

# 愛文芒果不良品分類競賽

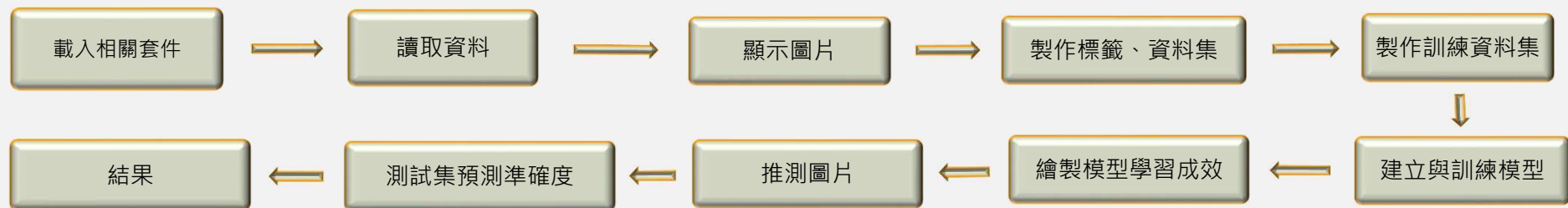
電子碩一 109368005 陳宥安

電子碩一 109368018 林祐民

電子碩一 109368082 胡家齊

電子碩一 109368083 吳怡瑩

## 流程圖



# 載入相關套件

▶ # 資料處理套件

```
import cv2
import csv
import random
import time
import numpy as np
import pandas as pd
import matplotlib.image as mpimg # mpimg 用於讀取圖片
import matplotlib.pyplot as plt # plt 用於顯示圖片
from keras.preprocessing import image
import seaborn as sns
import os
import random
```

[ ] # 設定顯示中文字體

```
from matplotlib.font_manager import FontProperties
#import FontProperties from matplotlib.font_manager
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']
```

[ ] # Keras深度學習模組套件

```
from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
from keras.optimizers import RMSprop
from keras.models import Sequential
from keras import utils as np_utils
from keras import backend as K
from keras import optimizers
```

[ ] # tensorflow深度學習模組套件

```
from tensorflow.keras import models, layers
from tensorflow import keras
import tensorflow as tf
```

[ ] from google.colab import drive  
drive.mount('/content/drive')

[ ] # 打印相關版本

```
print(pd.__version__)
print(tf.__version__)
```

[ ] # 查看通道位置

```
print(K.image_data_format())
```

[ ] !ls

## 讀取資料(1/2)

```
[ ] os.getcwd()
```

```
# 讀取資料集標籤檔
Sample_label = pd.read_csv("/content/drive/MyDrive/北科研究所/AlMango2/C2_TrainDev/train.csv",encoding="utf8")
```

```
[ ] # 顯示資料集標籤檔
Sample_label.head()
```

	38414.jpg	46	146	576	574	不良- 機械傷害	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13	Unnamed: 14	Unnamed: 15	Unnamed: 16	Unnamed: 17	Unnamed: 18	Unnamed: 19	Unnamed: 20
0	03182.jpg	581	277	97	93	不良- 機械傷害	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	29863.jpg	514	538	117	144	不良- 機械傷害	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

## 讀取資料(2/2)

```
[ ] # 串接圖片檔的路徑
    Sample_pics_path = os.path.join("/content/drive/MyDrive/北科研究所/AIMango2/C2_TrainDev/Train")

[ ] # 讀取路徑中的圖片
    train_mango_fnames = os.listdir(Sample_pics_path)

▶ # 檢視是否有讀取到圖片
    print(train_mango_fnames[0])
    print(train_mango_fnames[1])
    print(train_mango_fnames[2])

[ ] # 用於瀏覽標籤檔的概況
    label_Survey = pd.read_csv("/content/drive/MyDrive/北科研究所/AIMango2/train_Label_mod_label.csv", encoding="utf8")

[ ] label_Survey.head()

[ ] sns.countplot(label_Survey['not_good_reason'], hue = label_Survey["not_good_reason"])

[ ] sector = label_Survey.groupby('not_good_reason')
    sector.size()
```

