各類蔬果CNN辨識系統

109034531 黃怡菁

Agenda

01 5W1H

02 DATASET

03 MODEL

04 CONCLUSION

PART 01

5W1H

5W1H

越來越多店家採用自助結帳機台，結帳時顧客仍須一一將購物車內的物品拿出來逐一掃瞄條碼，可能因為不熟悉機台操作方式而增加結帳時間

WHY

任何購物後須使用自助結帳機台的場所，例如：大型商場、超市

WHERE

舉凡有在用實體商場或超市購買蔬果的人們

WHO

解決結帳時排隊或是條碼標示不清的困擾

WHAT

任何需要使用自助結帳機台的時候

WHEN

利用類神經網路訓練並建立一個可以識別水果種類的系統

HOW


PART 02
DATASET

DATASET

資料集來源

由Kaggle公開數據集中取得水果的圖像資料集，此次是使用「Fruit Recognition」，共有33種類別，訓練集總共16872張照片，測試集總共5641張照片。

```
!kaggle datasets list -s Fruit
```

```
!kaggle datasets download -d sshikamaru/fruit-recognition --force
```

```
!ls
```

```
fruit-recognition.zip  model01.h5  model01_history.pkl  sample_data
```

```
!unzip fruit-recognition.zip
```

DATASET

Data Pre-processing

每張圖片大小統一為100 x 100，將33個資料類別加上標籤0至32，並將圖片特徵標準化，全部除以255，轉為0到1之間的數據，幫助資料訓練。

```
X = []
Y = [] #33
# enumerate 顯示序號
for i,labelname in enumerate(os.listdir("train/train")):
    print(i,labelname)
    for imgname in os.listdir("train/train/{}".format(labelname)):
        img = image.load_img('train/train/{}/{}'.format(labelname,imgname))
        X.append(image.img_to_array(img)/255) # append新增資料。將img array/255轉成0~1之間
        y = [0 for j in range(33)]
        y[i] = 1
        Y.append(y)
```

```
x_val = []
for imgname in os.listdir("test/test"):
    img = image.load_img('test/test/{}'.format(imgname))
    x_val.append(image.img_to_array(img)/255) #新增資料
x_val = np.array(x_val)
```

PART 03

MODEL

MODEL

卷積神經網路 Model 1

- 卷積層：Conv2D
- 池化層：MaxPool2D
- 激活函數：relu
- 最後一層激活函數：sigmoid
- 優化器：Adam(lr = 0.01)
- 損失函數：categorical

```
#  建一個CNN  model
model  =  Sequential()
model.add(Conv2D(32,  kernel_size=(3,3),  input_shape=(100,100,3),  activation="relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))  #  30%不修正，避免過擬合
model.add(Conv2D(64,  kernel_size=(3,3),  activation="relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(32,activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(33,activation="sigmoid"))


optim  =  optimizers.Adam(lr  =  0.01)
model.compile(loss="categorical_crossentropy",  optimizer=optim,  metrics=["accuracy"])

model.summary()
```

MODEL

卷積神經網路 Model 1

◆訓練集8成，測試集2成

◆設定early_stopping，避免overfitting，patience= 5

◆開始訓練、batch_size設定為1024，epochs= 100

```
early_stopping  =  EarlyStopping(monitor="val_loss",patience=5,verbose=1)
```

```
history  =  model.fit(x_train,y_train,
    epochs=100,batch_size=1024,verbose=1,
    callbacks=[early_stopping],
    validation_data=(x_test,y_test))
```

MODEL

卷積神經網路 Model 1

◆在Epoch 35/100時出現early stopping，accuracy= 90.82%，val_accuracy= 99.70%

```
Epoch 35/100
14/14 [==============================] - 155s 11s/step - loss: 0.2596 - accuracy: 0.9082 - val_loss: 0.0228 - val_accuracy: 0.9970
Epoch 00035: early stopping
```

```
score  =  model.evaluate(x_test,y_test,verbose=0)
print(score)
y_val  =  model.predict(x_val)
print(y_val)
```

