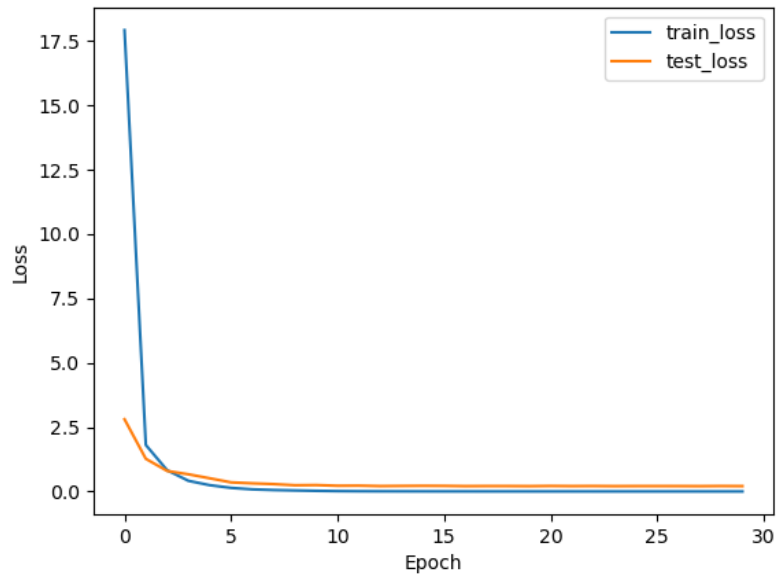
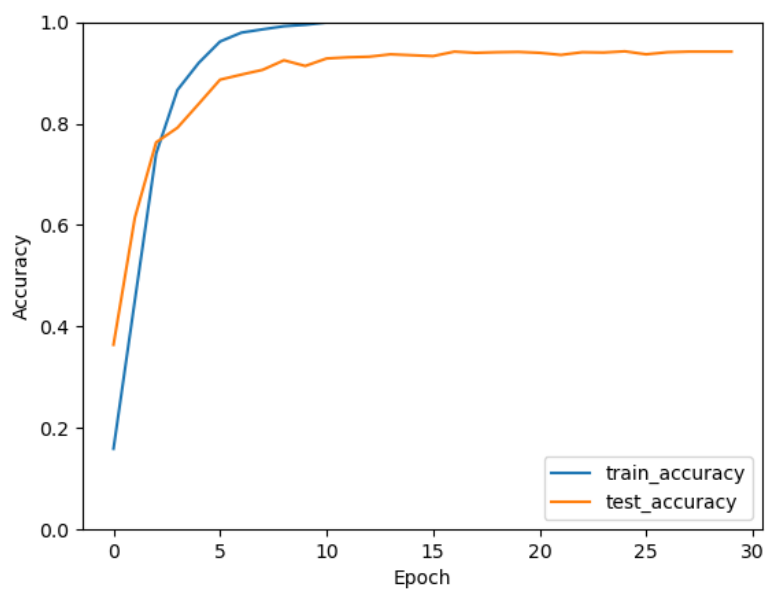


