

國立雲林科技大學
資訊管理所

機器學習專案報告

學生/學號：

張 靖 M10823001

陳逢麒 M10823028

平靖翔 M10823024

古力維 M10823040

中華民國 109 年 7 月

摘要

愛文芒果採收後依品質篩選為 A、B、C 三等級，依序為出口用、內銷用、加工用。然而愛文芒果現階段仍依靠人工篩選，這現象除了農村人口流失導致人力短缺，篩選芒果流程也因保鮮期壓縮地極短，導致篩選芒果階段約有 10% 的誤差，若以外銷金額估計，每年恐怕損失 1600 萬台幣，若能改善此部分則能大幅降低銷售成本及提高獲利及市占率。本研究透過演算法訓練資料進行測試資料的驗證準確度與降低誤差。使用 Python 撰寫類神經網路(Artificial Neural Network)進程式訓練模型，採用 BIIC Lab 所提供之 5600 筆芒果之圖像。透過此資料集來預測芒果的等級，使用 DenseNet 的預訓練模型訓練之下芒果的分類預測，並在 1600 張測試資料集上績效達到 0.7575。

關鍵字：機器學習、類神經網路、正規化、DenseNet

目錄

摘要	1
一、 緒論	1
1.1 動機	1
1.2 目的	1
二、 研究方法	2
三、 實驗	3
3.1 資料集	3
3.2 前置處理	3
3.3 實驗設計	4
四、 結論	6
五、 參考文獻	7

一、緒論

1.1 動機

本研究為了瞭解台灣重要出口農產品-愛文芒果，愛文芒果近年銷量不斷增長，不僅躍升為三大外銷高經濟生鮮果品之一，更外銷至其他國家拓展，例如：日本、中國、美國以及香港等地。雖然，台灣芒果較以往提高了知名度並拓展市佔率，卻還是遭遇其他同為芒果出口國的削價競爭，例如：菲律賓、泰國，因此諸多品種改良、採收後處理技術以及品牌行銷等提昇產品價值的工作，仍須仰賴科技的輔助，而其中優先改善的是採收後的處理技術。愛文芒果採收後依品質篩選為 A、B、C 三等級，依序為出口用、內銷用、加工用。然而愛文芒果現階段仍依靠人工篩選，這現象除了農村人口流失導致人力短缺，篩選芒果流程也因保鮮期壓縮地極短，導致篩選芒果階段約有 10% 的誤差，若以外銷金額估計，每年恐怕損失 1600 萬台幣，若能改善此部分則能大幅降低銷售成本及提高獲利及市占率。

因此本研究，採用 BIIC Lab 所提供之 5600 筆芒果之圖像。透過此資料集來預測芒果的等級。近年來，由於硬體設備的進步，類神經網路模型的運算得以被實現，人工智慧的再度崛起，使得各種行業紛紛投入其中，主要藉此減少人力的浪費以及提升產品生產的效率以及獲取更精確的資訊。為此，本研究採用類神經網路模型來預測芒果等級。

1.2 目的

本研究主要以機器學習為基礎，透過 Python 撰寫類神經網路之影像辨識程式，以此進行模型訓練，透過建立影像辨識演算法模型之類神經網路調整隱藏層層數、節點數、激活函數、優化器、loss 函數，並對愛文芒果影像進行三種等級分類，提出本研究之解決方案，以達到此資料最佳之預測績效。

二、研究方法

本研究採取「DenseNet121」預訓練 model 方式進行，首先將資料載入後，接下來執行資料前處理，如：正規化、資料增加等，對影響作特徵提取，針對 Model 參數做調整，調整完成後開始訓練 model，最後針對預測資料集放入做資料預測。



圖 1 研究流程圖

三、實驗

3.1 資料集

本研究利用 BIIC Lab 所提供之芒果影像，此資料分為訓練資料集 5600 筆，驗證資料集 800 筆，預測資料集 1600 筆，屬性特徵為矩陣，無缺失值，愛文芒果依品質篩選共分為 A、B、C 三等級。

表 1

資料集

數據及特徵	影像	訓練資料集	5600
驗證資料集	800	預測資料集	1600
屬性特徵	矩陣	缺少缺失值	否
相關任務	分類		

3.2 前置處理

本研究利用 keras 中的 imagedatagenerator 來進行照片的讀取與圖片前處理與增強。首先，先將圖片正規化讓矩陣數值介於 0-1 之間，並利用圖片平移 (shear)、圖片縮放(zoom)以及水平翻轉(horizontal)來增強圖片的訓練集。

```
x_test = changeRGBchannel(y_train_origin)
x_test = np.array(x_test)/255

'''
Imagenerartor with image preproccesing
'''

train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
valid_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_dataframe(
    dataframe = y_train_gen_label,
    directory = train_image_path,
    x_col = 'Name',
    y_col = 'Level',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical')
validation_generator = valid_datagen.flow_from_dataframe(
    dataframe = y_valid_gen_label,
    directory = valid_image_path,
    x_col = 'Name',
    y_col = 'Level',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical')
```

