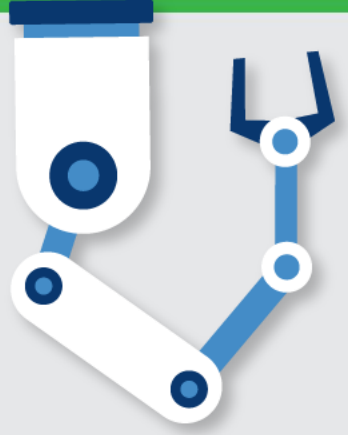
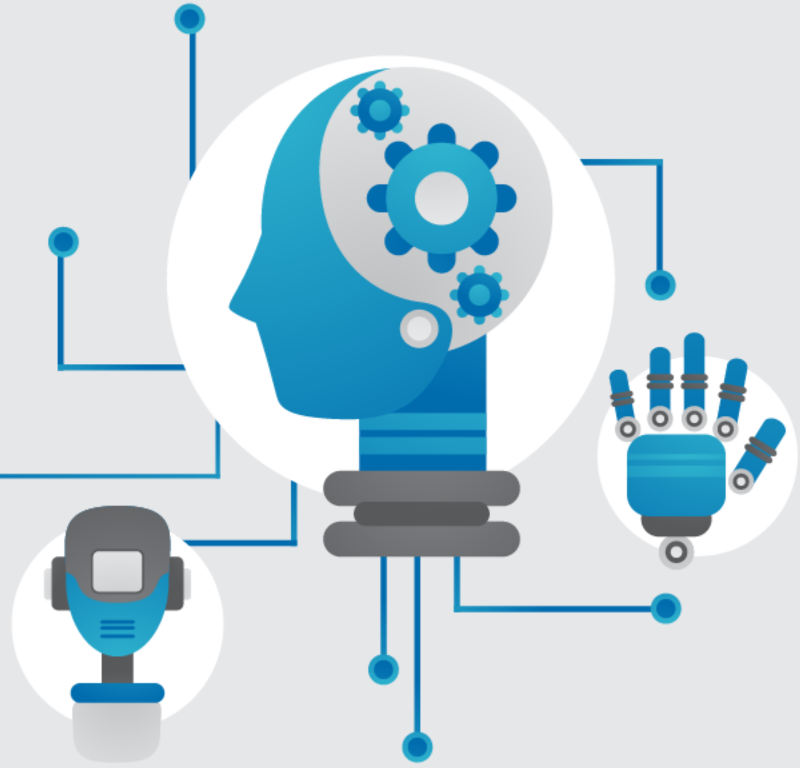
長榮 MOOCs
Chang Jung Christian University