## 一、中文手寫辨識準確率及損失率

Test loss ,Test accuracy 截圖：



(圖一) loss，橘線為 test loss



(圖二) accuracy，橘線為 test accuracy

```
10/10 [=====] - 4s 87ms/step - loss: 17.9285 - accuracy: 0.1592 - val_loss: 2.8058 - val_accuracy: 0.3641
Epoch 2/30
10/10 [=====] - 1s 54ms/step - loss: 1.8075 - accuracy: 0.4510 - val_loss: 1.2698 - val_accuracy: 0.6141
Epoch 3/30
10/10 [=====] - 1s 62ms/step - loss: 0.8253 - accuracy: 0.7408 - val_loss: 0.7974 - val_accuracy: 0.7629
Epoch 4/30
10/10 [=====] - 1s 55ms/step - loss: 0.4132 - accuracy: 0.8661 - val_loss: 0.6674 - val_accuracy: 0.7918
Epoch 5/30
10/10 [=====] - 1s 54ms/step - loss: 0.2447 - accuracy: 0.9196 - val_loss: 0.5120 - val_accuracy: 0.8388
Epoch 6/30
10/10 [=====] - 1s 58ms/step - loss: 0.1388 - accuracy: 0.9616 - val_loss: 0.3530 - val_accuracy: 0.8865
Epoch 7/30
10/10 [=====] - 1s 54ms/step - loss: 0.0816 - accuracy: 0.9792 - val_loss: 0.3158 - val_accuracy: 0.8965
Epoch 8/30
10/10 [=====] - 1s 54ms/step - loss: 0.0551 - accuracy: 0.9857 - val_loss: 0.2865 - val_accuracy: 0.9059
Epoch 9/30
10/10 [=====] - 1s 59ms/step - loss: 0.0385 - accuracy: 0.9918 - val_loss: 0.2424 - val_accuracy: 0.9247
Epoch 10/30
10/10 [=====] - 1s 58ms/step - loss: 0.0237 - accuracy: 0.9947 - val_loss: 0.2485 - val_accuracy: 0.9135
Epoch 11/30
10/10 [=====] - 1s 55ms/step - loss: 0.0131 - accuracy: 0.9992 - val_loss: 0.2218 - val_accuracy: 0.9282
Epoch 12/30
10/10 [=====] - 1s 59ms/step - loss: 0.0083 - accuracy: 0.9996 - val_loss: 0.2253 - val_accuracy: 0.9306
Epoch 13/30
10/10 [=====] - 1s 52ms/step - loss: 0.0057 - accuracy: 1.0000 - val_loss: 0.2092 - val_accuracy: 0.9318
Epoch 14/30
10/10 [=====] - 1s 60ms/step - loss: 0.0047 - accuracy: 1.0000 - val_loss: 0.2149 - val_accuracy: 0.9365
Epoch 15/30
10/10 [=====] - 1s 54ms/step - loss: 0.0035 - accuracy: 1.0000 - val_loss: 0.2190 - val_accuracy: 0.9347
Epoch 16/30
10/10 [=====] - 1s 55ms/step - loss: 0.0028 - accuracy: 1.0000 - val_loss: 0.2163 - val_accuracy: 0.9329
Epoch 17/30
10/10 [=====] - 1s 55ms/step - loss: 0.0023 - accuracy: 1.0000 - val_loss: 0.2071 - val_accuracy: 0.9418
Epoch 18/30
10/10 [=====] - 1s 56ms/step - loss: 0.0020 - accuracy: 1.0000 - val_loss: 0.2102 - val_accuracy: 0.9394
Epoch 19/30
10/10 [=====] - 1s 56ms/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 0.2098 - val_accuracy: 0.9406
Epoch 20/30
10/10 [=====] - 1s 55ms/step - loss: 0.0015 - accuracy: 1.0000 - val_loss: 0.2063 - val_accuracy: 0.9412
Epoch 21/30
10/10 [=====] - 1s 55ms/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.2135 - val_accuracy: 0.9394
Epoch 22/30
10/10 [=====] - 1s 53ms/step - loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.2081 - val_accuracy: 0.9353
Epoch 23/30
10/10 [=====] - 1s 54ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.2113 - val_accuracy: 0.9406
Epoch 24/30
10/10 [=====] - 1s 54ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.2065 - val_accuracy: 0.9400
Epoch 25/30
10/10 [=====] - 1s 61ms/step - loss: 9.4379e-04 - accuracy: 1.0000 - val_loss: 0.2089 - val_accuracy: 0.9424
Epoch 26/30
10/10 [=====] - 1s 58ms/step - loss: 8.6618e-04 - accuracy: 1.0000 - val_loss: 0.2091 - val_accuracy: 0.9365
Epoch 27/30
10/10 [=====] - 1s 55ms/step - loss: 7.9926e-04 - accuracy: 1.0000 - val_loss: 0.2087 - val_accuracy: 0.9406
Epoch 28/30
10/10 [=====] - 1s 56ms/step - loss: 7.4524e-04 - accuracy: 1.0000 - val_loss: 0.2062 - val_accuracy: 0.9418
Epoch 29/30
10/10 [=====] - 1s 55ms/step - loss: 6.9573e-04 - accuracy: 1.0000 - val_loss: 0.2109 - val_accuracy: 0.9418
Epoch 30/30
10/10 [=====] - 1s 54ms/step - loss: 6.3781e-04 - accuracy: 1.0000 - val_loss: 0.2077 - val_accuracy: 0.9418
```

