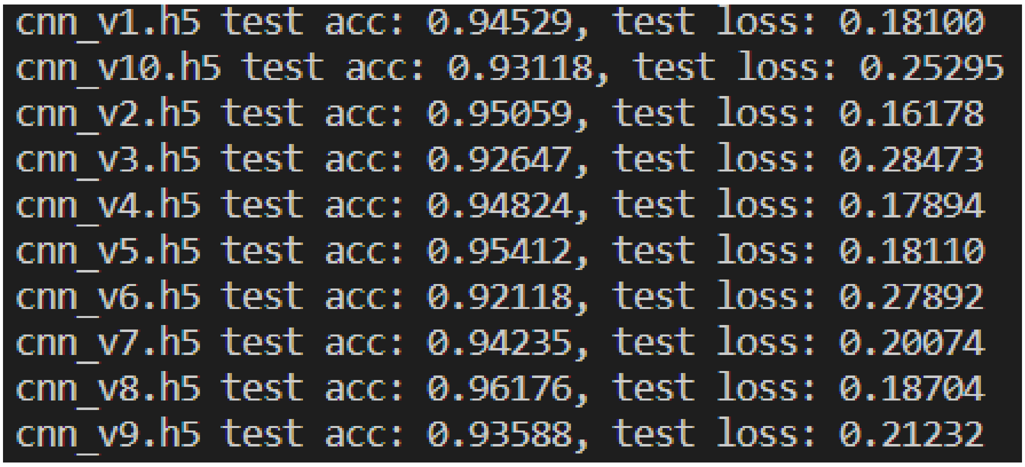
演算法 Term_Project 報告 大氣 4A 106601015 黃展皇

1. 中文手寫辨識準確率(accuracy)，以截圖方式呈現：

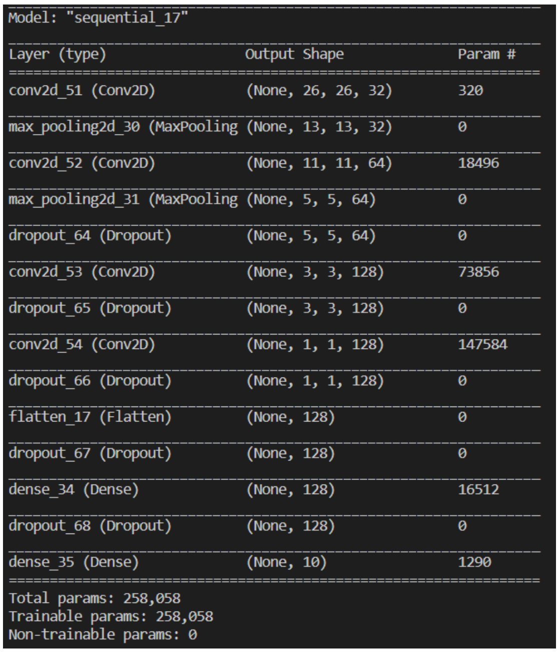


```
cnn_v1.h5 test acc: 0.94529, test loss: 0.18100
cnn_v10.h5 test acc: 0.93118, test loss: 0.25295
cnn_v2.h5 test acc: 0.95059, test loss: 0.16178
cnn_v3.h5 test acc: 0.92647, test loss: 0.28473
cnn_v4.h5 test acc: 0.94824, test loss: 0.17894
cnn_v5.h5 test acc: 0.95412, test loss: 0.18110
cnn_v6.h5 test acc: 0.92118, test loss: 0.27892
cnn_v7.h5 test acc: 0.94235, test loss: 0.20074
cnn_v8.h5 test acc: 0.96176, test loss: 0.18704
cnn_v9.h5 test acc: 0.93588, test loss: 0.21232
```

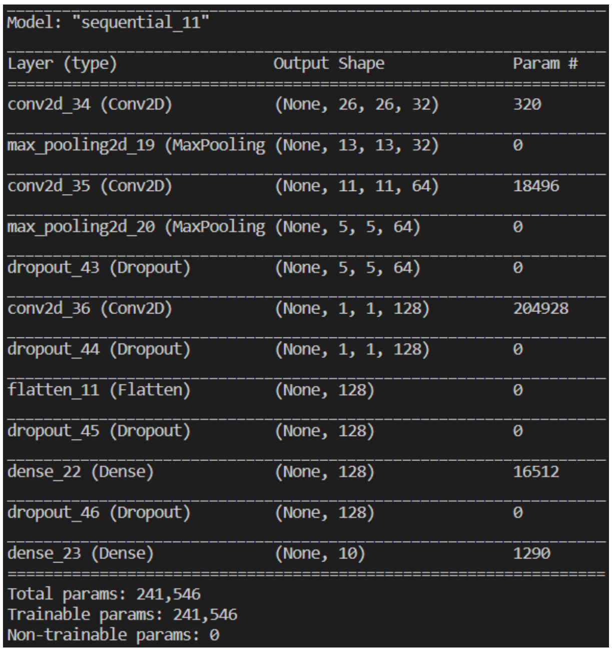
如圖，共做了 10 種版本的 CNN 模型，分別測試深度、kernal 大小及其排列組合、Dropout 大小、有無 Pooling 等等。

