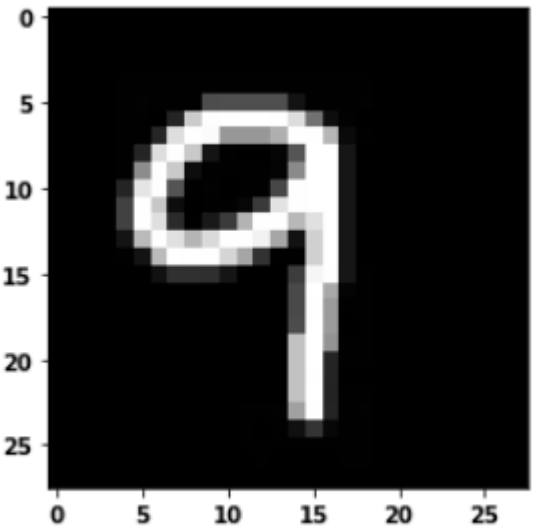

```
result:6
```

↑使用 converted_model.tflite 對助教圖片 4.jpg 做預測




```
result:8
```

↑使用 converted_model.tflite 對助教圖片 5.jpg 做預測




```
result:3
```

↑使用 converted_model.tflite 對助教圖片 9.jpg 做預測
10 張圖片有 4 張是錯誤的，正確率 60%。

Determine the performance of PoseNET model and write down the possible reason why the performance is good or bad.

完全不行，使用 camera 偵測同學是否跌倒永遠只有一個答案:FALL，比較可能的原因是 threshold 設定太高，兩點距離永遠<50 以至於只有 FALL 的答案，若將 threshold 修改成 10 的話情況會稍微改善，原作者的閥值設得太高，也可能是因為攝影機解析度不同，因此雙方讀到的 pixel 值也不同，可能我們的 10pixel=原作者的 50pixel。