智慧化企業整合

皮膚癌類型之圖形辨識分析－
以 ISIC 公開資料集為例

指導教授：邱銘傳 教授

學生：109034532 張郁杰

中 華 民 國 １１０ 年 一 月 七 日

目錄

圖目錄

表目錄

1 研究背景與目的

皮膚癌是人類最常見的惡性腫瘤，主要透過視覺檢查，從臨床篩檢開始，至皮膚鏡分析與活體組織切片等。現階段常見的皮膚癌種類共可分為以下七種，依序為黑色素細胞痣、黑色素瘤、良性角化病樣病變、基底細胞癌、光化性角化病、血管病變與皮膚纖維瘤，其示意圖如圖 1 所示。因皮膚損傷的細微變化，使用圖形辨識進行皮膚癌種類分辨是一項困難的任務！此研究使用 ISIC 公開資料集，用於深度學習與人類專家進行比較。

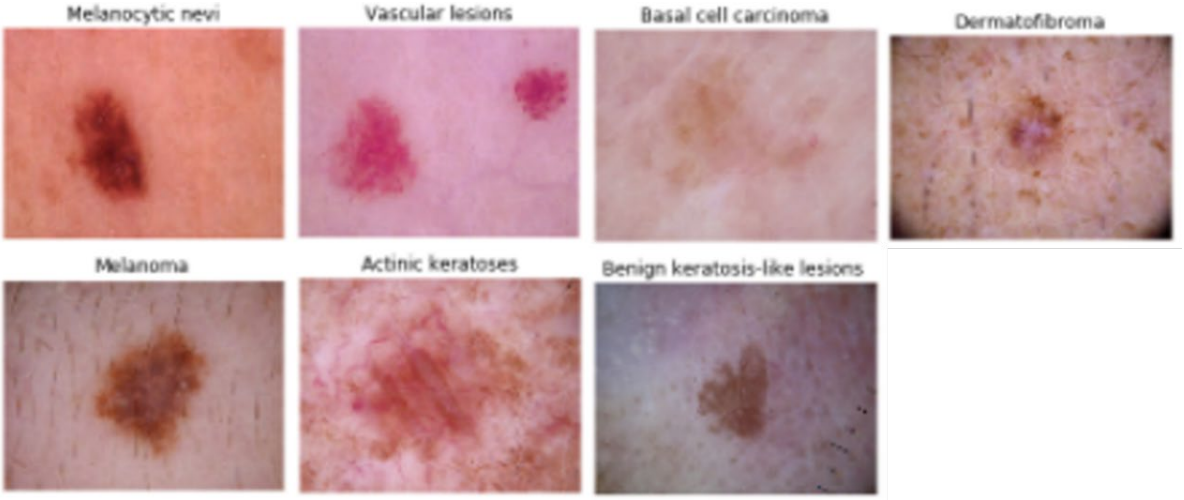

圖 1　皮膚癌種類示意圖

以 5W1H 對本研究進行問題分析，並了解實際研究之求且方法與目標，其分析如下圖 2 所示。

圖 2　5W1H 問題分析

2 公開資料集介紹

本研究使用國際皮膚影像合作學會(ISIC)提供之公開資料集進行研究，主要可分為圖形資料與數值資料，數值資料示意如圖 3 所示，圖形資料如圖 4 所示。數值資料標籤共有記錄每一筆圖形資料之圖檔編號、年齡、性別、皮膚癌位置、皮膚癌種類、檢測方式等，當中針對 7 種皮膚癌種類新增編號欄位其數值為 0~6。圖形資料則共計 10,015 筆。

	lesion_id	image_id	dx	dx_type	age	sex	localization	path	cell_type	cell_type_idx
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2

