

腦部腫瘤MRI影像之 分類模型建立

Brain Tumor MRI images Classification

指導教授：邱銘傳 教授

組別：第二組

學生：110034402 黃彥蓉

110034403 祝煜恆

110034552 陳庚鼎

110034587 曾琮祐



CONTENTS

01

背景介紹

03

模型訓練與績效

02

研究方法

04

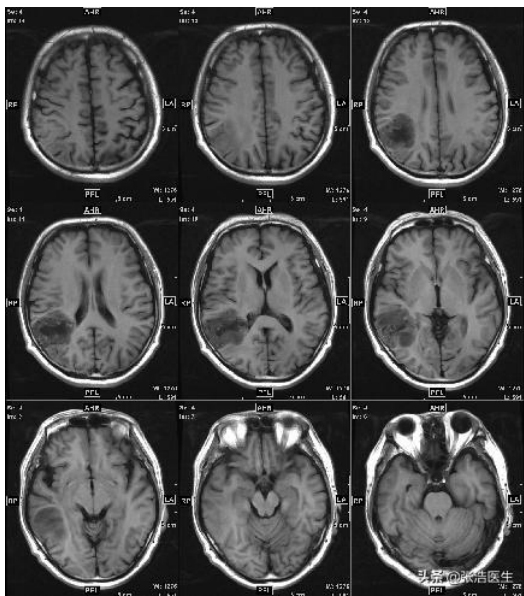
結果與未來展望

01

背景介紹



背景介紹



背景與動機

患有腦癌的人5年生存率: 男性 ~ 34% ; 女性 ~ 36%

腦腫瘤及其特性的複雜程度，手動檢查可能容易出錯

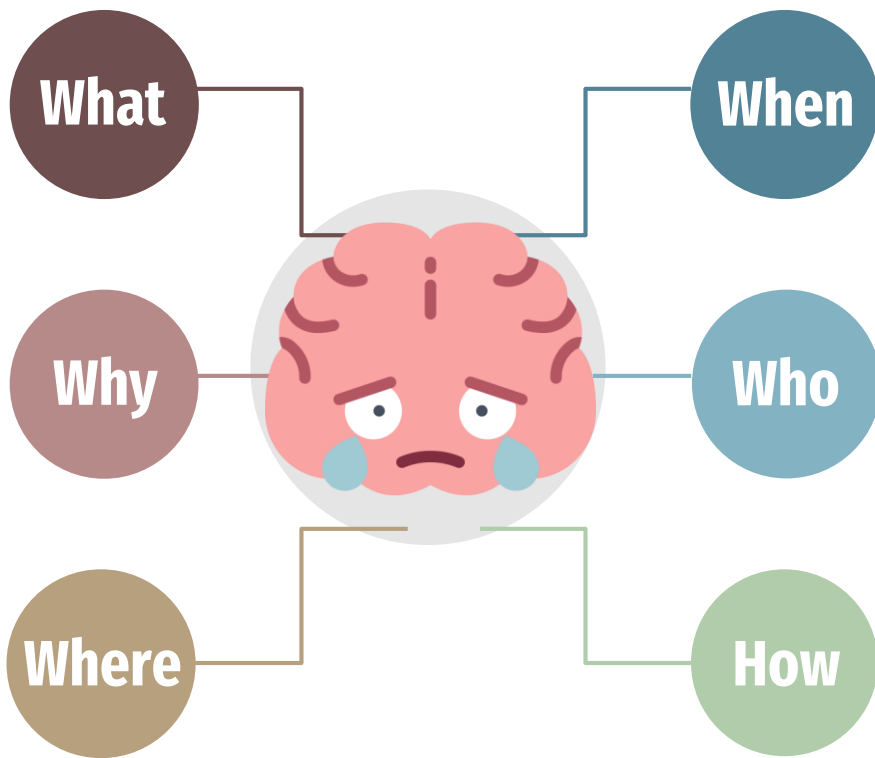
加上MRI非常依賴醫生的經驗，有機會因經驗不足而辨識錯誤

從MRI判斷腫瘤報告變得非常具有挑戰性和耗時

運用卷積神經網路(CNN)，建立腦部腫瘤MRI影像的分類模型，藉由模型辨識MRI影像中的腦腫瘤之類別，讓醫生作為診斷參考，減輕事務繁雜的醫生的負擔，並提升MRI掃描生成報告之速度及準確率，期望能讓病患及其家屬盡早了解身體狀況