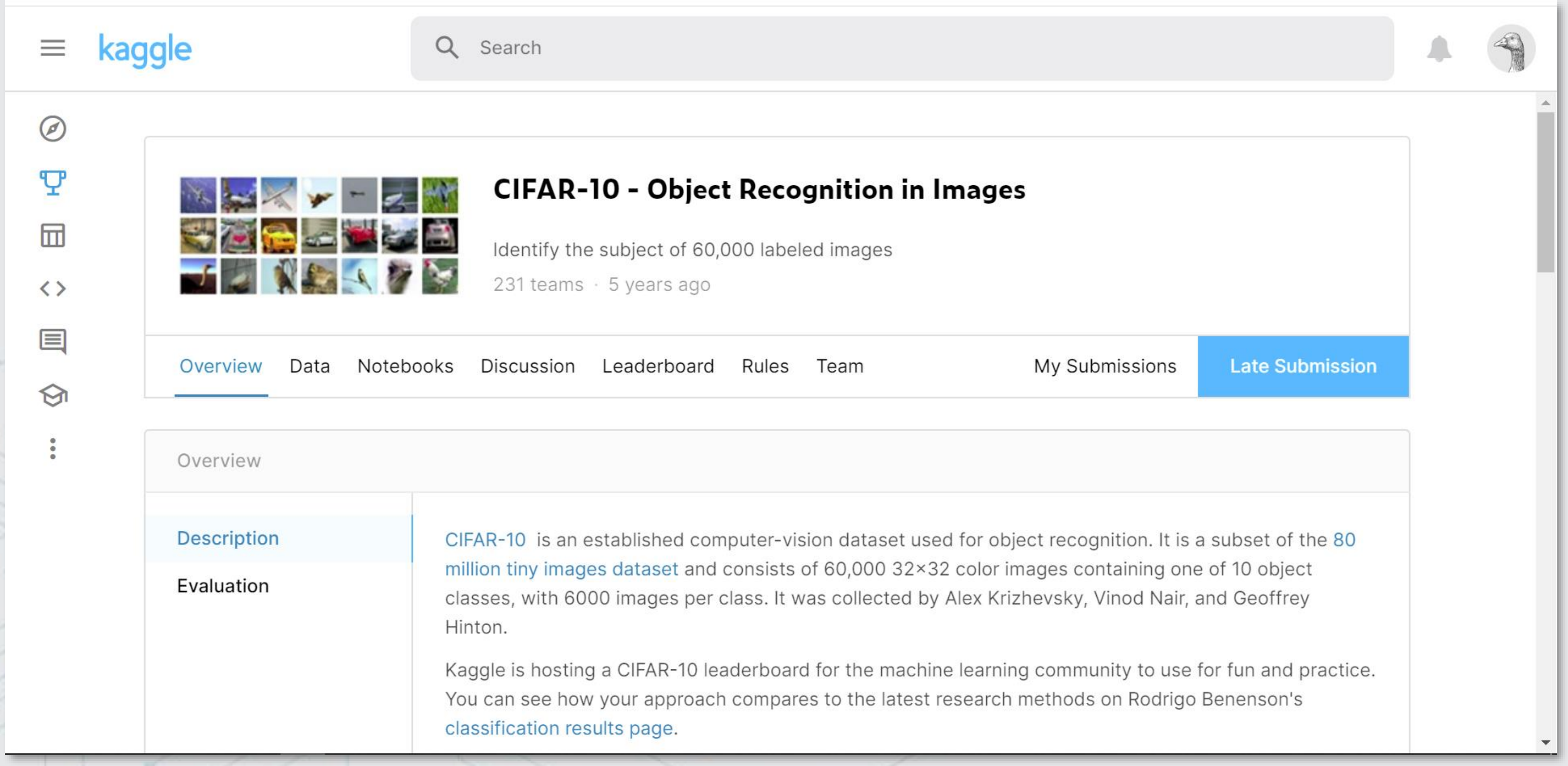


CNN模型建置流程



Kaggle競賽 - Object Recognition in Images

- 網址：https://www.kaggle.com/c/cifar-10/overview



CIFAR-10 - Object Recognition in Images

Identify the subject of 60,000 labeled images

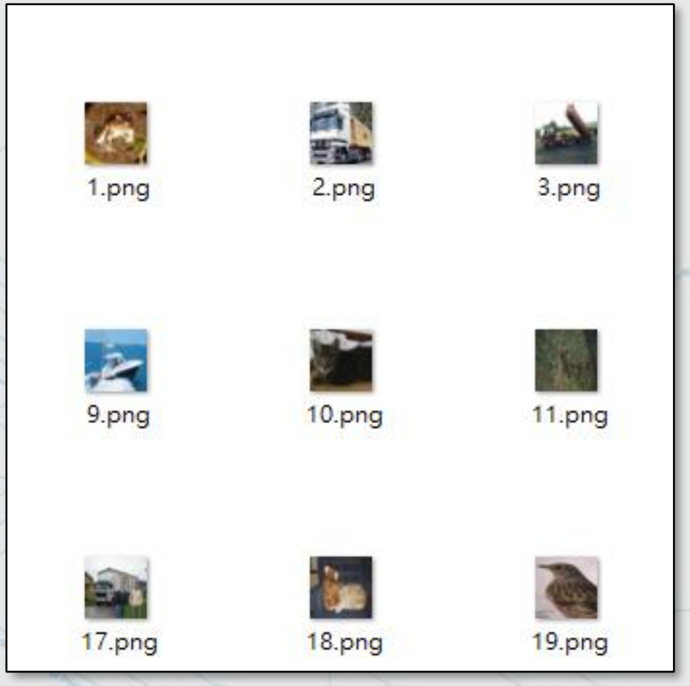
231 teams · 5 years ago

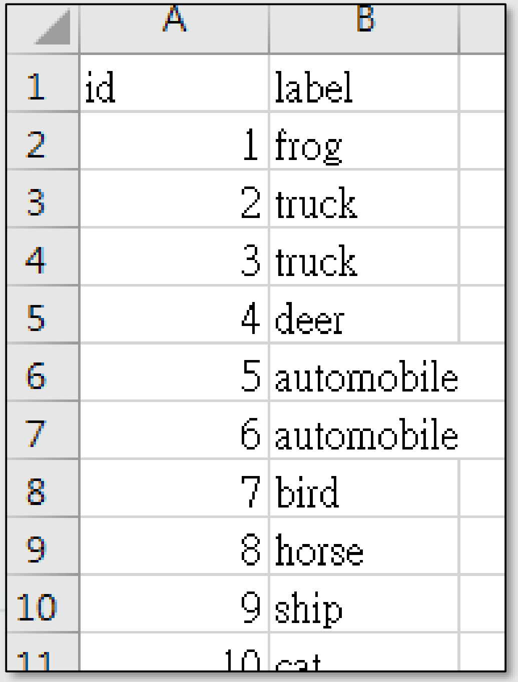
CIFAR-10 is an established computer-vision dataset used for object recognition. It is a subset of the 80 million tiny images dataset and consists of 60,000 32×32 color images containing one of 10 object classes, with 6000 images per class. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

Kaggle is hosting a CIFAR-10 leaderboard for the machine learning community to use for fun and practice. You can see how your approach compares to the latest research methods on Rodrigo Benenson's classification results page.

Kaggle競賽 - Object Recognition in Images

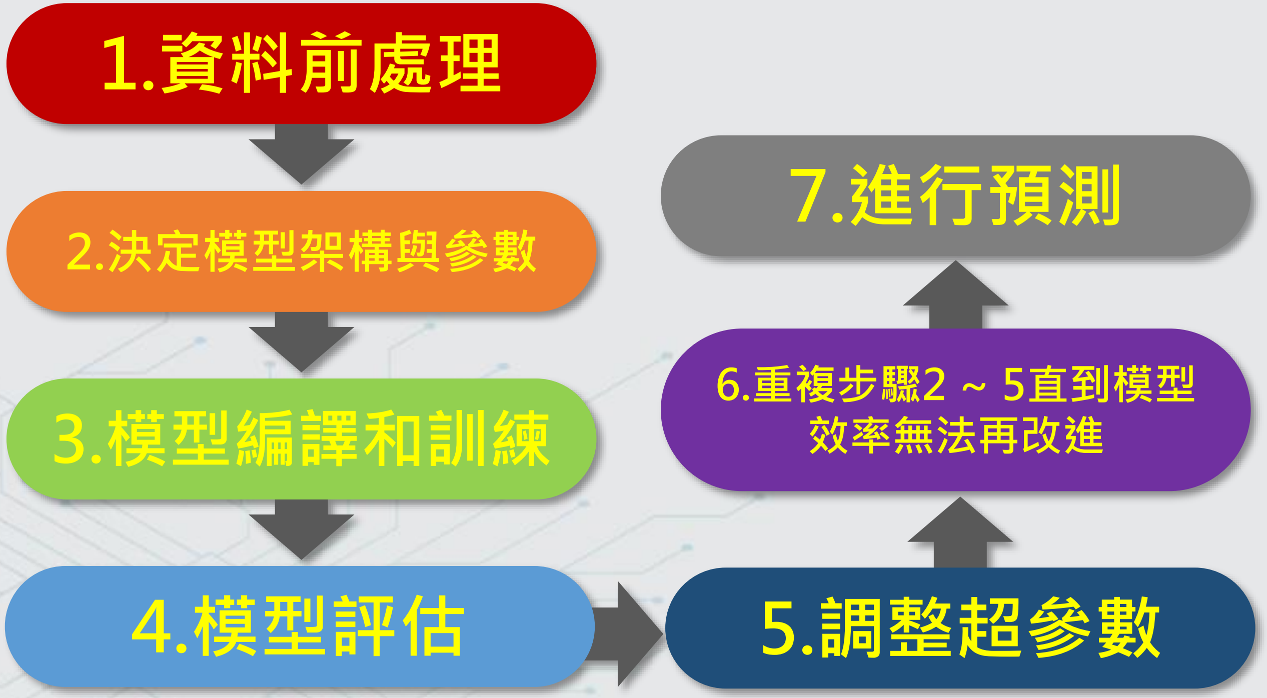
- 資料檔案：train.7z, test.7z, trainLabels.csv和sampleSubmission.csv
- train.7z和test.7z為png圖檔壓縮
- trainLabels.csv檔內容為圖檔對應的答案





	A	B
1	id	label
2	1	frog
3	2	truck
4	3	truck
5	4	deer
6	5	automobile
7	6	automobile
8	7	bird
9	8	horse
10	9	ship
11	10	cat