圖 3　皮膚癌數值資料示意圖

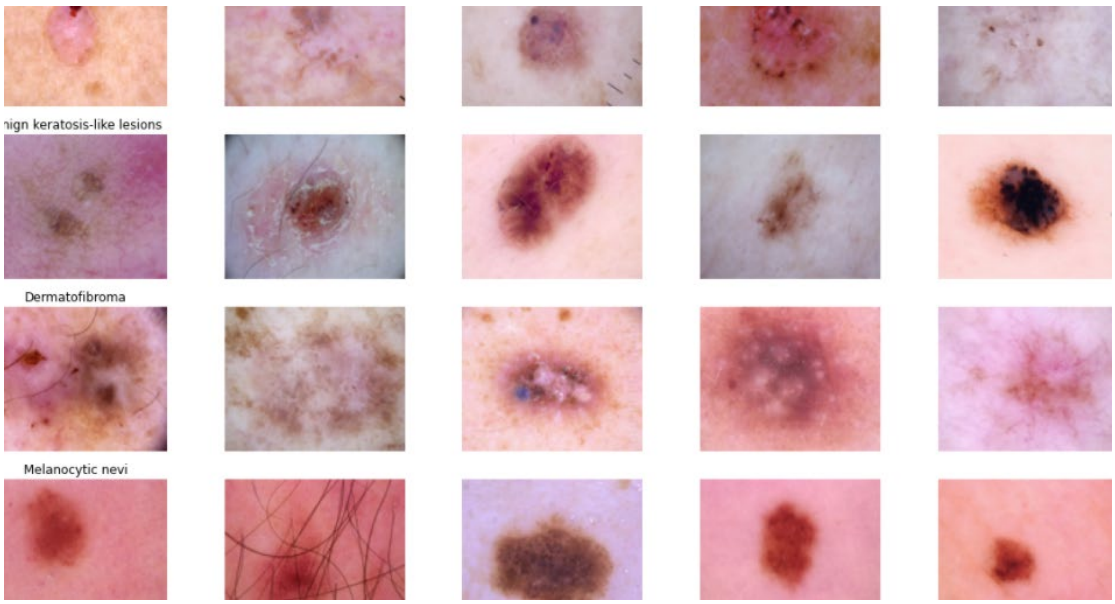

圖 4 皮膚癌圖形資料示意圖

3 研究方法

圖 5 為本次實驗的研究架構圖，首先會進行資料讀取以及資料異常之處理，當中僅於年齡欄位出現 57 筆空值資料，以年齡之平均值進行補值，而後進行資

料探索分析，透過簡易分析了解現行皮膚癌於各年齡層、性別、檢測方式之分佈等。完成以上程序在繼續對皮膚癌種類資料進行 one-hot encoding，以及資料標準化，後續為了避免產生過擬合之機會因子進行資料增強。

以上主要為資料處理階段，後續則進行本研究之模型建立，先建構 CNN 模型，並進行超參數調整以找出針對本研究之最佳模型設置。最終進行評估分析並針對本研究給予結論建議。