## 顯示圖片

```
[ ] # 讀取圖檔
img = mpimg.imread("/content/drive/MyDrive/北科研究所/AIMango2/C2_TrainDev/Train/00001.jpg")
# 查看資料型態
print(type(img))
# 顯示圖片的比例
print(img.shape)

[ ] # 把圖片的比例壓縮至800x800
res = cv2.resize(img,(800,800),interpolation=cv2.INTER_LINEAR)

[ ] # 顯示壓縮後圖片的比例
res.shape

[ ] # 顯示原圖的芒果照片
plt.imshow(img)
plt.axis('off')
plt.show()

[ ] # 顯示壓縮過原圖的芒果照片
plt.imshow(res)
plt.axis('off')
plt.show()
```

## 製作標籤、資料集(1/3)

```
[ ] csvfile = open('/content/drive/MyDrive/北科研究所/AIMango2/C2_TrainDev/train.csv')
reader = csv.reader(csvfile)
```

```
[ ] # 讀取csv標籤
labels = []
for line in reader:
    tmp = [line[0],line[5]]
    # print tmp
    labels.append(tmp)

csvfile.close()
```

```
[ ] picnum = 800 #len(labels)
print("芒果圖片數量: ",picnum)
```

```
[ ] labels[1]
```

```
[ ] # 轉換圖片的標籤
x = []
y = []

for i in range(len(labels)):
    labels[i][1] = labels[i][1].replace("不良-著色不佳","0")
    labels[i][1] = labels[i][1].replace("不良-乳汁吸附","1")
    labels[i][1] = labels[i][1].replace("不良-炭疽病","2")
    labels[i][1] = labels[i][1].replace("不良-黑斑病","3")
```

## 製作標籤、資料集(2/3)

```
[ ] # 隨機讀取圖片
a = 0
items= []

for a in range(0,picnum):
    items.append(a)

[ ] # 製作訓練用資料集及標籤
for i in random.sample(items,picnum):
    img = cv2.imread("/content/drive/MyDrive/北科研究所/AlMango2/C2_TrainDev/Train/" + labels[i+1][0] )
    res = cv2.resize(img,(800,800),interpolation=cv2.INTER_LINEAR)
    res = img_to_array(res)
    X.append(res)
    y.append(labels[i+1][1])

y_label_org = y

[ ] print(len(X))
    print(len(y))

[ ] # 轉換至array的格式
X = np.array(X)
y = np.array(y)

# 轉換至float的格式
for i in range(len(X)):
    X[i] = X[i].astype('float32')
```



## 製作標籤、資料集(3/3)

```
[ ] # 打映圖片訓練集的概況
    # print(X[0])
    print(type(X))
    print(X.shape)
    print(X[0].shape)
    print(type(X[0]))
```

```
[ ] # 將標籤轉換至float格式
    y = tf.strings.to_number(y, out_type=tf.float32)

    # 標籤進行one-hotencoding
    y = np_utils.to_categorical(y, num_classes = 5)
```

```
[ ] #檢查轉換結果
    y[0]
```

## 製作訓練資料集

```
[ ] # 分配訓練集及測試集比例
    x_train = x[:600]
    y_train = y[:600]
    x_test = x[600:]
    y_test = y[600:]

[ ] y_train_label = [0.,0.,0.,0.,0.]
    y_test_label = [0.,0.,0.,0.,0.]

    for i in range(0,len(y_train)):
        y_train_label = y_train[i] + y_train_label

    for i in range(0,len(y_test)):
        y_test_label = y_test[i] + y_test_label

[ ] print(y_train_label)
    print(y_test_label)
```

## 建立與訓練模型(1/2)

```
[ ] # 建立深度學習CNN Model

model = tf.keras.Sequential()

model.add(layers.Conv2D(16,(3,3),
                        strides=(1,1),
                        input_shape=(800, 800, 3),
                        padding='valid',
                        activation='relu',
                        ))

model.add(layers.BatchNormalization())

model.add(layers.MaxPooling2D(pool_size=(2,2),strides=None))

model.add(layers.Conv2D(32,(3,3),
                        strides=(1,1),
                        padding='valid',
                        activation='relu',
                        ))

model.add(layers.MaxPooling2D(pool_size=(2,2),strides=None))

model.add(layers.Conv2D(64,(3,3),
                        strides=(1,1),
                        padding='valid',
                        activation='relu',
                        ))
```