CNN模型建置流程



1.資料前處理

2.決定模型架構與參數

3.模型編譯和訓練

4.模型評估

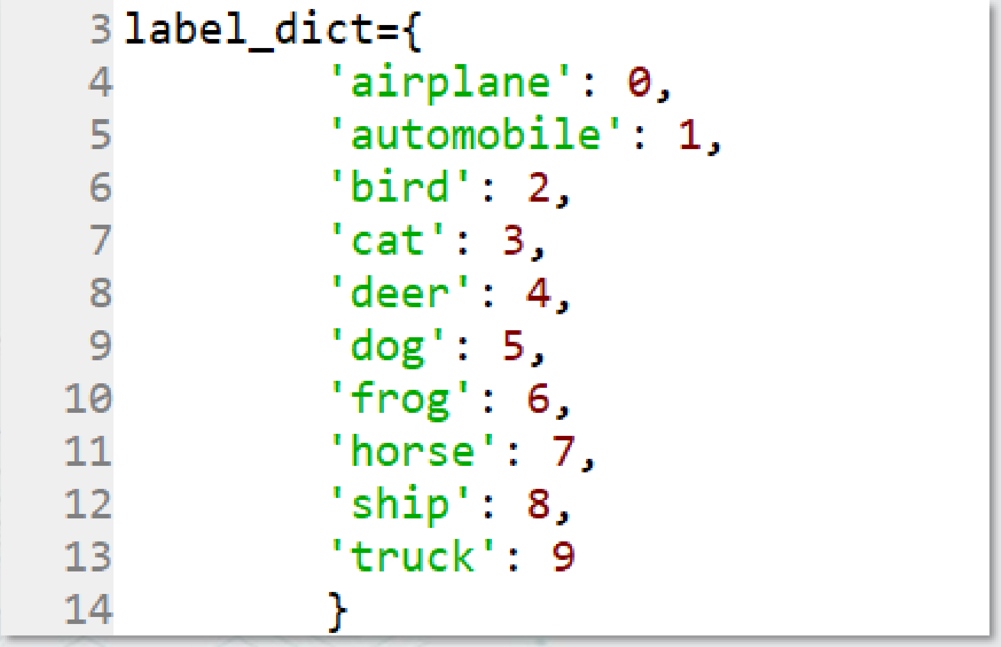
5.調整超參數

6.重複步驟2 ~ 5直到模型效率無法再改進

7.進行預測

資料前處理

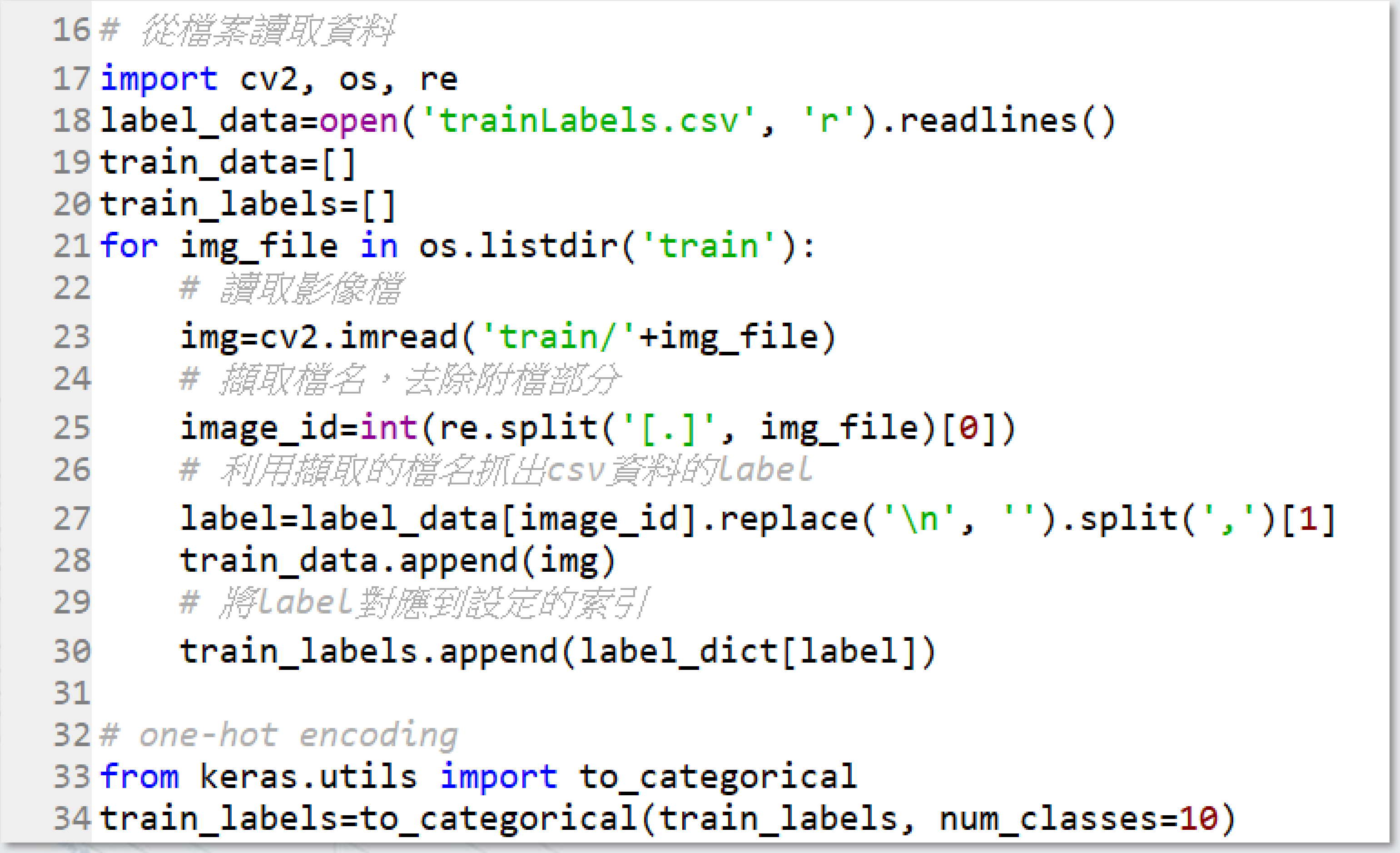
- 建立dictionary將類別文字對應到0~9的數字



```
label_dict={
        'airplane': 0,
        'automobile': 1,
        'bird': 2,
        'cat': 3,
        'deer': 4,
        'dog': 5,
        'frog': 6,
        'horse': 7,
        'ship': 8,
        'truck': 9
        }
```

資料前處理

- 從圖檔和trainLabels.csv讀進訓練資料和標記

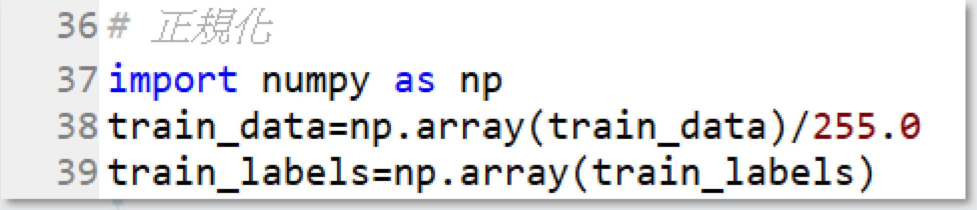


```python
# 從檔案讀取資料
import cv2, os, re
label_data=open('trainLabels.csv', 'r').readlines()
train_data=[]
train_labels=[]
for img_file in os.listdir('train'):
    # 讀取影像檔
    img=cv2.imread('train/'+img_file)
    # 擷取檔名，去除附檔部分
    image_id=int(re.split('[.]', img_file)[0])
    # 利用擷取的檔名抓出csv資料的label
    label=label_data[image_id].replace('\n', '').split(',')[1]
    train_data.append(img)
    # 將label對應到設定的索引
    train_labels.append(label_dict[label])

# one-hot encoding
from keras.utils import to_categorical
train_labels=to_categorical(train_labels, num_classes=10)
```

資料前處理

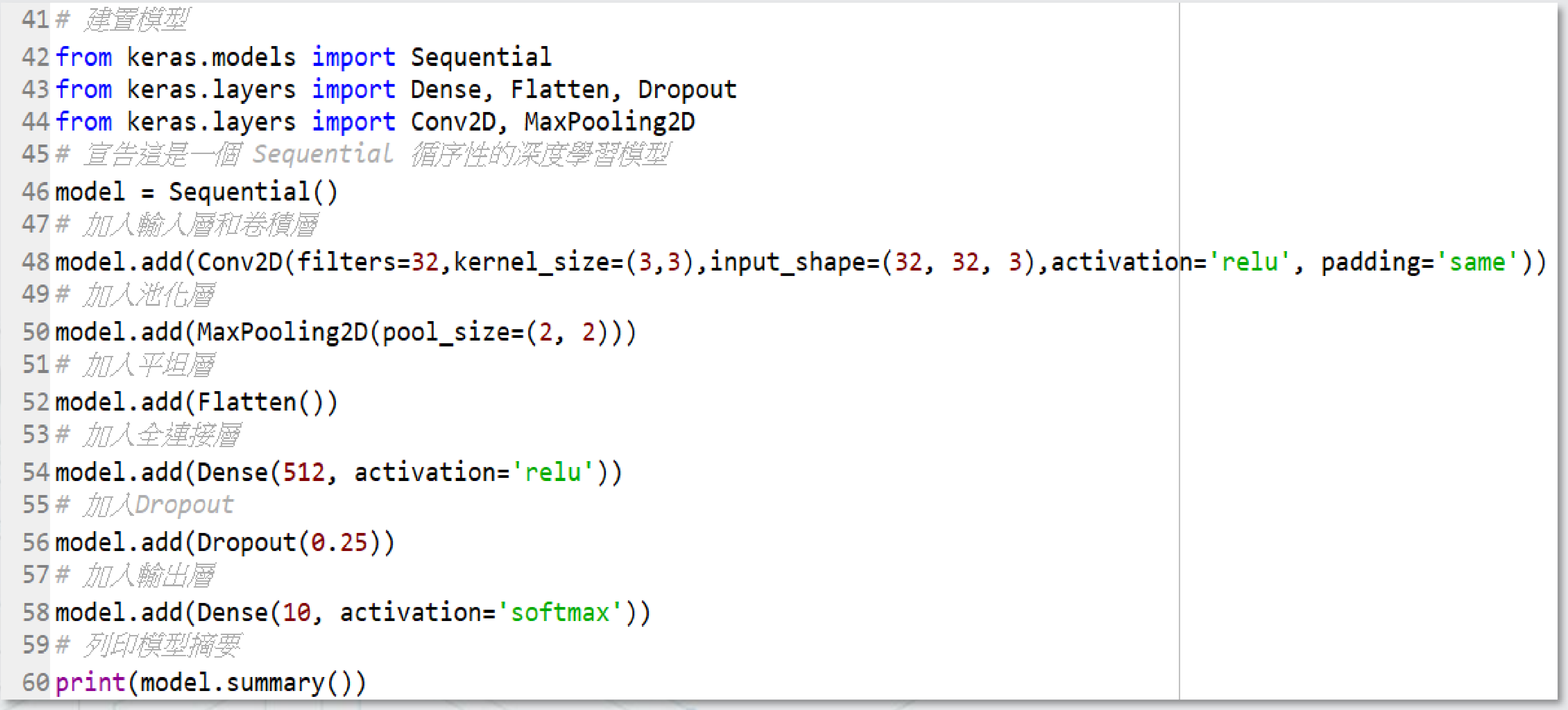
- 資料正規化



```python
# 正規化
import numpy as np
train_data=np.array(train_data)/255.0
train_labels=np.array(train_labels)
```

決定模型架構與參數

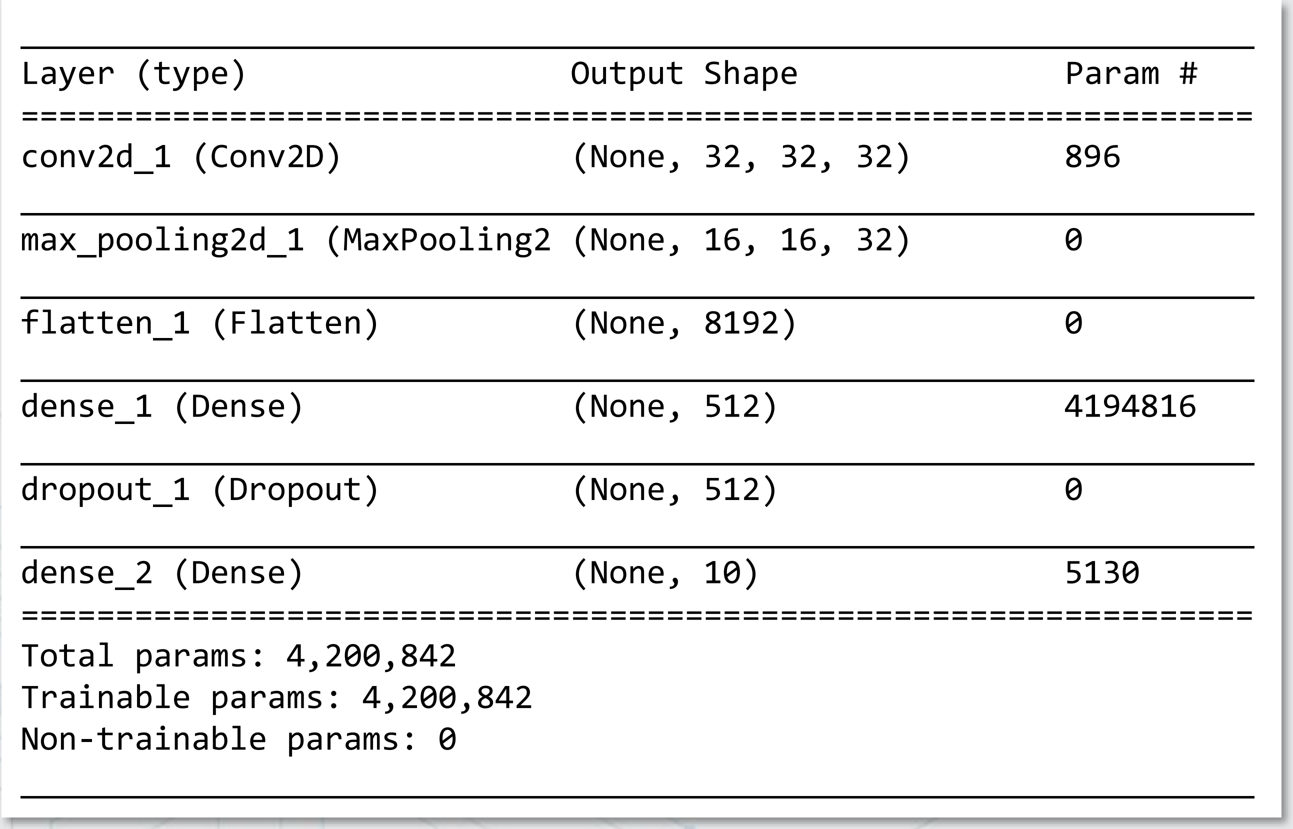
› 模型建置



```
# 建置模型
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout
from keras.layers import Conv2D, MaxPooling2D
# 宣告這是一個 Sequential 循序性的深度學習模型
model = Sequential()
# 加入輸入層和卷積層
model.add(Conv2D(filters=32,kernel_size=(3,3),input_shape=(32, 32, 3),activation='relu', padding='same'))
# 加入池化層
model.add(MaxPooling2D(pool_size=(2, 2)))
# 加入平坦層
model.add(Flatten())
# 加入全連接層
model.add(Dense(512, activation='relu'))
# 加入Dropout
model.add(Dropout(0.25))
# 加入輸出層
model.add(Dense(10, activation='softmax'))
# 列印模型摘要
print(model.summary())
```