(圖三) 詳細訓練紀錄，此處 loss 為 train\_loss，accuracy 為 train\_accuracy，val\_loss 為 test\_loss，val\_accuracy 為 test\_accuracy

## 二、Source code 之逐行解釋

### 1\_prepareData.py 解釋：

此程式目的為方便訓練時讀檔，將訓練資料檔案路徑重新規劃

```
1  # import之後要用到的Librarys
2  import shutil
3  import os
4  from PIL import Image
5
6  # 為了方便訓練時讀檔，我將train_image和test_image資料路徑處理成：
7  # -----
8  # /train_image
9  #   /0
10 #     /1.bmp
11 #     /2.bmp
12 #     /...
13 #   /1
14 #     /6.bmp
15 #     /7.bmp
16 #     /...
17 #   /2
18 #   /...
19 # -----
20 # 就是把資料拉出來，去掉學號那層folder
21
22 # 準備資料路徑
23 train_path = './train_image'
24 test_path = './test_image'
25 # 準備第二層folder中的學號名稱
26 train_studentID = os.listdir(train_path)
27 test_studentID = os.listdir(test_path)
```

2：import shutil，之後用來刪除空的資料夾

3：import OS，之後用來建立資料夾、讀資料名稱

4：import Image，之後用來儲存影像

23：train\_path 為 train 資料路徑變數名稱，./train\_image 為資料路徑

24：test\_path 為 test 資料路徑變數名稱，./test\_image 為資料路徑

26：train\_studentID 為 list，其內容為 train\_path 中所有檔案的檔名

27：test\_studentID 為 list，其內容為 test\_path 中所有檔案的檔名

```

28
29 # 建立資料夾
30 ~for i in range(10):
31     os.mkdir(train_path + '/' + str(i))
32     os.mkdir(test_path + '/' + str(i))
33

```

30：因為有十種標籤，建立十個名稱為 1~10 的資料夾，以便放入相對應片

31：在 train\_path 路徑中，製作名稱為 i 的資料夾，i 為 1~10

32：在 test\_path 路徑中，製作名稱為 i 的資料夾，i 為 1~10

```

34 # 將train_image中的資料去除學號那層folder
35 count = 1
36 for s in train_studentID:
37     for i in os.listdir(train_path + '/' + s):
38         for j in os.listdir(train_path + '/' + s + '/' + i):
39             # file_path為原始圖檔的完整路徑
40             file_path = train_path + '/' + s + '/' + i + '/' + j
41             # 因為要同類圖片放入同資料夾，所以將抓出的圖片重新命名為連續數字
42             save_path = train_path + f'/{i}/{count}.png'
43             # 將.bmp轉為.png方便訓練
44             Image.open(file_path).resize((28,28)).save(save_path)
45             count += 1
46
47 # 將test_image中的資料去除學號那層folder
48 count = 1
49 for s in test_studentID:
50     for i in os.listdir(test_path + '/' + s):
51         for j in os.listdir(test_path + '/' + s + '/' + i):
52             # file_path為原始圖檔的完整路徑
53             file_path = test_path + '/' + s + '/' + i + '/' + j
54             # 因為要同類圖片放入同資料夾，所以將抓出的圖片重新命名為連續數字
55             save_path = test_path + f'/{i}/{count}.png'
56             # 將.bmp轉為.png方便訓練
57             Image.open(file_path).resize((28,28)).save(save_path)
58             count += 1
59
60 # 刪除空的學號folder
61 for i in train_studentID:
62     shutil.rmtree(train_path + '/' + i, ignore_errors=True)
63 for i in test_studentID:
64     shutil.rmtree(test_path + '/' + i, ignore_errors=True)

```

35：count 為 int 變數，目的在記住目前加入了幾張圖片

36：s 為 train 資料夾中，學號資料夾的學號

37：i 為 s 學號資料夾內的所有標籤名稱，包括 1~10

38：j 為 s 學號資料夾內的所有圖片名稱

40：file\_path 為指定圖片的完整路徑

42：save\_path 為要儲存圖片的路徑

44：讀取 file\_path 的圖片，存在 save\_path  
45：count 遞增 1，圖片命名不重複  
48：count 為 int 變數，目的在記住目前加入了幾張圖片  
49：s 為 test 資料夾中，學號資料夾的學號  
50：i 為 s 學號資料夾內的所有標籤名稱，包括 1~10  
51：j 為 s 學號資料夾內的所有圖片名稱  
53：file\_path 為指定圖片的完整路徑  
55：save\_path 為要儲存圖片的路徑  
57：讀取 file\_path 的圖片，存在 save\_path  
58：count 遞增 1，圖片命名不重複  
61：i 為 train 資料夾中的學號資料夾名稱  
62：移除 i 資料夾  
63：i 為 test 資料夾中的學號資料夾名稱  
64：移除 i 資料夾

## 2\_train.py 解釋：

訓練模型並輸出 accuracy 和 loss

```
1 # import之後要用到的libraries
2 import tensorflow as tf
3 import matplotlib.pyplot as plt
4 from tensorflow.keras import datasets, layers, models
5
6 # batch size 參數
7 batch_size = 256
8 # 影像shape
9 img_height = 28
10 img_width = 28
11
12 # 資料路徑(由1_prepareData.py處理完成的資料)
13 train_dir = './train_image'
14 test_dir = './test_image'
15
16 # 準備training dataset
17 train_ds = tf.keras.utils.image_dataset_from_directory(
18     train_dir, # train data路徑
19     color_mode='grayscale', # 灰階
20     seed=666, # 亂數seed
21     image_size=(img_height, img_width),
22     batch_size=batch_size)
23
24 # 準備testing dataset
25 test_ds = tf.keras.utils.image_dataset_from_directory(
26     test_dir, # test data路徑
27     color_mode='grayscale', # 灰階
28     seed=666, # 亂數seed
29     image_size=(img_height, img_width),
30     batch_size=batch_size)
31
```

2：import tensorflow，訓練模型需要用到

3：import matplotlib.pyplot，輸出圖片需要用到

4：import 更多 tensorflow 套件

7：設定訓練時的 batch size

9：設定影像高度

10：設定影像寬度

13：train data 路徑

14：test data 路徑

17：建立 train dataset，資料夾名稱為標籤時使用 image\_dataset\_from\_directory

18：參數，train data 路徑

- 19：參數，灰階圖
- 20：參數，種子碼
- 21：參數，影像大小
- 22：參數，batch size
- 25：建立 test dataset，資料夾名稱為標籤時使用 image\_dataset\_from\_directory
- 26：參數，test data 路徑
- 27：參數，灰階圖
- 28：參數，種子碼
- 29：參數，影像大小
- 30：參數，batch size

```
32 # 檢查class names
33 class_names = train_ds.class_names
34 print(class_names)
35
36 # 檢查data shape
37 for image_batch, labels_batch in train_ds:
38     print(image_batch.shape)
39     print(labels_batch.shape)
40     break
```