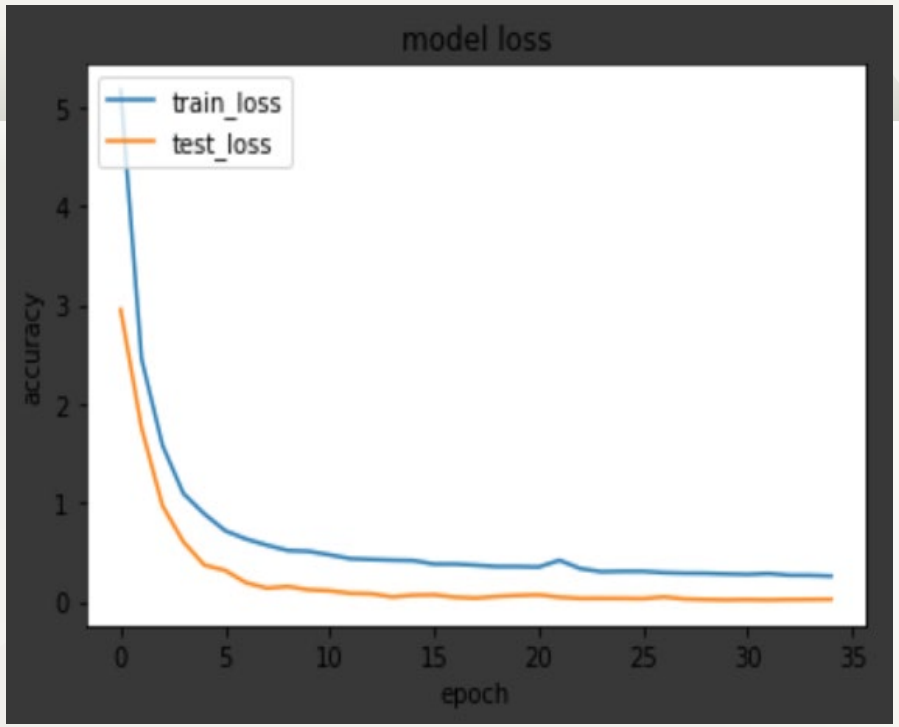
```
[0.02279523015022278, 0.9970335364341736]
[[2.2292754e-06 1.3314720e-12 9.2935413e-02 ... 8.0686164e-08
```

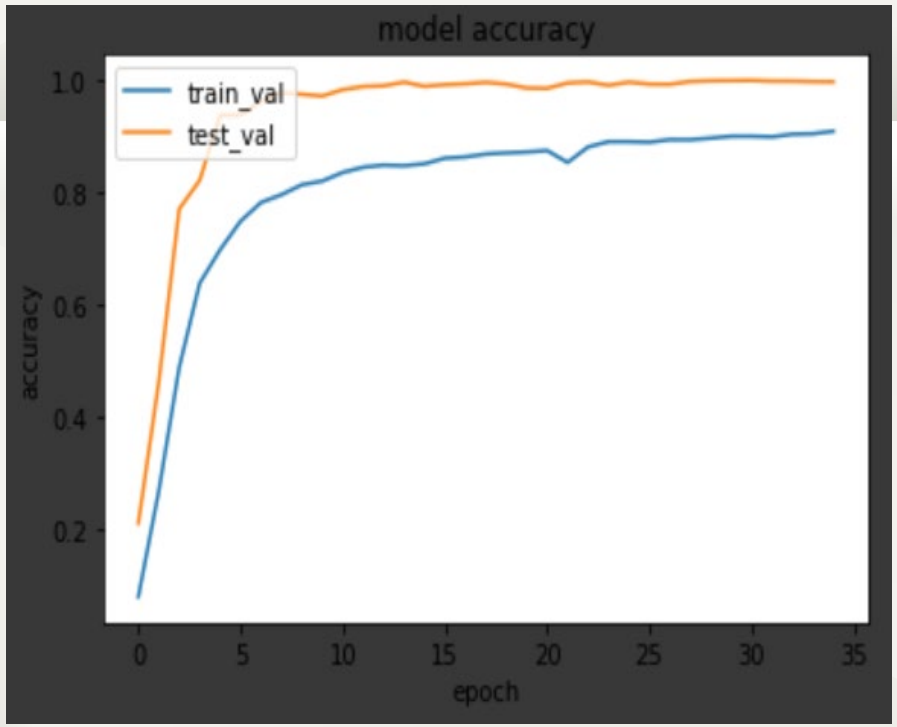
MODEL

卷積神經網路 Model 1

◆將訓練與測試結果繪製成圖

◆發現train_loss高於test_loss，出現underfitting的狀況





MODEL

卷積神經網路 Model 2

- 卷積層：Conv2D
- 池化層：MaxPool2D
- 激活函數：relu
- 最後一層激活函數：softmax
- 優化器：Adam(lr = 0.001)
- 損失函數：categorical

```
input   =  Input(shape=(100,100,3))

conv1_1  =  Conv2D(32,  kernel_size=(3,3),  activation="relu")(input)
conv1_2  =  Conv2D(64,  kernel_size=(3,3),  activation="relu")(conv1_1)
pool1_3  =  MaxPool2D(pool_size=(2,2))(conv1_2)
drop1_4  =  Dropout(0.3)(pool1_3)
flat1_5  =  Flatten()(drop1_4)

conv2_1  =  Conv2D(32,  kernel_size=(5,5),  activation="relu")(input)
conv2_2  =  Conv2D(64,  kernel_size=(5,5),  activation="relu")(conv2_1)
pool2_3  =  MaxPool2D(pool_size=(2,2))(conv2_2)
drop2_4  =  Dropout(0.3)(pool2_3)
flat2_5  =  Flatten()(drop2_4)

conv3_1  =  Conv2D(32,  kernel_size=(7,7),  activation="relu")(input)
conv3_2  =  Conv2D(64,  kernel_size=(7,7),  activation="relu")(conv3_1)
pool3_3  =  MaxPool2D(pool_size=(2,2))(conv3_2)
drop3_4  =  Dropout(0.3)(pool3_3)
flat3_5  =  Flatten()(drop3_4)

merge   =  Concatenate()([flat1_5,flat2_5,flat3_5])
hidden   =  Dense(128,activation="relu")(merge)
output   =  Dense(33,activation="softmax")(hidden)
model   =  Model(inputs=input,   outputs=output)
optim   =  optimizers.Adam(lr   =  0.001)

model.compile(loss="categorical_crossentropy",   optimizer=optim,   metrics=["accuracy"])

model.summary()
```