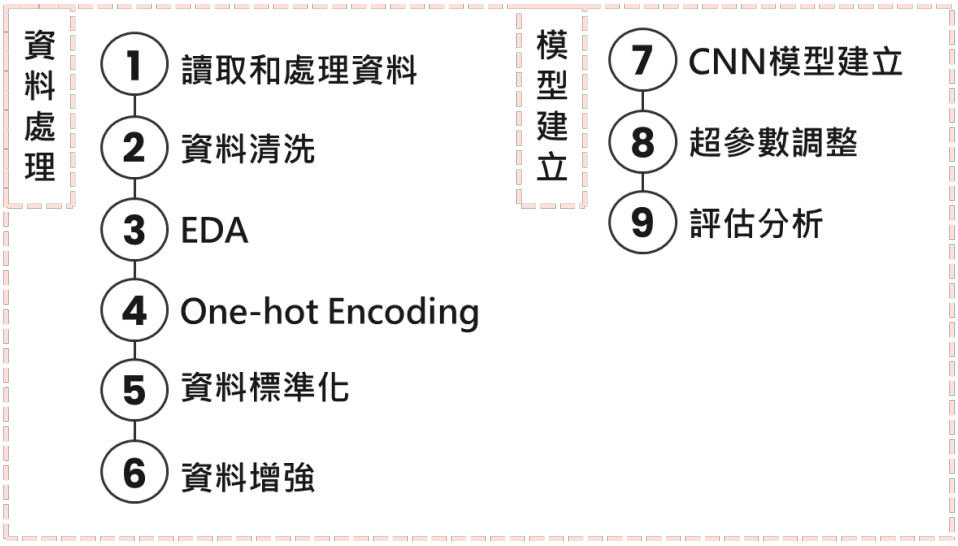

圖 5 研究架構圖

4 研究流程

本研究以 Python 進行實作，先將欲使用之套件匯入其中，如圖 6 所示。

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
from glob import glob
import seaborn as sns
from PIL import Image
np.random.seed(4321)
from sklearn.preprocessing import label_binarize
from sklearn.metrics import confusion_matrix
import itertools

import tensorflow as tf

import keras
from keras.utils.np_utils import to_categorical # used for converting labels to one-hot-encoding
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from keras import backend as K
import itertools
from keras.layers.normalization import BatchNormalization
from keras.utils.np_utils import to_categorical # convert to one-hot-encoding

from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau
from sklearn.model_selection import train_test_split
```

圖 6 套件匯入程式碼示意圖

4.1 資料讀取和處理

接著將公開資料集給予之數值與圖形資料進行整併，整併後便能將數值與圖形資料結合，程式碼如下圖 7 所示。

```
base_skin_dir = os.path.join('', 'dataverse_files')

# Merging images from both folders HAM10000_images_part1.zip and HAM10000_images_part2.zip into one dictionary

imageid_path_dict = {os.path.splitext(os.path.basename(x))[0]: x
                     for x in glob(os.path.join(base_skin_dir, '*', '*.jpg'))}

# This dictionary is useful for displaying more human-friendly labels later on

lesion_type_dict = {
    'nv': 'Melanocytic nevi',
    'mel': 'Melanoma',
    'bkl': 'Benign keratosis-like lesions ',
    'bcc': 'Basal cell carcinoma',
    'akiec': 'Actinic keratoses',
    'vasc': 'Vascular lesions',
    'df': 'Dermatofibroma'
}
len(imageid_path_dict)
```

```
skin_df = pd.read_csv(os.path.join(base_skin_dir, 'HAM10000_metadata.csv'))

# Creating New Columns for better readability

skin_df['path'] = skin_df['image_id'].map(imageid_path_dict.get)
skin_df['cell_type'] = skin_df['dx'].map(lesion_type_dict.get)
skin_df['cell_type_idx'] = pd.Categorical(skin_df['cell_type']).codes
```

圖 7 資料匯入與合併程式碼示意圖

4.2 資料清洗

完成資料整併以後，欲進行資料清洗，當中發現共計 57 筆空值資料出現於年齡類別，因此將其以平均值補值，其結果如下圖 8 所示

```
skin_df.isnull().sum()
```

```
lesion_id         0
image_id          0
dx                0
dx_type           0
age              57
sex               0
localization      0
path              0
cell_type         0
cell_type_idx     0
dtype: int64
```

```
skin_df['age'].fillna((skin_df['age'].mean()), inplace=True)
```

```
skin_df.isnull().sum()
```

```
lesion_id        0
image_id         0
dx               0
dx_type          0
age              0
sex              0
localization     0
path             0
cell_type        0
cell_type_idx    0
dtype: int64
```

圖 8 資料空值處理示意圖

4.3 EDA (Exploratory Data Analysis, EDA)

完成前述步驟，即可進行探索資料分析 ，藉由此了解整理資料分佈架構與特性。下圖 9 為 7 種皮膚癌類別之資料分佈，由圖可發現現行皮膚癌患者最多數為黑色素細胞痣，其餘種類人數較少一些：圖 10 為檢測種備分佈圖，主要可觀察現在皮膚癌判斷仍多以組織病理學進行評斷 (活體切片等)，其次為長期追蹤調查之皮膚鏡檢測。

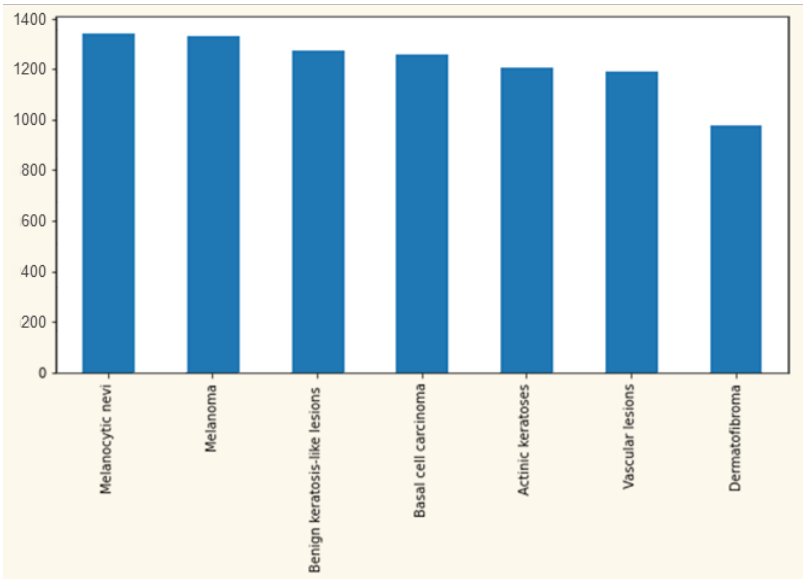

圖 9　皮膚癌種類分佈圖

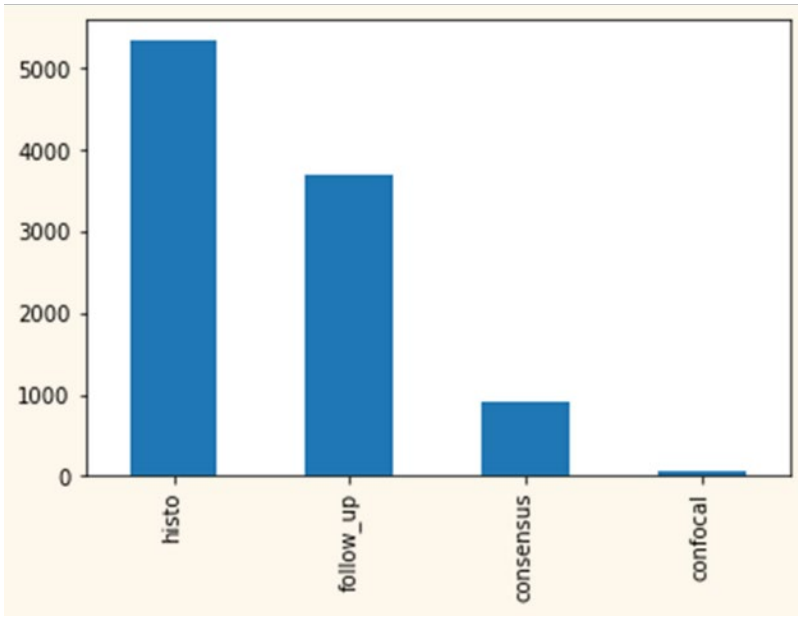

圖 10 皮膚癌檢測方式分佈圖

下圖 11 為皮膚癌位置分佈圖與圖 12 之皮膚癌患者之年齡分佈，皮膚癌患者其症狀位置多分佈於背部、下肢；年齡部分則屬中老年患者居多約介於 30~60 歲之間。

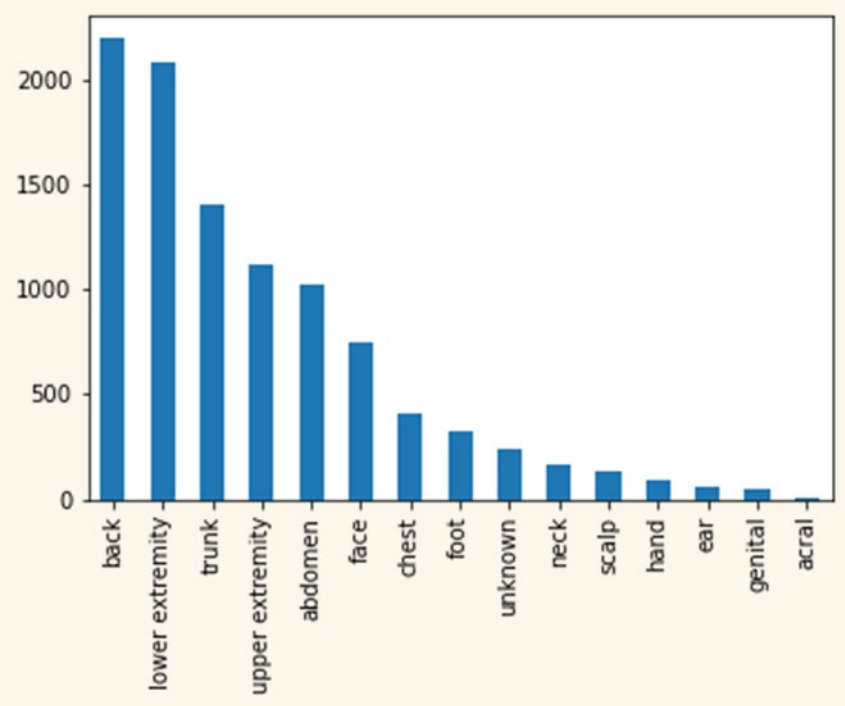

圖 11　皮膚癌症狀位置分佈圖

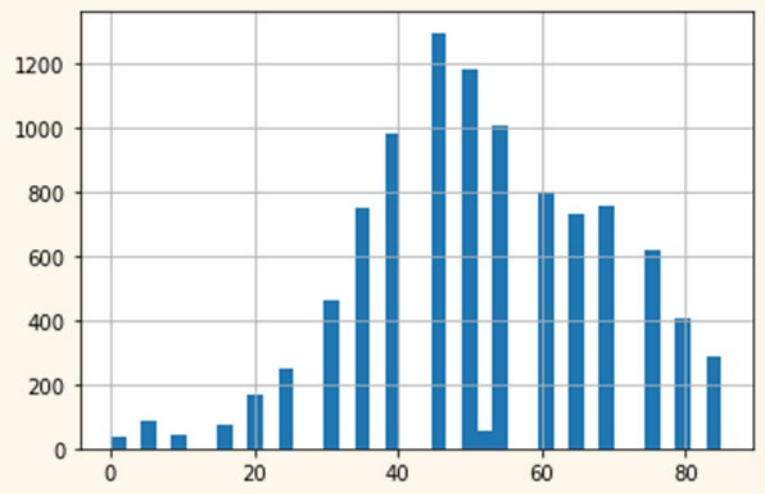

圖 12 皮膚癌年齡分佈圖

下圖 13~14 為皮膚癌患者性別分佈與各年齡層所受之皮膚癌種類分佈，可觀察於性別部分男性較女性高出一些，而於 20 歲以下幾乎不會出現種類 0、1、3、5 之症狀，依序為黑色素細胞痣、良性角化病樣病變、基底細胞癌與皮膚纖維瘤。

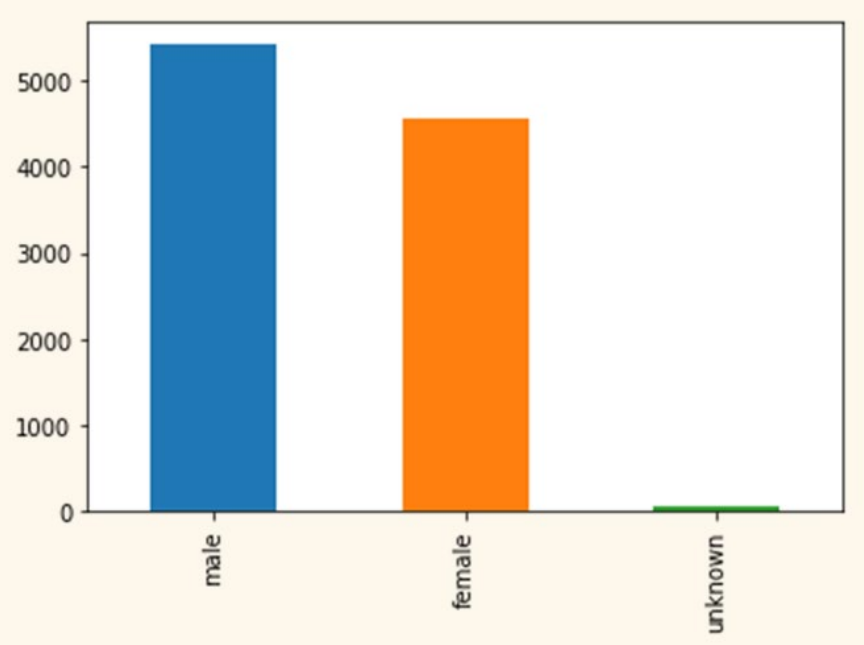

圖 13　皮膚癌性別分佈圖

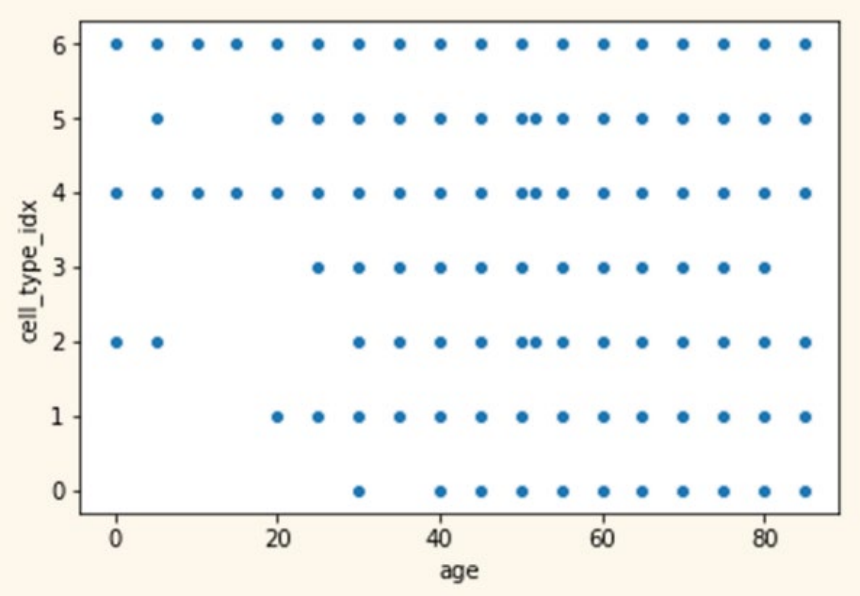

圖 14　皮膚癌各年齡層患症種類分佈圖

4.4 資料分割

本研究將資料以 80:20 比例分割為訓練資料與測試資料，當中，再以 90:10 比例將訓練資料分割為訓練資料與驗證資料。資料分割程式碼如下圖 15。

```
x_train_o, x_test_o, y_train_o, y_test_o = train_test_split(features, target, test_size=0.20,random_state=666)

x_train, x_validate, y_train, y_validate = train_test_split(x_train, y_train, test_size = 0.1, random_state = 999)
```

圖 15　資料分割程式碼示意圖

4.5 One-hot Encoding

為了讓電腦在進行資料分析時方便處理，將癌症種類之資料進行 One-hot Encoding，其程式碼如下圖 16 所示。

```
# Perform one-hot encoding on the labels
y_train = to_categorical(y_train_o, num_classes = 7)
y_test = to_categorical(y_test_o, num_classes = 7)
```

圖 16　One-hot-Encoding 示意圖

4.6 資料正規化

此階段將輸入資料進行正規化處理，縮短資料處理時各資料之間的差異。其程式碼如下圖 17 所示

```
x_train = np.asarray(x_train_o['image'].tolist())
x_test = np.asarray(x_test_o['image'].tolist())

x_train_mean = np.mean(x_train)
x_train_std = np.std(x_train)

x_test_mean = np.mean(x_test)
x_test_std = np.std(x_test)

x_train = (x_train - x_train_mean)/x_train_std
x_test = (x_test - x_test_mean)/x_test_std
```

圖 17 資料正規化圖

4.7 資料增強

後續為了避免於訓練過程出現過擬和現象，於此進行資料增強，常見的處理手法包含灰度，水平翻轉，垂直翻轉，隨機裁剪，顏色抖動，平移，旋轉等。

本研究選擇將圖像隨機旋轉 10 度、隨機放大縮小 10%、將圖像隨機水平、垂直移動 10%長度。資料增強示意如圖 18 所示。

```
# With data augmentation to prevent overfitting

datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.10,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.10,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True)  # randomly flip images

datagen.fit(x_train)
```

圖 18 資料增強圖

4.8 CNN 模型建立

本研究選用 CNN 模型進行深度學習分析，其架構如下：進行 2 次神經元數為 32 之卷積層、進行一次池化層、經過比率為 0.16 之 Dropout；再進行 2 次神經元數為 32 之卷積層、進行一次池化層、經過比率為 0.20 之 Dropout；再進行 2 次神經元數為 64 之卷積層、進行一次池化層、經過比率為 0.25 之 Dropout。然後進行扁平層將神經元拉直，接著進行一次神經元數為 256 與 128 之全連接層、0.4 之 Dropout 比率。CNN 預設模型架構如圖 19。

```
input_shape = (125, 100, 3)
num_classes = 7

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding = 'Same',input_shape=input_shape))
model.add(Conv2D(32,kernel_size=(3, 3), activation='relu',padding = 'Same',))
model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.16))

model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding = 'Same'))
model.add(Conv2D(32,kernel_size=(3, 3), activation='relu',padding = 'Same',))
model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.20))

model.add(Conv2D(64, (3, 3), activation='relu',padding = 'same'))
model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(num_classes, activation='softmax'))
model.summary()
```

圖 19　CNN 模型架構圖

最終，選擇 Adam 做為優化器並以 Categorical_crossentropy 為 Loss Function，且於學習率部分新增一退火方式使每次執行過程於準確率於顯著提升時自動下降學習率增加收斂速度。上述設置如下圖 20 所示。

```
# Define the optimizer
optimizer = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, decay=0.0, amsgrad=False)
```

```
# Compile the model
model.compile(optimizer = optimizer , loss = "categorical_crossentropy", metrics=["accuracy"])
```

```
# Set a learning rate annealer
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy',
                                            patience=4,
                                            verbose=1,
                                            factor=0.5,
                                            min_lr=0.00001)
learning_rate_reduction
```

圖 20　模型設置圖

4.9 超參數調整

接著進行超參數調整，首先調整 Activation Function，本研究比較三種 Activation Function，如 softmax、ReLU、Sigmoid。可以得知 Activation Function 為 softmax 的時候模型表現最好，因此保留其參數設定。表 1 為 Activation Function 評估比較表。

表 1　Activation Function 評估比較表

	Training	Validation	Testing
Softmax	0.7639	0.7319	0.7224
ReLU	0.6692	0.6633	0.6724
Sigmoid	0.7470	0.7157	0.7079

接著進行 Batch size 調整，本研究比較三種 Batch size，依序為 10、50、100。可以得知 Batch size 為 100 的時候模型表現最好，因此保留其參數設定。表 2 為 Batch size 評估比較表。

表 2　Batch size 評估比較表

	Training	Validation	Testing
10	0.7639	0.7319	0.7224
50	0.7973	0.7543	0.7448
100	0.8110	0.7618	0.7619

接著進行 Neuron 調整，本研究比較不同之 Neuron 排序，依序為{32, 32, 64}、{32, 64, 64}、{64, 64, 64}、{64, 64, 32}、{64, 32, 32}、{32, 32, 32}。可以得知 Neuron 於 3 大卷積層，排序為{32, 32, 64}的時候模型表現最好，因此保留其參數設定。表 3 為 Neuron 評估比較表。

表 3　Neuron 評估比較表

	Training	Validation	Testing
32, 32, 64	0.8110	0.7618	0.7619
32, 64, 64	0.8003	0.7656	0.7554
64, 64, 64	0.7882	0.7606	0.7564
64, 64, 32	0.7825	0.7593	0.7564
64, 32, 32	0.7786	0.7556	0.7613
32, 32, 32	0.7690	0.7592	0.7607

從超參數調整的過程中，可以發現模型的訓練資料、驗證資料與測試資料準確度已經從原始設定的 72.24%提升至 76.19%。另外以混淆矩陣的形式也可以看出模型預測的大致上為正確。圖 21 為模型預測結果混淆矩陣。

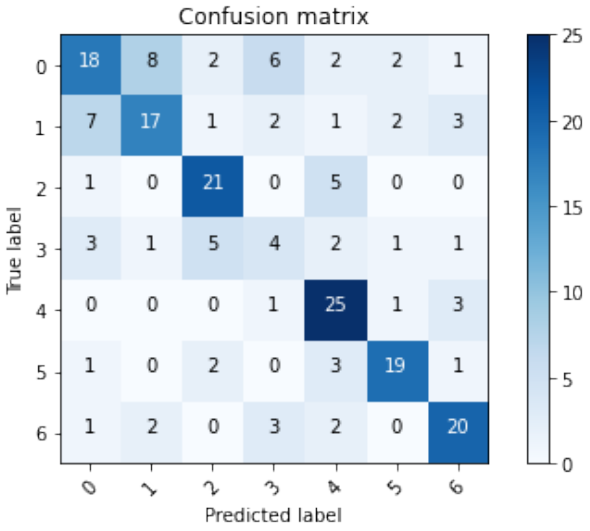

圖 21　模型結果預測混淆矩陣

5 結論

由前述之過程與結果分析，應用深度學習於皮膚癌症狀辨別仍有進步空間，各類別識別值錯誤率如下圖 22 所示，可見於 Label3(基底細胞癌)之症狀其幾乎無法有效地正確識別，可能因皮膚鏡之影像過於類似，仍需透過更深入之病理學進行刪減觀測內部細胞組織才能進行識別；Label4(黑色素細胞痣)為現行多數人較易獲得之皮膚癌症狀，且其症狀明顯顏色差異大較易識別，若此模型識別能持續改善，亦能有效較低醫療上因病理學檢查(活體切片)等造成的時間與成本浪費，本研究仍有參考價值。未來發展層面，因現行皆以皮膚鏡所蒐集之圖片進行檢測，若檢測能藉由其他方式，亦能產生額外機會提升模型識別率。

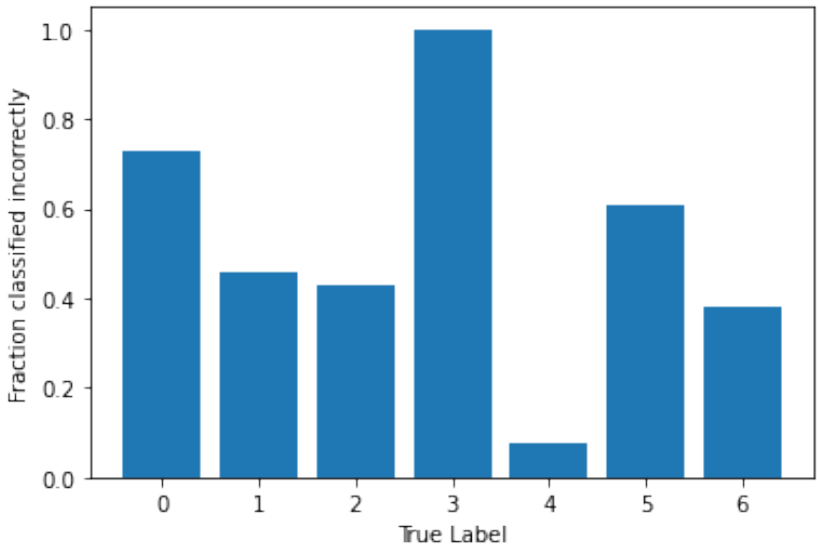

圖 22　模型結果錯誤率分佈圖