- 33：class\_names 為 train data 資料夾中的所有資料夾名稱，即 1~10
- 34：印出 class\_names
- 37：從 train\_ds 中取出 image 和 label，train\_ds 資料型態為 tuple
- 38：印出 image\_batch 的 shape
- 39：印出 labels\_batch 的 shape
- 40：印出一次後直接跳出迴圈

```
42 # 建模型(以下激活函數都為relu)
43 model = models.Sequential()
44 # 捲積層3x3
45 model.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(28, 28, 1)))
46 # 池化層2x2
47 model.add(layers.MaxPooling2D((2, 2)))
48 # 捲積層3x3
49 model.add(layers.Conv2D(128, (3, 3), activation='relu'))
50 # 池化層2x2
51 model.add(layers.MaxPooling2D((2, 2)))
52 # 捲積層3x3
53 model.add(layers.Conv2D(128, (3, 3), activation='relu'))
54 # 全連接層
55 model.add(layers.Flatten())
56 model.add(layers.Dense(128, activation='relu'))
57 model.add(layers.Dense(10))
58
59 model.summary()
```

- 43：建立模型，使用 Tensorflow 的 Sequential
- 45：加入一層捲積層，3x3，64 個，激活函數為 Rectified Linear Unit，input 形狀為訓練圖大小
- 47：加入一層池化層，2x2
- 49：加入一層捲積層，3x3，128 個，激活函數為 Rectified Linear Unit
- 51：加入一層池化層，2x2
- 53：加入一層捲積層，3x3，128 個，激活函數為 Rectified Linear Unit
- 55：加入一層 flatten 層，把二維矩陣轉成一維矩陣
- 56：加入一層全連接層，128 個節點，激活函數為 Rectified Linear Unit
- 57：加入一層全連接層，10 個節點
- 59：輸出模型狀態(圖四)

```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)              (None, 26, 26, 64)       640
max_pooling2d (MaxPooling2D) (None, 13, 13, 64)       0
conv2d_1 (Conv2D)            (None, 11, 11, 128)      73856
max_pooling2d_1 (MaxPooling2D) (None, 5, 5, 128)       0
conv2d_2 (Conv2D)            (None, 3, 3, 128)       147584
flatten (Flatten)            (None, 1152)             0
dense (Dense)                (None, 128)              147584
dense_1 (Dense)              (None, 10)               1290
-----
Total params: 370,954
Trainable params: 370,954
Non-trainable params: 0
```

(圖四) model.summary()輸出結果



```

60
61 # compile
62 model.compile(optimizer='adam',
63               loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
64               metrics=['accuracy'])
65
66 # 開始訓練(30個epochs)
67 history = model.fit(train_ds, epochs=30, validation_data=test_ds)
68
69 # 繪製accuracy歷史
70 plt.plot(history.history['accuracy'], Label='train_accuracy')
71 plt.plot(history.history['val_accuracy'], Label = 'test_accuracy')
72 plt.xlabel('Epoch')
73 plt.ylabel('Accuracy')
74 plt.ylim([0, 1])
75 plt.legend(loc='lower right')
76 plt.savefig('accuracy.png')
77 plt.close()
78
79 # 繪製loss歷史
80 plt.plot(history.history['loss'], Label='train_loss')
81 plt.plot(history.history['val_loss'], Label = 'test_loss')
82 plt.xlabel('Epoch')
83 plt.ylabel('Loss')
84 plt.legend(loc='upper right')
85 plt.savefig('loss.png')
86 plt.close()

```

- 62：compile 模型，優化函數為 adam
- 63：參數，loss 為 sparse categorical crossentropy
- 64：參數，訓練和測試期間的模型評估標準為 accuracy
- 67：開始訓練，總共訓練 30 個 epochs
- 70：繪製 train 的 accuracy 線(圖二)
- 71：繪製 test 的 accuracy 線(圖二)
- 72：圖二的 x 軸名稱為 Epoch
- 73：圖二的 y 軸名稱為 Accuracy
- 74：圖二的 y 軸數值上限為 1，下限為 0
- 75：圖二的圖示放置在右下角
- 76：圖二的名稱為 accuracy.png
- 77：關閉圖二
- 80：繪製 train 的 loss 線(圖一)
- 81：繪製 test 的 loss 線(圖一)
- 82：圖一的 x 軸名稱為 Epoch
- 83：圖一的 y 軸名稱為 Loss
- 84：圖一的圖示放置在右上角
- 85：圖一的名稱為 loss.png
- 86：關閉圖一