其中 acc 表現最好的是 v8(下左圖)，得到 0.96176 的準確度

loss 表現最好的是 v2(下右圖) ，loss 下降到 0.16178



```
_________________________________________________________________
Model: "sequential_17"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_51 (Conv2D)           (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_30 (MaxPooling (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_52 (Conv2D)           (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_31 (MaxPooling (None, 5, 5, 64)          0
_________________________________________________________________
dropout_64 (Dropout)         (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_53 (Conv2D)           (None, 3, 3, 128)         73856
_________________________________________________________________
dropout_65 (Dropout)         (None, 3, 3, 128)         0
_________________________________________________________________
conv2d_54 (Conv2D)           (None, 1, 1, 128)         147584
_________________________________________________________________
dropout_66 (Dropout)         (None, 1, 1, 128)         0
_________________________________________________________________
flatten_17 (Flatten)         (None, 128)               0
_________________________________________________________________
dropout_67 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_34 (Dense)             (None, 128)               16512
_________________________________________________________________
dropout_68 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_35 (Dense)             (None, 10)                1290
=================================================================
Total params: 258,058
Trainable params: 258,058
Non-trainable params: 0
```



```
_________________________________________________________________
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_34 (Conv2D)           (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 5, 5, 64)          0
_________________________________________________________________
dropout_43 (Dropout)         (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 1, 1, 128)         204928
_________________________________________________________________
dropout_44 (Dropout)         (None, 1, 1, 128)         0
_________________________________________________________________
flatten_11 (Flatten)         (None, 128)               0
_________________________________________________________________
dropout_45 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_22 (Dense)             (None, 128)               16512
_________________________________________________________________
dropout_46 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_23 (Dense)             (None, 10)                1290
=================================================================
Total params: 241,546
Trainable params: 241,546
Non-trainable params: 0
```

2. Source code 之逐行解釋

```
if __name__ == '__main__':
此區域執行 main，動作分別為 train 與 test
```

```
Main(operator):
1. 初始化 term_project_path 以及 model_dict
model_dict 調用寫好的 functions 回傳模型，keys 為 cnn models version，values
為各種模型實體

2. if operator == 'train':
建立 train_datapath，並檢查 train_data_x.npy 是否存在，若是則 np.load，若不是
則呼叫 data_x_y_preprocess 讀取 train 圖片並 np.save
對 model_dict 的每個模型做迴圈，若已經訓練過(有.h5 檔)則跳過，若無訓練過則：
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
model.fit(
    train_data_x, train_data_y,
    batch_size=32,
    epochs=30,
    verbose=1,
    validation_split=0.1
)
以上超參數皆保持不變
最後呼叫 show_training_curve 儲存訓練過程繪圖，並且 save_model

3. if operator == 'test':
建立 test_datapath，並檢查 test_data_x.npy 是否存在，若是則 np.load，若不是則
呼叫 data_x_y_preprocess 讀取 test 圖片並 np.save
接著針對 term_project_path 中所有.h5 檔案做 load_model 並
model.evaluate(test_data_x, test_data_y, verbose=0)得到 acc 跟 loss 並印出
```

```
def show_training_curve(train_history):
輸入 model.fit output 的 train_history 並針對 loss、val_loss 隨著 epochs 繪圖
並儲存
```

```
def create_cnn_model_v1(): ~ def create_cnn_model_v10():
Sequential 一個循序模型，依照要測試的模型不同加入不同層，最後 return model
```

```
def data_x_y_preprocess(datapath):
輸入 datapath 路徑(只會是 train_image/test_image)，並且 os.walk 對 datapath
內的所有圖片做讀取、nomalize、reshape，並對 labels 做 to_categorical，最後
return data_x, data_y 分別對應資料以及 labels
```

```
import os
import random
import numpy as np
from PIL import Image
#import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.models import save_model, load_model
from tensorflow.keras.applications import ResNet152V2, ResNet50
from tensorflow.keras.utils import to_categorical
from matplotlib import pyplot as plt
import datetime
```

補充以上程式所需要的套件及版本：

tensorflow (-gpu)=2.4.0、pillow =7.2.0、numpy =1.19.0、matplotlib=3.2.2

電腦配置：~~垃圾~~windows10、conda env、tensorflow-gpu、GeForce GTX 1050 Ti 4GB、DDR4-2666 8G*2