決定模型架構與參數

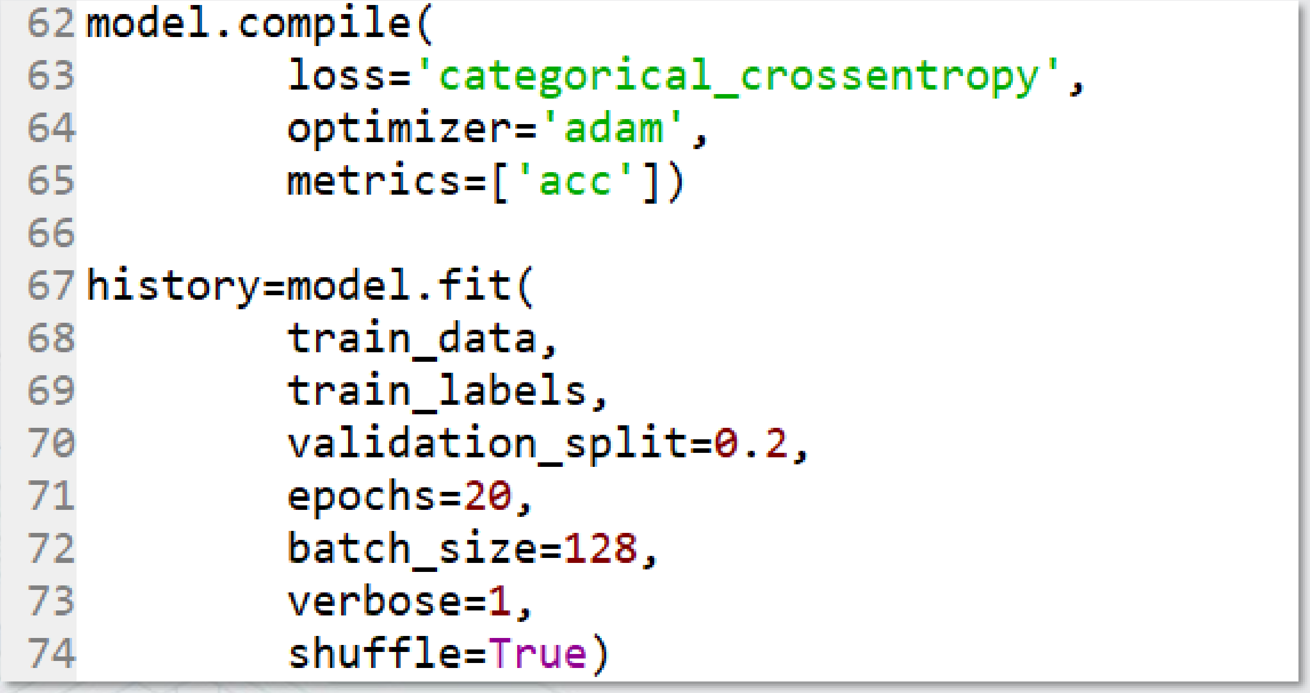
- 模型架構摘要



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               4194816
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 4,200,842
Trainable params: 4,200,842
Non-trainable params: 0
_________________________________________________________________
```

模型編譯和訓練

- 設定參數，訓練模型

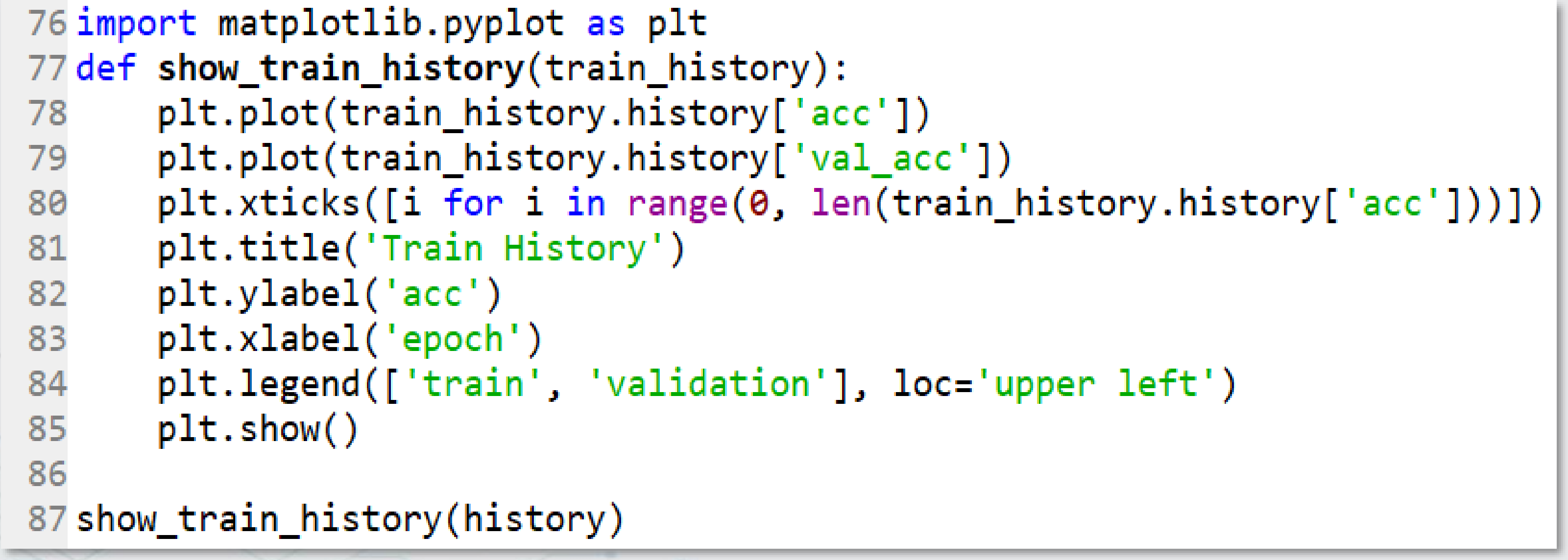