MODEL

卷積神經網路 Model 2

◆設定early_stopping，避免overfitting，patience= 5

◆開始訓練、batch_size= 512，epochs= 1000

```
history  =  model.fit(x_train, y_train,
    epochs=1000, batch_size=512, verbose=1,
    callbacks=[early_stopping],
    validation_data=(x_test, y_test))
```

◆在Epoch 59/1000時出現early stopping，test loss : 0.00013364345068112016，test accuracy : 1

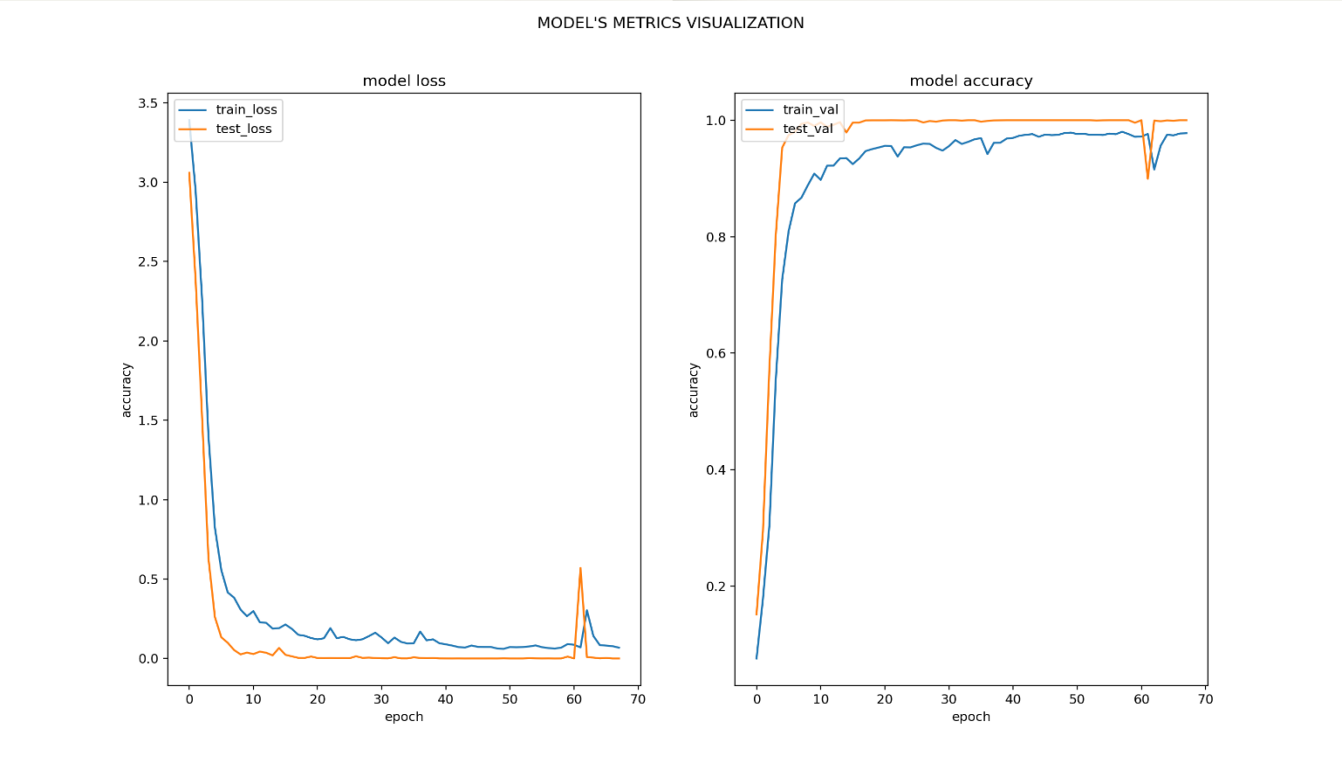
```
Epoch 59/1000
24/24 [==============================] - 276s 12s/step - loss: 0.0688 - accuracy: 0.9739 - val_loss: 1.2249e-04 - val_accuracy: 1.0000
Epoch 00059: early stopping
```

MODEL

卷積神經網路 Model 2

◆將訓練與測試結果繪製成圖

◆模型的loss沒有再繼續掉下去，後面幾乎持平，因此判定沒有under fitting



PART 04

CONCLUSION

CONCLUSION

- 本次僅針對水果作訓練及系統建置，日後可以針對各種蔬果進行訓練，使得此系統更加完善實用。
- 在樣本數上可以再進行擴大，增加準確率與應用。
- CNN模型的辨識準確率在大部分測試中都有不錯的表現。


THANK YOU