## 建立與訓練模型(2/2)

```
[ ] adam = tf.keras.optimizers.Adam(lr=5)
    model.compile(optimizer='adam',
                  loss=tf.keras.losses.CategoricalCrossentropy(),
                  metrics=['acc'])

[ ] # 設定超參數HyperParameters
    batch_size = 32
    epochs = 128

[ ] # 檔名設定
    file_name = str(epochs)+'_'+str(batch_size)

[ ] # 加入EarlyStopping以及Tensorboard等回調函數
    CB = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=20)
    TB = keras.callbacks.TensorBoard(log_dir='./log'+ "_" +file_name, histogram_freq=1)

[ ] history = model.fit(
    x = x_train , y = y_train,
    batch_size = batch_size,
    epochs = epochs,
    validation_split = 0.2,
    callbacks = [CB]
)
```

## 繪製模型學習成效

```
[ ] def plot_learning_curves(history):  
    pd.DataFrame(history.history).plot(figsize=(8,5))  
    plt.grid(True)  
    plt.gca().set_ylim(0,1)  
    plt.show()
```

```
[ ] plot_learning_curves(history)
```

## 推測圖片

```
[ ] test_mango_dir = os.path.join("/content/drive/MyDrive/北科研究所/AlMango2/C2_TrainDev/Dev")
test_mango_fnames = os.listdir(test_mango_dir)

[ ] test_mango_fnames[0]

[ ] img_files = [os.path.join(test_mango_dir,f) for f in test_mango_fnames]
img_path = random.choice(img_files)

# 讀入待測試圖像並秀出
img = load_img(img_path, target_size=(800, 800)) # this is a PIL image
plt.title(img_path)
plt.grid(False)
plt.imshow(img)

[ ] labels = ['不良-乳汁吸附', '不良-機械傷害', '不良-炭疽病', '不良-著色不佳', '不良-黑斑病']

[ ] # 將圖像轉成模型可分析格式(800x800x3, float32)
x = img_to_array(img) # Numpy array with shape (800, 800, 3)
x = x.reshape((1,) + x.shape) # Numpy array with shape (1, 800, 800, 3)
x /= 255 # Rescale by 1/255

start = time.time() # 啟動計時器
result = model.predict(x) # 對輸入圖像進行推論(預測)
finish = time.time() # 結束計時器

pred = result.argmax(axis=1)[0]
pred_prob = result[0][pred]

print("Result = %f" %pred_prob) # 印出結果可能機率值(0.0 ~ 1.0)
print("Test time :%f second." %(finish-start)) # 印出推論時間

# 設定分類門檻值並印出推論結果
print("有 {:.2f}% 機率為{}".format(pred_prob * 100, labels[pred])) # 印出推論時間
```

# 測試集預測準確度

```
[ ] # 測試集標籤預測
y_pred = model.predict(x_test)

[ ] # 整體準確度
count = 0
for i in range(len(y_pred)):
    if(np.argmax(y_pred[i]) == np.argmax(y_test[i])): #argmax函数找到最大值的索引，即为其类别
        count += 1
score = count/len(y_pred)
print('正确率为: %.2f%%' % (score*100, '%'))

[ ] # 模型預測後的標籤
predict_label = np.argmax(y_pred,axis=1)
print(predict_label)
print(len(predict_label))

[ ] # 模型原標籤
true_label = y_label_org[600:]
true_label = np.array(true_label)
print(true_label)
print(len(true_label))

[ ] # 模型預測後的標籤
predictions = model.predict_classes(x_test)
print(predictions)
print(len(predictions))
```

# 結果



\*Only csv file is accepted

Public Leaderboard				Private Leaderboard			
檔名	Upload time	Evaluation result	Ranking				
0.5482399.csv YUMINLIN	2021-01-07 22:10:22	0.5482399	127/222				