```
model.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['acc'])

history=model.fit(
        train_data,
        train_labels,
        validation_split=0.2,
        epochs=20,
        batch_size=128,
        verbose=1,
        shuffle=True)
```

模型評估

- 顯示訓練歷程

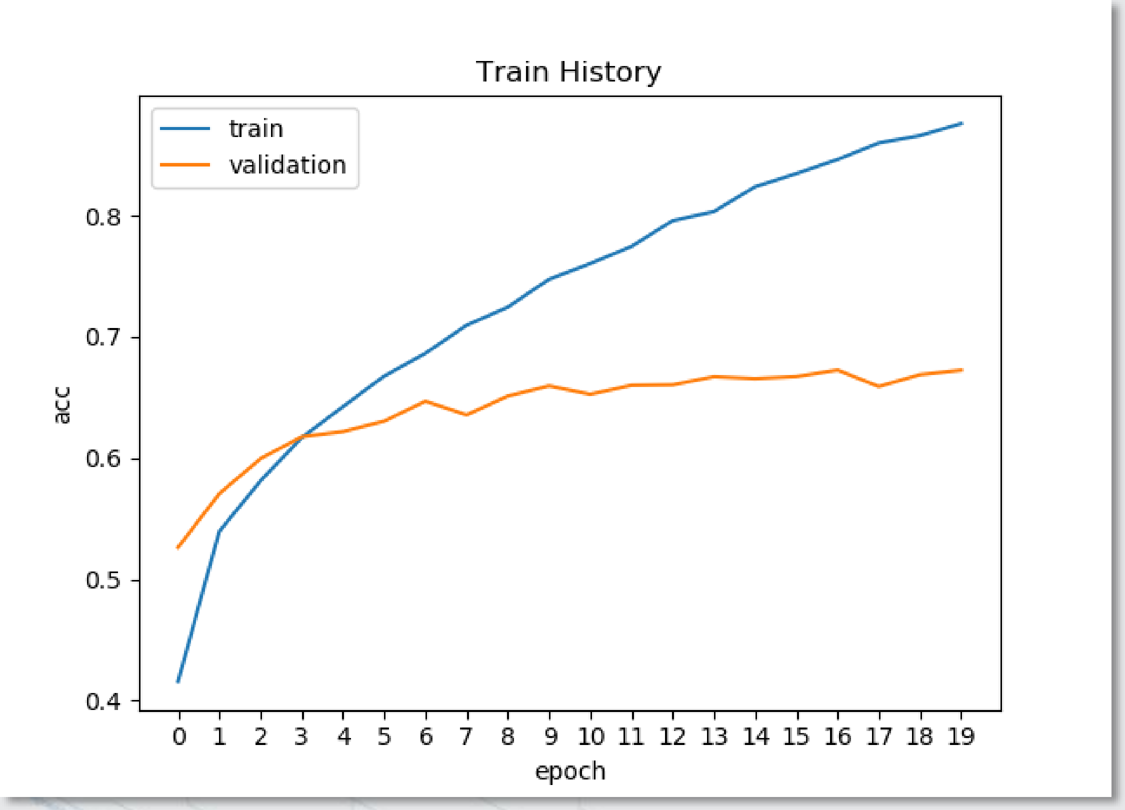


```python
import matplotlib.pyplot as plt
def show_train_history(train_history):
    plt.plot(train_history.history['acc'])
    plt.plot(train_history.history['val_acc'])
    plt.xticks([i for i in range(0, len(train_history.history['acc']))])
    plt.title('Train History')
    plt.ylabel('acc')
    plt.xlabel('epoch')
    plt.legend(['train', 'validation'], loc='upper left')
    plt.show()

show_train_history(history)
```

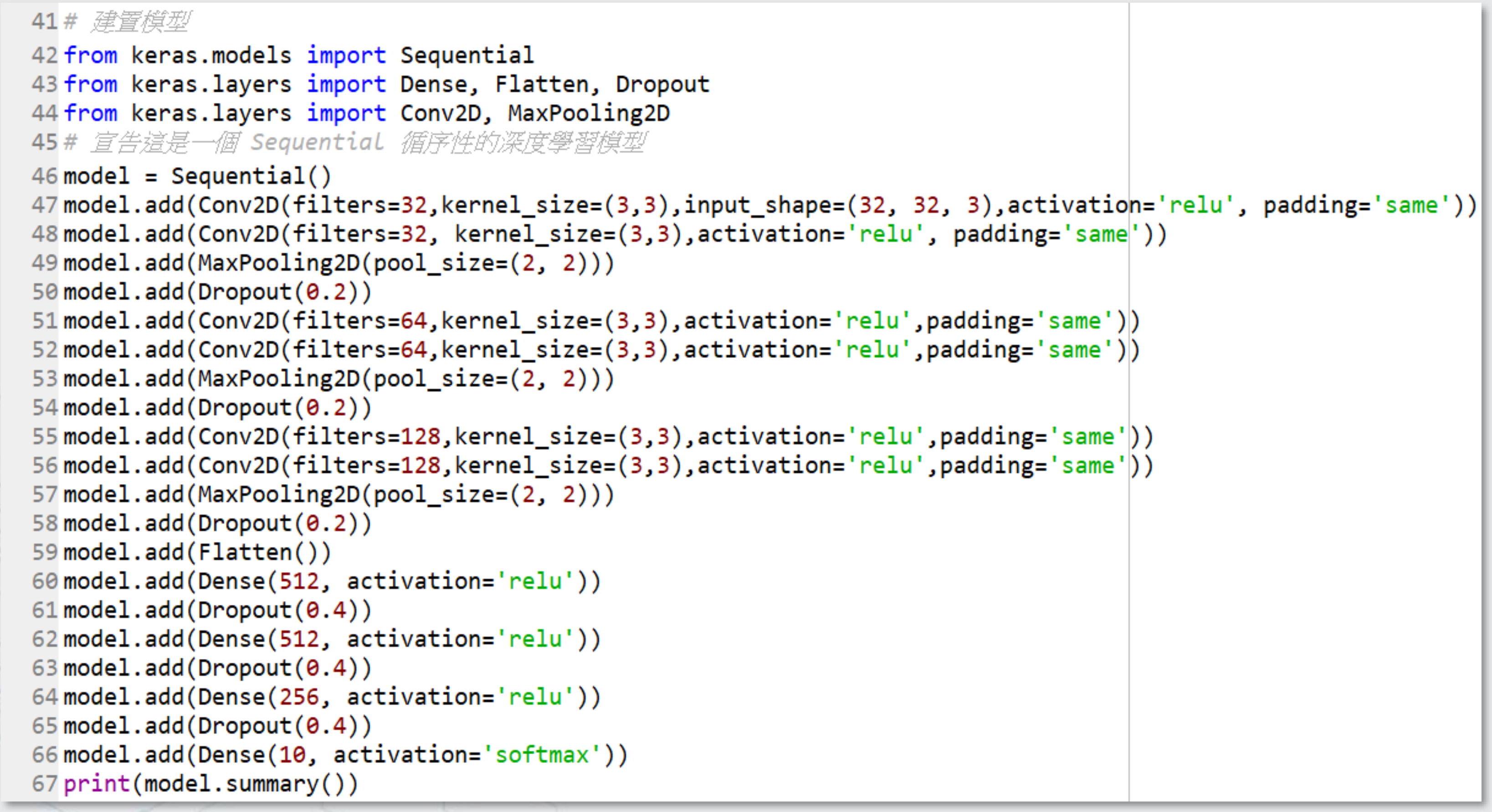
模型評估

- loss：0.3531 – acc：**0.8759**
- val_loss：1.1197 - val_acc：**0.6725**



調整超參數

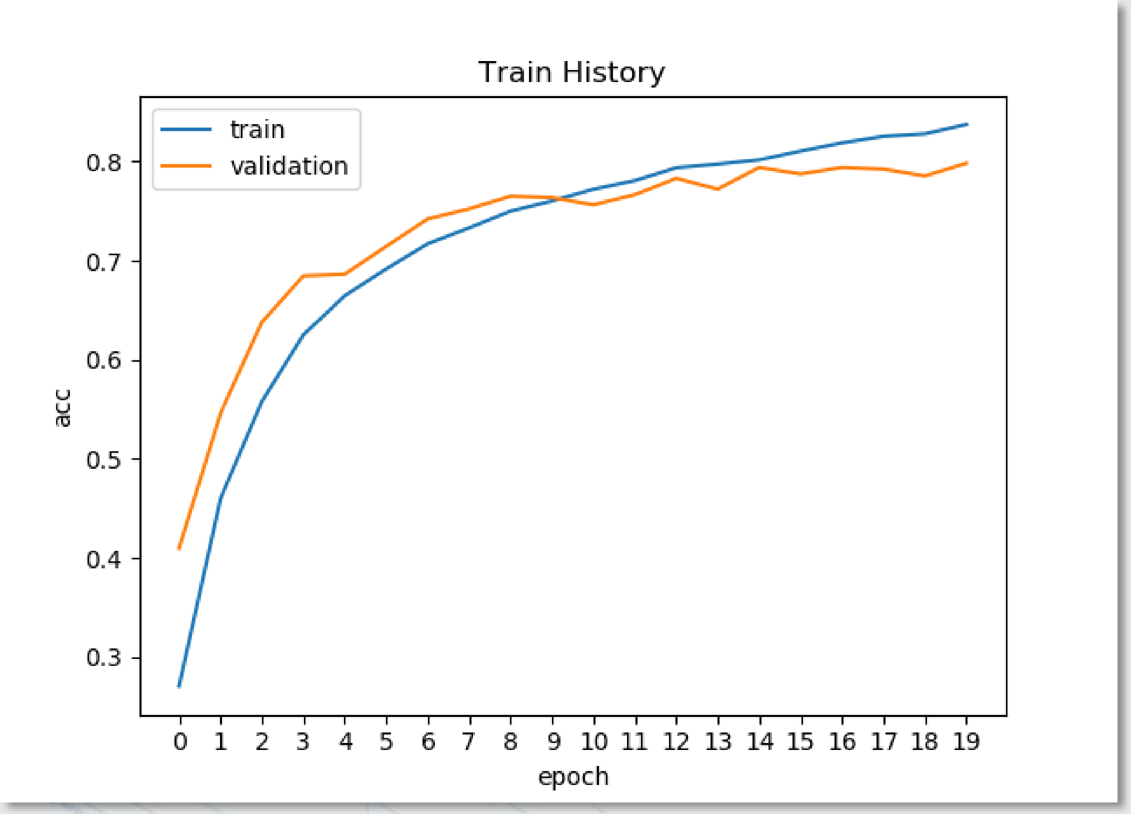
- 增加卷積層、增加及調高Dropout的值