圖 2 資料前處理

3.3 實驗設計

本研究是利用 Anaconda3 Jupyter Notebook(Python version = 3.7)環境進行開發，使用的套件有 keras、pandas、numpy、tensorflow、matplotlib。

本研究採用 keras 所提供的 DenseNet121 的預訓練模型以及 Imagenet 預訓練權重來進行模型的建立。

3.3.1 研究模型架構

DenseNet121 類神經架構包含了五個卷曲神經網路層，第一層為輸入層，圖片輸入的型態 224*224*3，第二層包含了 6 個 dense block，第三層包含了 12 個 dense block，第四層包含 24 個 dense block，第五層為 16 個 dense block，此時的輸出通道為 1024，之後再新增一層隱藏層，並且使用 relu 當作激勵函數，最終的輸出層使用 softmax 作為激勵函數以利於進行圖片分類的預測。

本研究模型所採用的優化器為 adam，為現今大多數模型常用的優化器，主要是可以讓參數更新較穩定，損失函數使用 categorical_crossentropy，因為此次研究主要是針對芒果進行分類，而此 loss 函數會將輸出目標值設定 3 為的向量(芒果分類有三個等級)，並會將表示該類別的向量索引設為 1，其餘皆為 0，以此計算損失值，為此本研究採用此損失函數。

3.3.2 Callbacks 函數

本研究 Callbacks 函數包含了兩種模式，一種為 earlystopping，monitor 設為 val_loss，透過驗證集的 loss 來監控學習的效率，並設定閾值為 0.001；另一種為 ReduceLROnPlateau 為一種調整學習率的函式，factor 為控制學習率的減少程度，公式為： $\text{new_lr} = \text{lr} * \text{factor}$ ，在這邊採用預設值 0.1。

```
def DenseModel(n_classes=3, input_shape=(224,224,3)):
    base_model = DenseNet121(weights='imagenet', include_top=False, input_shape=input_shape)
    x = AveragePooling2D(pool_size=(3,3), name='avg_pool')(base_model.output)
    x = Flatten()(x)
    x = Dense(1024, activation='relu', name='dense_post_pool')(x)
    x = Dropout(0.5)(x)
    output_ = Dense(n_classes, activation='softmax')(x)
    model = Model(inputs=base_model.input, outputs=output_)
    return model

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

early_stop = EarlyStopping(monitor='val_loss', patience=8, verbose=1, min_delta=1e-4)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=4, verbose=1, min_delta=1e-4)
callbacks_list = [early_stop, reduce_lr]

nb_validation_samples = 800
model_history = model.fit_generator(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples // batch_size,
    callbacks=callbacks_list)
```

圖 3 model 訓練程式碼

```
prediction = model.predict(x_test)
```

```
prediction[0]
```

```
array([1.8129771e-05, 1.8282820e-02, 9.8169911e-01], dtype=float32)
```

```
level_arr=[]  
for i in prediction:  
    level = np.argmax(i)  
    if level == 0:  
        level = 'A'  
    elif level == 1:  
        level = 'B'  
    else:  
        level = 'C'  
    level_arr.append(level)  
test_upload['label']=level_arr
```

```
upload_path = './Doc/0614_Dense121.csv'  
test_upload.to_csv(upload_path,index = False,encoding='utf_8_sig')
```

```
model.save_weights('Mongo_weight.h5')
```

圖 4 預測程式碼

3.4 實驗結果

本研究小組利用卷積神經網路及 DenseNet 進行預測對資料集進行分析的結果如下：

上傳者	上傳時間	評估結果	排名
 GeorgePing 0614_Dense121.csv	2020-06-15 16:46:28	0.7575	302/543

圖 5 績效

四、結論

本研究是以 BIIC Lab 所提供之 5600 筆芒果之圖像為訓練資料，並使用卷積神經網路及 DenseNet 去進行分類並針對 1600 預測訓練集做預測，最終我們得出的績效成績約等於 0.7575。

五、參考文獻

DenseNet :

<https://keras.io/api/applications/densenet/#densenet121-function>

<https://zhuanlan.zhihu.com/p/37189203>

ImageDataGenerator :

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator