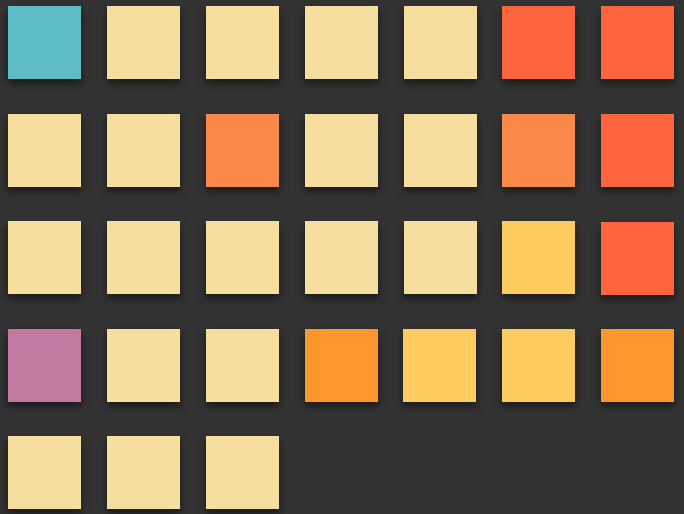


皮膚癌類型之圖形辨識分析 -以 ISIC 公開資料集為例

109034532_Group9_張郁杰

研究目的

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。

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。

5W1H

When

皮膚癌症狀顯現
前中後期

Who

皮膚癌患者

Where

於醫院檢測環境

What

皮膚出現異常症狀需
要進行檢測確認

Why

篩檢過程複雜，皮膚
細微損傷難以辨識

How

以深度學習進行圖像
識別癌症種類

研究流程

資料處理

1. 讀取和處理資料
2. 資料清洗
3. EDA
4. One-hot Encoding
5. 資料標準化
6. 資料增強

模型建立

7. CNN模型建立
8. 超參數調整
9. 評估分析

資料介紹/處理

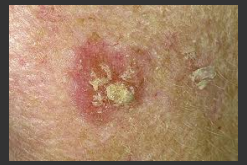
Melanocytic nevi
黑色素細胞痣

Melanoma
黑色素瘤

Dermatofibroma
良性角化病樣病變

Basal cell carcinoma
基底細胞癌

Actinic keratoses