```
# 建置模型
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout
from keras.layers import Conv2D, MaxPooling2D
# 宣告這是一個 Sequential 循序性的深度學習模型
model = Sequential()
model.add(Conv2D(filters=32,kernel_size=(3,3),input_shape=(32, 32, 3),activation='relu', padding='same'))
model.add(Conv2D(filters=32, kernel_size=(3,3),activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu',padding='same'))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters=128,kernel_size=(3,3),activation='relu',padding='same'))
model.add(Conv2D(filters=128,kernel_size=(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(10, activation='softmax'))
print(model.summary())
```

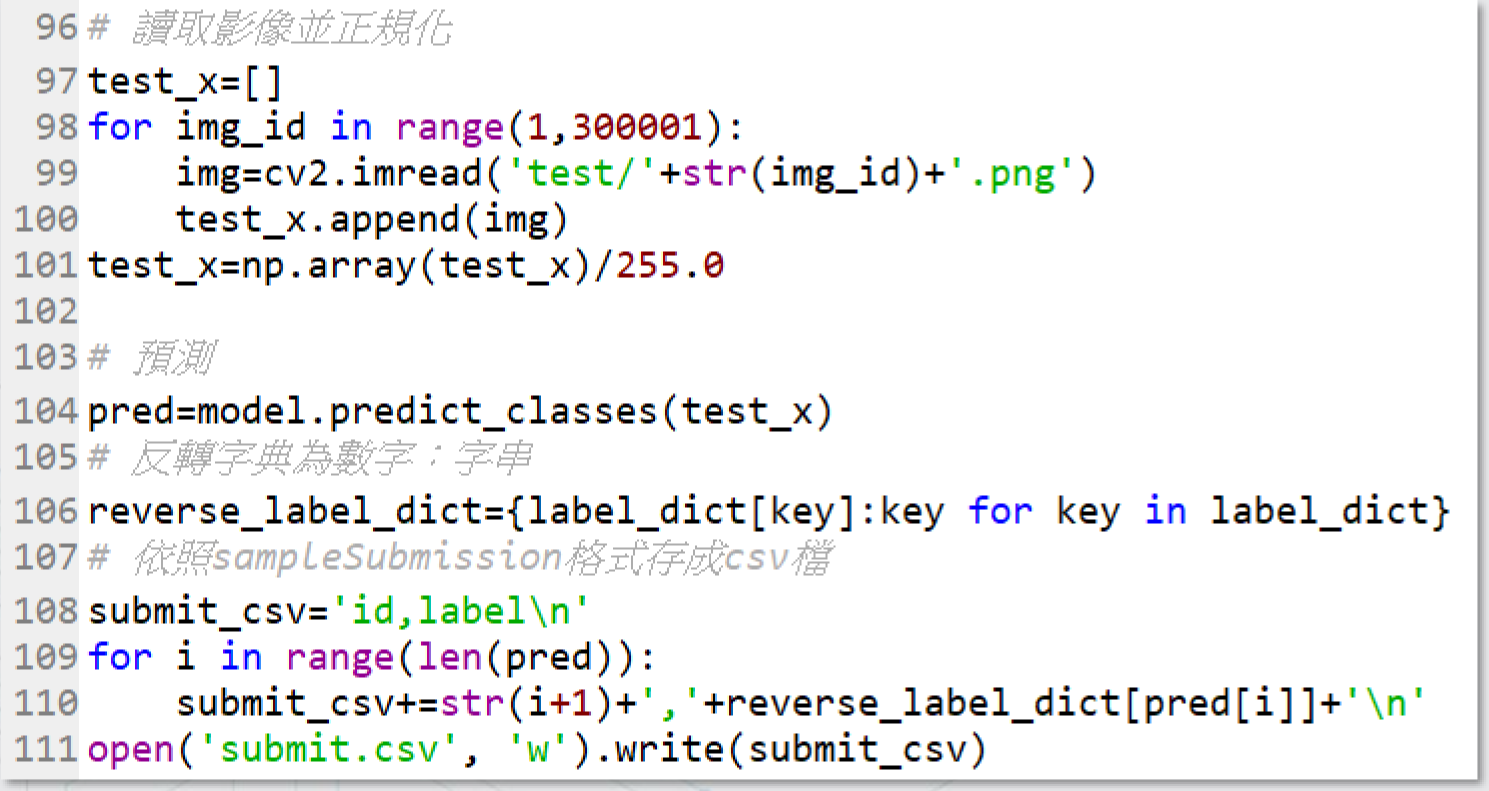
模型評估

- loss：0.4817 – acc：**0.8376**
- val_loss：0.6091 - val_acc：**0.7982**



進行預測

- 將預測結果寫入檔案，上傳Kaggle平台

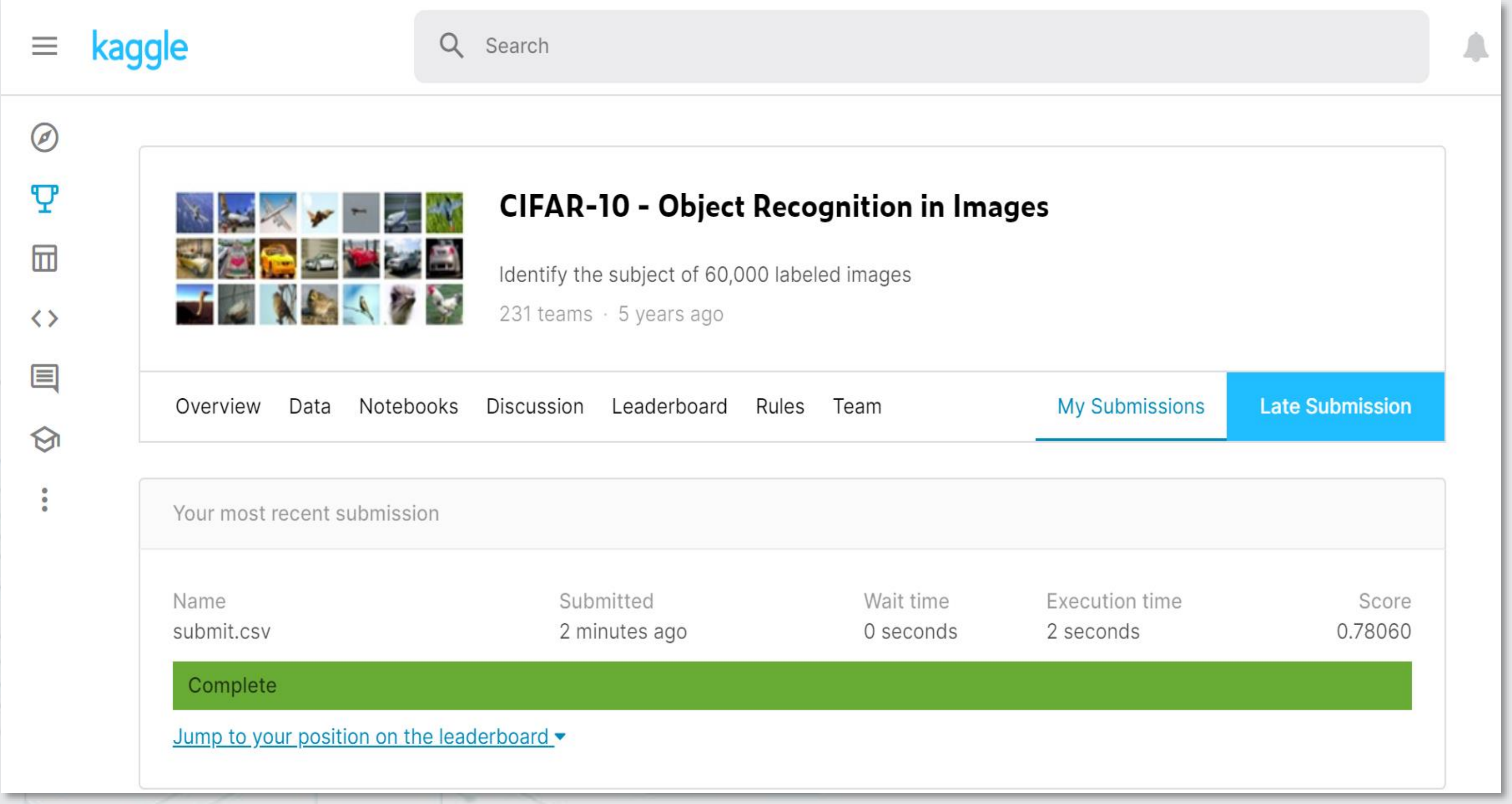


```
# 讀取影像並正規化
test_x=[]
for img_id in range(1,300001):
    img=cv2.imread('test/'+str(img_id)+'.png')
    test_x.append(img)
test_x=np.array(test_x)/255.0

# 預測
pred=model.predict_classes(test_x)
# 反轉字典為數字：字串
reverse_label_dict={label_dict[key]:key for key in label_dict}
# 依照sampleSubmission格式存成csv檔
submit_csv='id,label\n'
for i in range(len(pred)):
    submit_csv+=str(i+1)+','+reverse_label_dict[pred[i]]+'\n'
open('submit.csv', 'w').write(submit_csv)
```

進行預測

- 將預測結果寫入檔案，上傳Kaggle平台



CIFAR-10 - Object Recognition in Images

Identify the subject of 60,000 labeled images

231 teams · 5 years ago

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submit.csv	2 minutes ago	0 seconds	2 seconds	0.78060

Complete

Jump to your position on the leaderboard ▾