光化性角化病



Benign keratosis - like lesions
血管病變

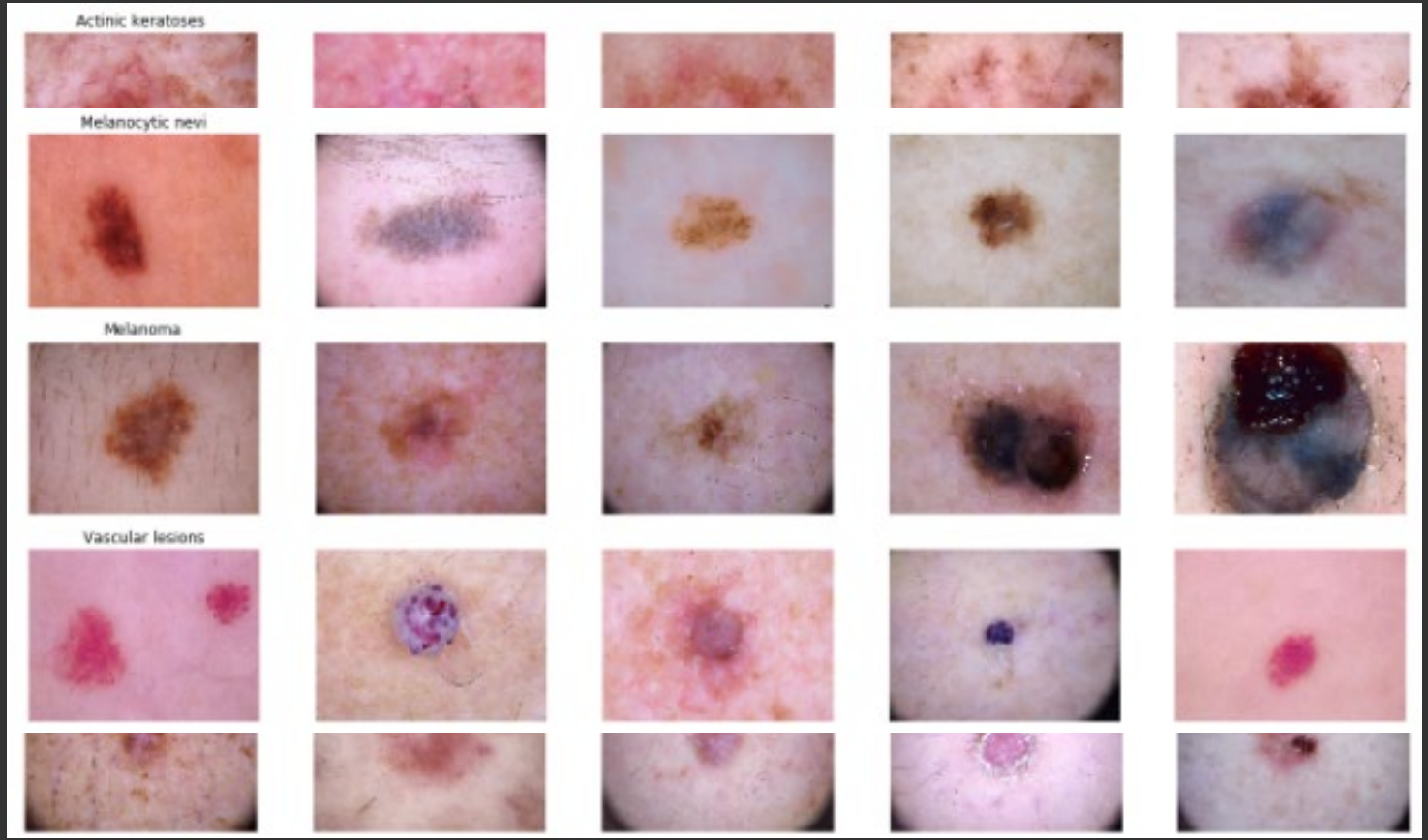
Vascular lesions
皮膚纖維瘤

資料介紹

- 共計 10,015 筆圖形與數值資料
- 共計 57 筆空值於年齡資料，以平均值補值
- 新增 Cell_type_idx 將 7 種皮膚癌類型編號為 0 ~ 6
- 因圖片維度過大 (450*600*3)，縮小為 (125*100*3)

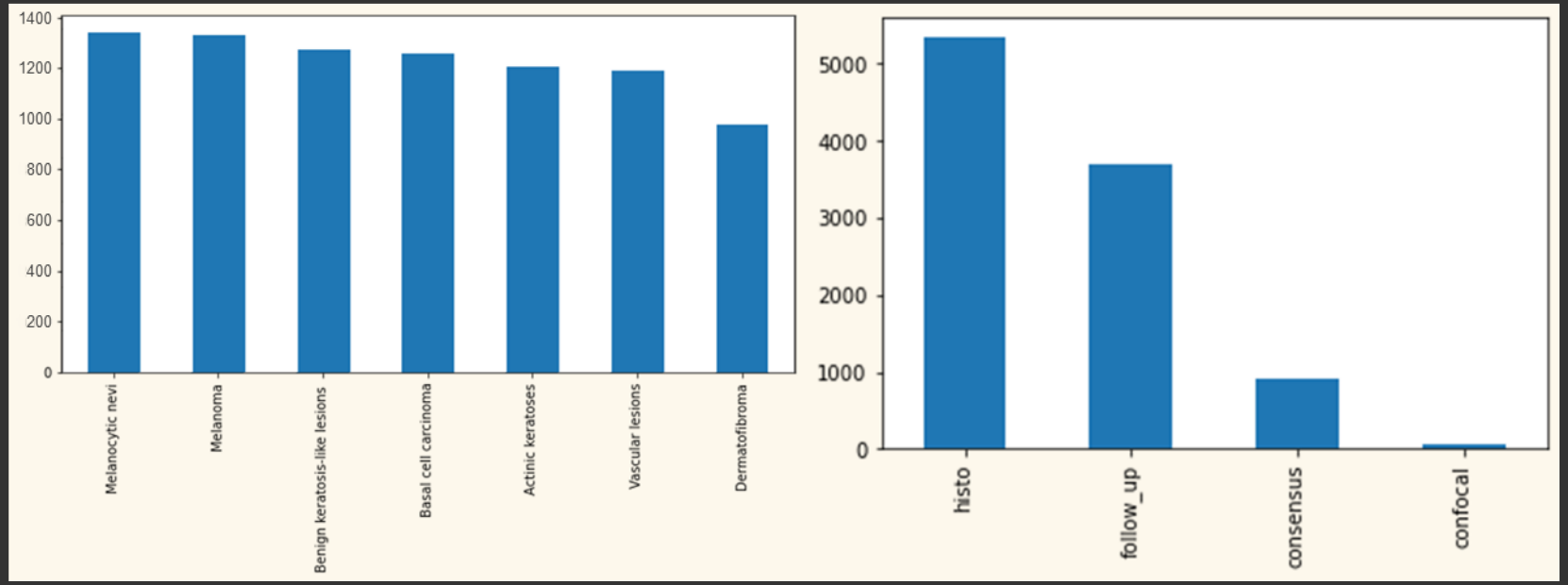
	lesion_id	image_id	dx	dx_type	age	sex	localization	path	cell_type	cell_type_idx
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...	Benign keratosis-like lesions	2
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	C:/Users/Jay/Desktop\dataverse_files\HAM10000_...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

資料介紹


Actinic keratoses
Melanocytic nevi
Melanoma
Vascular lesions

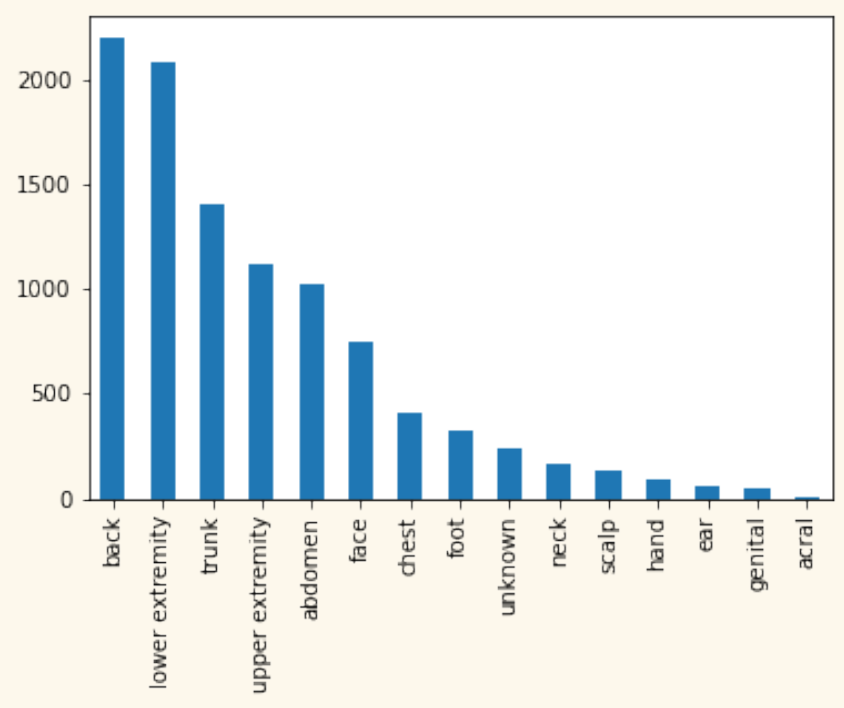
EDA

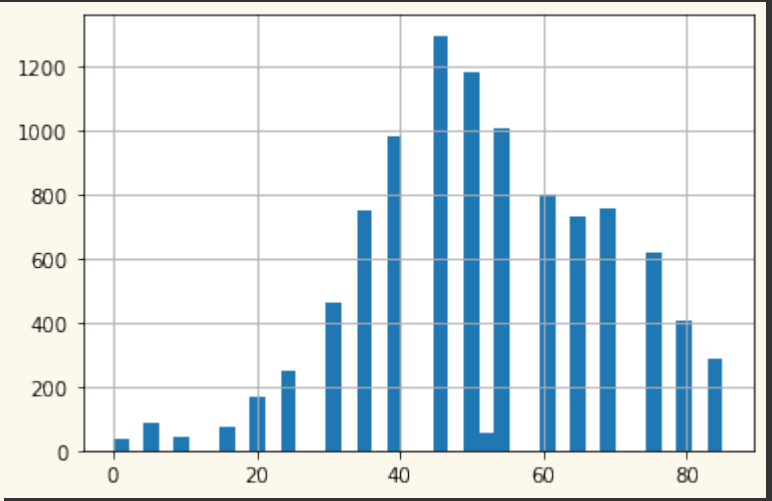
- 皮膚癌種類 & 識別分佈



EDA

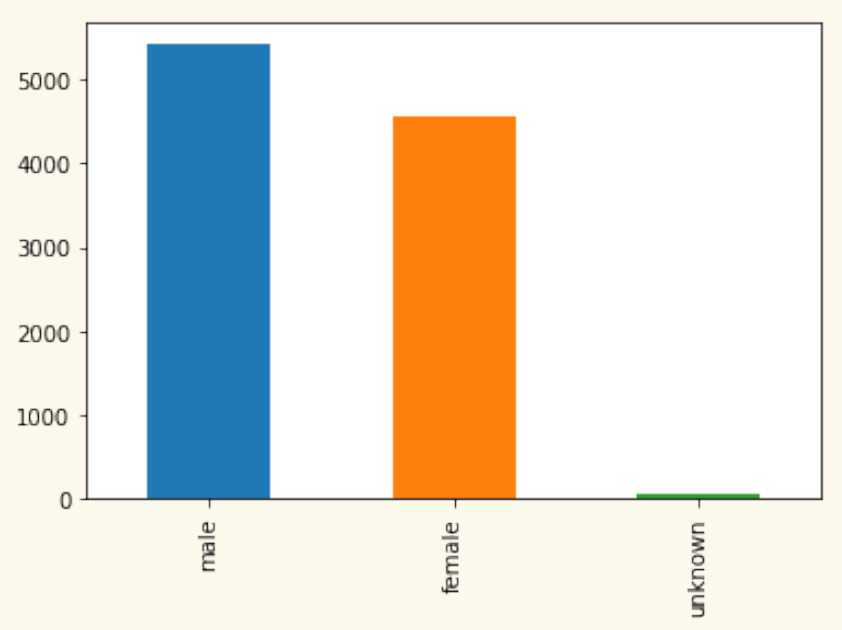
- 皮膚癌位置 & 年齡分佈

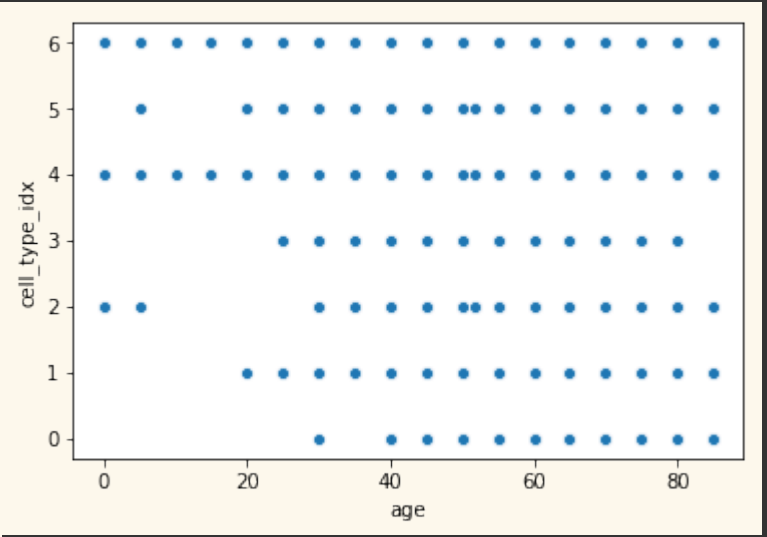




EDA

- 皮膚癌性別 & 年齡種類分佈





資料前處理

- 以 80:20 之比例將資料分割為訓練資料與測試資料
- 資料正規化
- one-hot encoding

```
x_train_o, x_test_o, y_train_o, y_test_o = train_test_split(features, target, test_size=0.20,random_state=1234)
```

```
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
```

```
# 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)
```

資料前處理

- 以 90:10 之比例再將訓練資料分割出驗證資料
- 確認各資料集圖形維度

```
x_train, x_validate, y_train, y_validate = train_test_split(x_train, y_train, test_size = 0.1, random_state = 2)
```

```
# Reshape image in 3 dimensions (height = 75px, width = 100px , canal = 3)
x_train = x_train.reshape(x_train.shape[0], *(75, 100, 3))
x_test = x_test.reshape(x_test.shape[0], *(75, 100, 3))
x_validate = x_validate.reshape(x_validate.shape[0], *(75, 100, 3))
```

Model Building CNN

OPTIMIZER: Adam

LOSS: Categorical

LEARNING RATE: 0.001

EPOCHS: 50

BATCH SIZE: 10

DATA AUGMENTATION

```
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()
```

Model Building CNN

OPTIMIZER: Adam

LOSS: Categorical

LEARNING RATE: 0.001

EPOCHS: 50

BATCH SIZE: 10

DATA AUGMENTATION

```
# 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_acc',
                                            patience=3,
                                            verbose=1,
                                            factor=0.5,
                                            min_lr=0.00001)
```

```
# 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=0.12,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.12,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True)  # randomly flip images

datagen.fit(x_train)
```

超參數優化

- 調整 Activation Function

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

超參數優化

- 調整 Batch Size

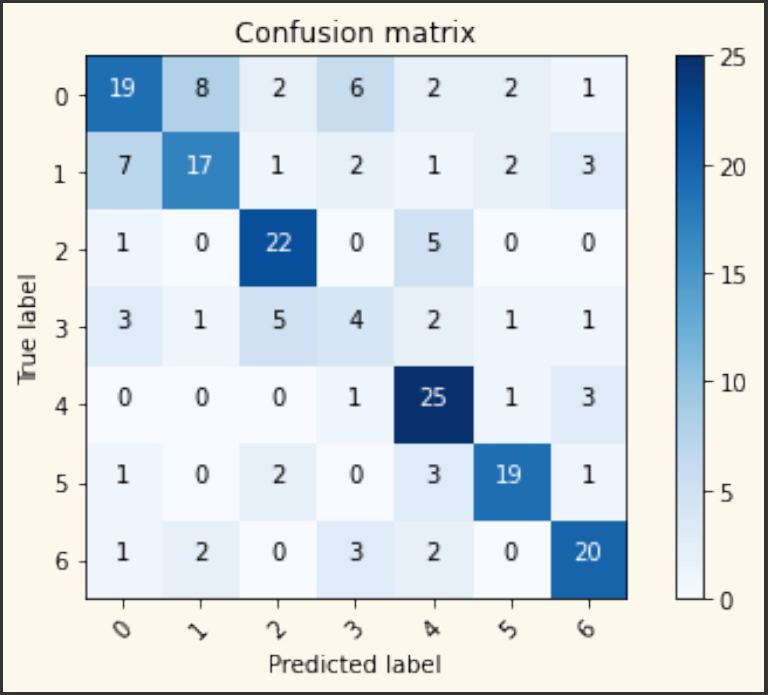
	Training	Validation	Testing
10	0.7639	0.7319	0.7224
50	0.7973	0.7543	0.7448
100	0.8110	0.7618	0.7619

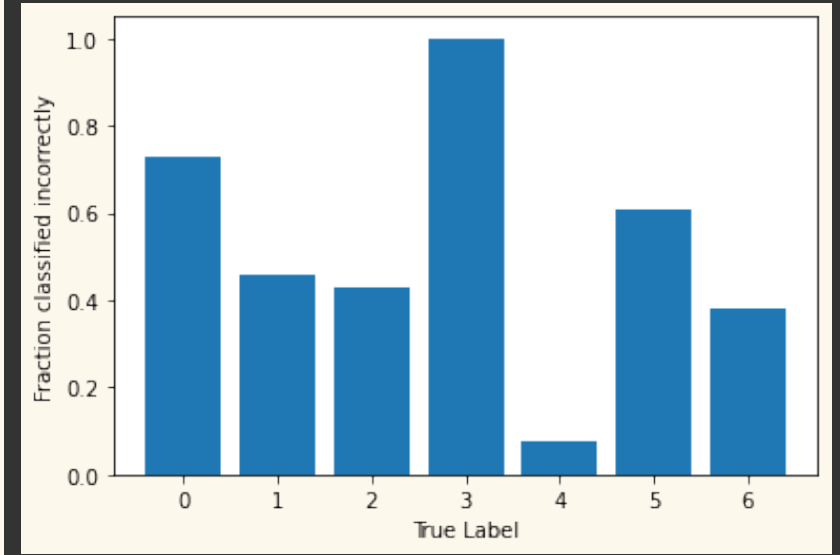
超參數優化

- 調整 Neuron

	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

評估分析


Confusion matrix
True label
Predicted label
0 1 2 3 4 5 6
19 8 2 6 2 2 1
7 17 1 2 1 2 3
1 0 22 0 5 0 0
3 1 5 4 2 1 1
0 0 0 1 25 1 3
1 0 2 0 3 19 1
1 2 0 3 2 0 20
25 20 15 10 5 0


Fraction classified incorrectly
True Label
0.0 0.2 0.4 0.6 0.8 1.0
0 1 2 3 4 5 6

結論

研究結果

- Label3(基底細胞癌)之症狀其幾乎無法有效地正確識別
- 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

未來展望

- 持續改善，減少病理學檢查(活體切片)等造成的時間與成本浪費
- 發展創新檢測模式新增識別機會，提升模型識別率

THANKS FOR LISTENING