研究目的

5W1H



MRI人工閱片複雜耗時
需依賴醫生的經驗
經驗不足而導致辨識錯誤

提供醫生作診斷參考根據
減少閱片時間及
降低人工辨識錯誤機會

醫院、放射科、
神經外科

病患有腦腫瘤疑慮需進行**MRI**影像分析時

病患、
病患家屬、
神經外科醫生

卷積神經網絡
(**CNN**)模型訓練

02

研究方法



CNN 網路架構名稱
EfficientNetB0
EfficientNetB1
EfficientNetB2
EfficientNetB3
EfficientNetB4
EfficientNetB5
EfficientNetB6
EfficientNetB7
DenseNet121
DenseNet169
DenseNet201
InceptionResNetV2
InceptionV3

CNN 網路架構名稱
VGG16
VGG19
Xception
MobileNet
MobileNetV2
MobileNetV3Large
MobileNetV3Small
NASNetMobile
ResNet50
ResNet50V2
ResNet101
ResNet101V2
ResNet152

選出 EfficientNetB5 、 DenseNet121 、 InceptionV3 這三個模型作進一步研究

- 訓練速度快和穩定
- 相對少的epoch數達到高Accuracy
- 較新且知名的模型

EfficientNetB5 (1/2)

- **Google**提出的新基準網路
- 不同的深度/寬度/解析度會影響準確度，但需要平衡調優
- 應用固定的縮放係數均勻地縮放：
 - Depth** (網絡層數)
 - Width** (通道數)
 - Resolution** (圖片大小)

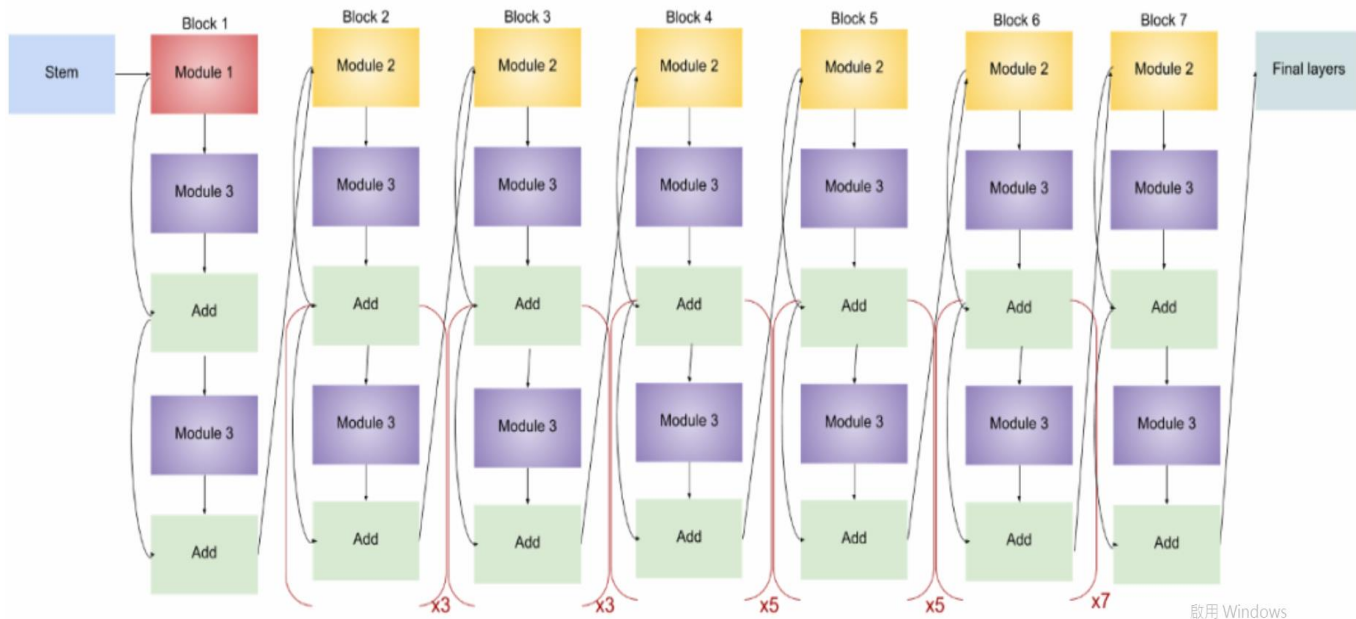
優點：

- 固定資源限制下達到更好的精確度

Table 1. EfficientNet-B0 baseline network – Each row describes a stage i with \hat{L}_i layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels \hat{C}_i . Notations are adopted from equation 2.

Stage i	Operator \hat{F}_i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

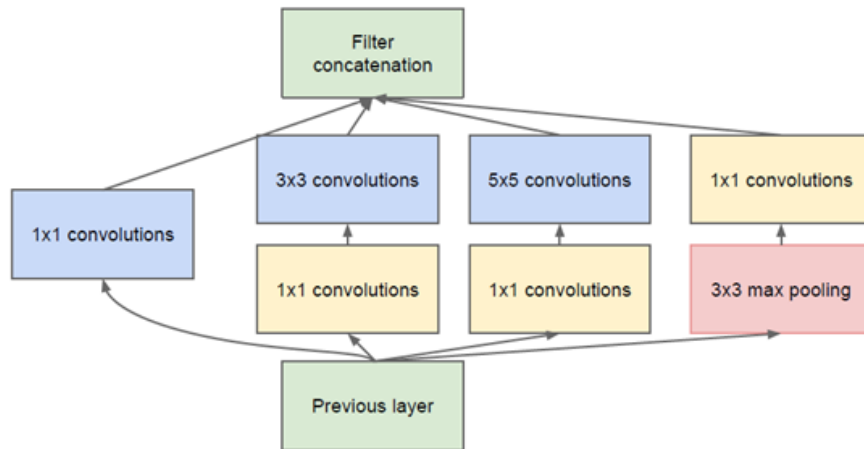
EfficientNetB5 (2/2)



Inception V3 (1/3)

- GoogLeNet – Inception V1
- 使用了 **Inception** 模塊，對輸入圖像並行執行多個操作，將結果拼接成特徵圖
- 全連接層替換成全局平均池
- 中間層使用輔助分類節點

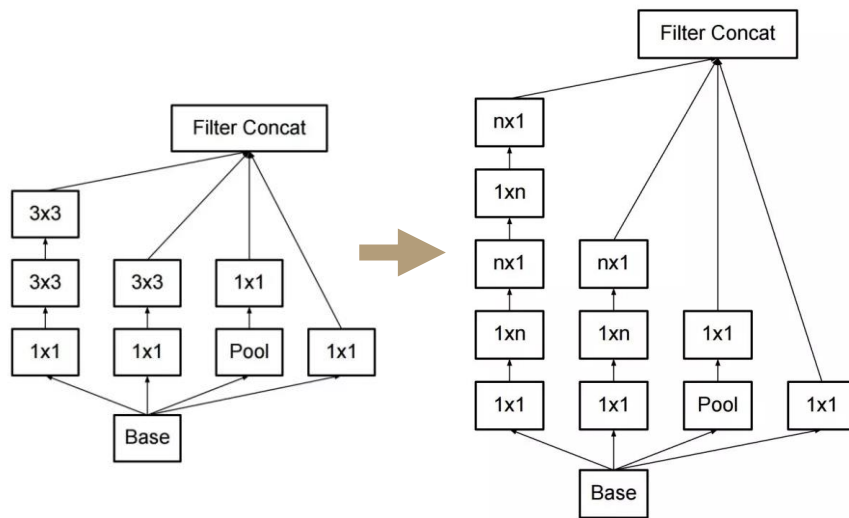
優點：
- 減輕梯度消失的問題
- 大幅地減少了參數計算量



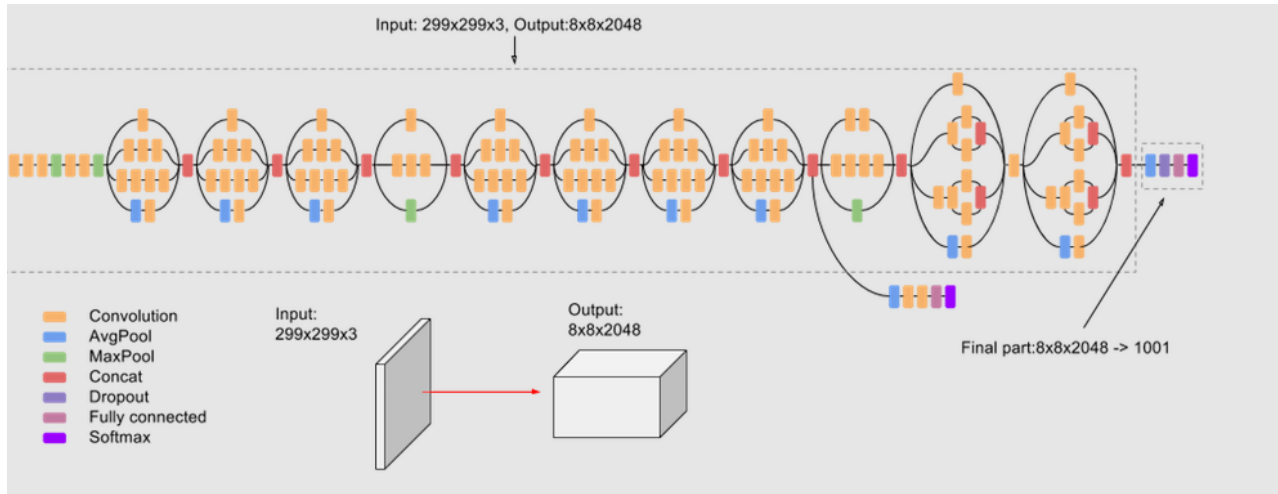
Inception V3 (2/3)

- 將大濾波器分解成數個小濾波器
- 優化輔助分類節點 $2 \Rightarrow 1$
- 更複雜的池化層
- **Label Smoothing Regularization(LSR)**

優點：
- 加速計算速度
- 提高訓練精準度
- 降低模型過擬合程度



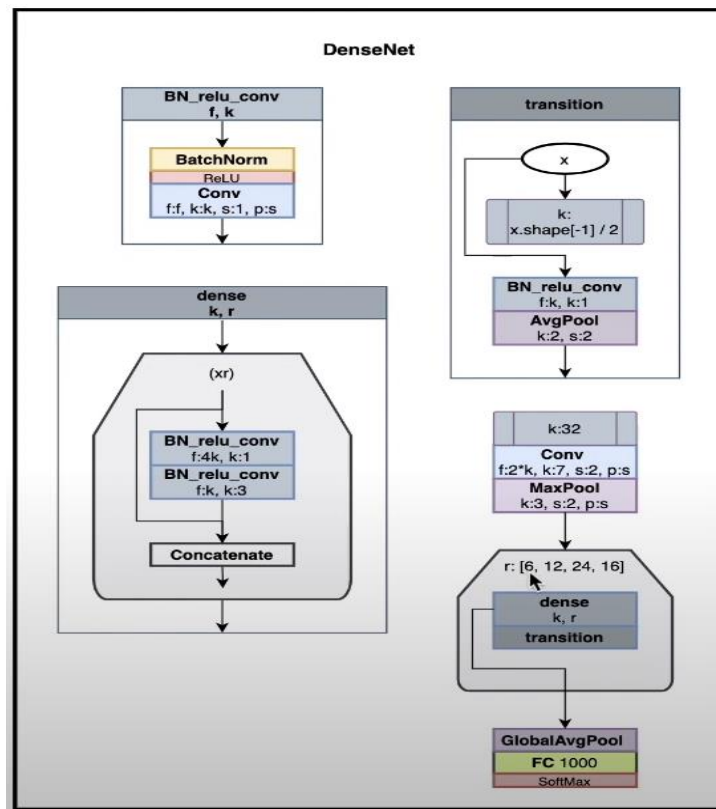
Inception V3 (3/3)



DenseNet121 (1/2)

- 提出了一個更激進的密集連接機制
- 通過特徵在 **channel** 上的連接來實現特徵重用(**feature reuse**)

優點：
- 加強了特徵的利用
- 減輕梯度消失的問題
- 大幅地減少了參數計算量



DenseNet121 (2/2)

Layers	Output Size	DenseNet-121($k = 32$)	DenseNet-169($k = 32$)	DenseNet-201($k = 32$)	DenseNet-161($k = 48$)
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Transfer Learning

- 遷移學習(Transfer Learning)是一種機器學習方法
- 把一個領域(源領域)的知識，遷移到另外一個領域(目標領域)，使得目標領域能夠取得更好的學習效果。
- 當目標域數據量較小或者數據的標籤很難獲取，可以通過數據量充足或者容易獲取標籤且和該任務相似的任務(源域)來遷移學習。
- 從頭建立模型會較複雜且耗時，因此通過遷移學習可以加快學習效率。

03

模型訓練與績效



資料介紹

資料來源

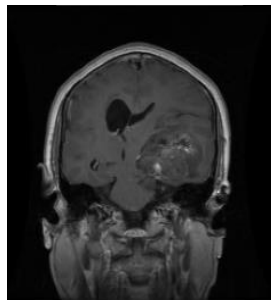
本研究使用Kaggle網站中腦腫瘤MRI圖像的公開資料集。資料集中包含四種腦瘤類別之MRI圖像，分別為膠質瘤 (glioma tumor)、腦膜瘤 (meningioma tumor)、腦下垂體腫瘤 (pituitary tumor) 及無腫瘤(no tumor)。

原始資料集

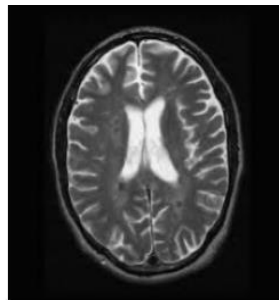
分為 **train** 及 **test** 兩個資料集
訓練資料集 (**train**) 包含 **2870** 筆資料
測試資料集 (**test**) 包含 **394** 筆資料
→ 合共 **3264** 筆數據

腫瘤類別	訓練集 數量	測試集 數量
glioma tumor(膠質瘤)	826張	100張
meningioma tumor(腦膜瘤)	822張	115張
no tumor(無腫瘤)	395張	105張
pituitary tumor(垂體瘤)	827張	74張

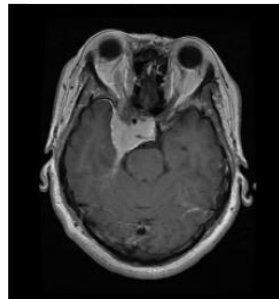
glioma tumor



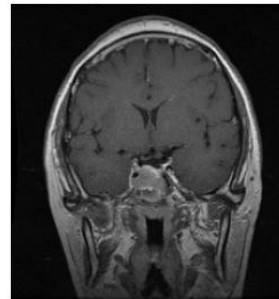
no tumor



meningioma tumor



pituitary tumor

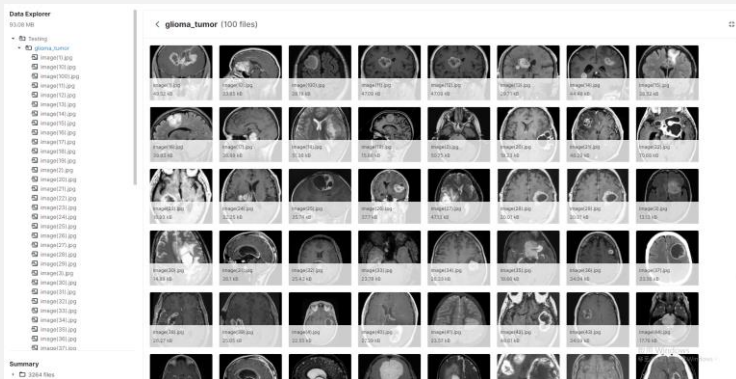


資料前處理 (1/3)

Step 1

資料檢查

檢查資料集中是否有空白
或模糊影像

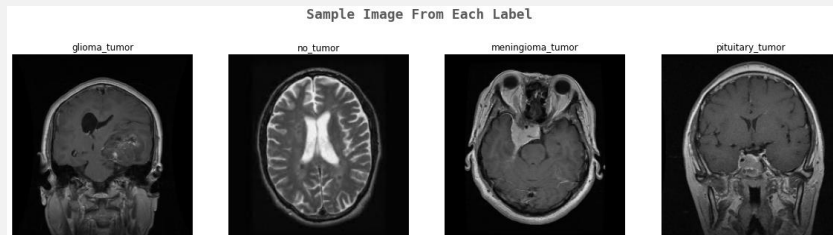


檢查資料集狀況

Step 2

圖片大小統一

應用resize手法統一圖片大小



使圖片大小統一

資料前處理 (2/3)

Step 3

Label轉碼

應用one hot encoding方式
將label轉碼

```
[ ] y_train_new = []
for i in y_train:
    y_train_new.append(labels.index(i))
y_train = y_train_new
y_train = tf.keras.utils.to_categorical(y_train)

y_test_new = []
for i in y_test:
    y_test_new.append(labels.index(i))
y_test = y_test_new
y_test = tf.keras.utils.to_categorical(y_test)
```

label轉碼

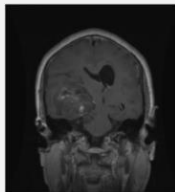
Step 4

資料數據增強

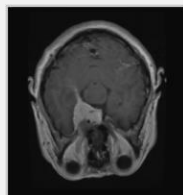
應用水平與垂直翻轉兩種數據
增強方式

```
folderPath = os.path.join('brain-tumor-classification-mri2', 'Training', i)
# tqdm python中展示進度條的語法
ind=1
for j in tqdm(os.listdir(folderPath)):
    # 從指定的資料夾中讀取圖像
    img = cv2.imread(os.path.join(folderPath, j))

    new_img = cv2.flip(img, 0) #cv2.flip(img, 0) 代表上下翻轉, cv2.flip(img, 1)代表左右翻轉
    save_path='./Training2/no_tumor/'+str(ind)+'.jpg'
    ind=ind+1
    '''呼叫cv.2的imwrite函式儲存圖片'''
    cv2.imwrite(save_path, new_img)
```



數據增強



資料前處理 (3/3)

Step 5

資料混和+重新分割

將訓練與測試集混和再以
9:1重新分割訓練與測試集

```
[ ] #shuffle是將圖片順序隨機打亂
X_train, y_train = shuffle(X_train,y_train, random_state=101)

# train_test_split為交叉驗證的函數, 函數中X_train代表所要劃分的樣本特徵集, y_train代表所要劃分的樣本結果, test_size代表樣本占比
X_train,X_test,y_train,y_test = train_test_split(X_train,y_train, test_size=0.1,random_state=101)
```

Step 6

資料分割 (訓練與驗證集)

以9:1將訓練集分割成訓練與
驗證集

```
[ ] history = model.fit(X_train,y_train,validation_split=0.1, epochs =50, verbose=1, batch_size=28,
                        callbacks=[tensorboard,checkpoint,reduce_lr])
```

模型建立與訓練 (1/3)

利用**TensorFlow**開源軟體庫進行訓練，使用其中預訓練過的**CNN**網路架構

```
InceptionV3 = tf.keras.applications.InceptionV3(weights='imagenet', include_top=False, input_shape=(image_size, image_size, 3))
```

額外加入**2D**全局平均池化層減少模型參數，避免過擬合；以及加入**Dropout**層減少訓練的時間，最後再利用**Softmax**函數確保能更有效的分類

```
model = InceptionV3.output
model = tf.keras.layers.GlobalAveragePooling2D()(model)
model = tf.keras.layers.Dropout(rate=0.6)(model)
model = tf.keras.layers.Dense(4, activation='softmax')(model)
model = tf.keras.models.Model(inputs=InceptionV3.input, outputs = model)
```

模型建立與訓練 (2/3)

使用**Optimizer**自動調整學習率，加快訓練速度，與避免學習率過大導致無法收斂。

```
model.compile(loss='categorical_crossentropy', optimizer = 'AdaDelta', metrics= ['accuracy'])
```

應用了 **ReduceLRonPlateau** 函數，只要連續幾個 **epoch** 正確率無提高時，學習率便會乘上一個小於 **1** 的倍數，幫助訓練。

```
[ ] # ReduceLRonPlateau 可以在訓練過程中優化學習率。val_accuracy表示要監控的是accuracy, 只要它一直沒提升就會調整學習率; factor表示要改變學習率時所改變的幅度  
# patience表示每幾個epoch沒改變就會去調整學習率; min_delta為閾值, 只有改變量超過這個數字才會被採用  
reduce_lr = ReduceLRonPlateau(monitor = 'val_accuracy', factor = 0.3, patience = 2, min_delta = 0.001,  
                               mode='auto', verbose=1)
```

模型建立與訓練 (3/3)

模型架構圖:

```
batch_normalization_92 (Batch Normalization) (None, 8, 8, 384) 1152 ['conv2d_92[0][0]']
conv2d_93 (Conv2D) (None, 8, 8, 192) 393216 ['average_pooling2d_8[0][0]']
batch_normalization_85 (Batch Normalization) (None, 8, 8, 320) 960 ['conv2d_85[0][0]']
activation_87 (Activation) (None, 8, 8, 384) 0 ['batch_normalization_87[0][0]']
activation_88 (Activation) (None, 8, 8, 384) 0 ['batch_normalization_88[0][0]']
activation_91 (Activation) (None, 8, 8, 384) 0 ['batch_normalization_91[0][0]']
activation_92 (Activation) (None, 8, 8, 384) 0 ['batch_normalization_92[0][0]']
batch_normalization_93 (Batch Normalization) (None, 8, 8, 192) 576 ['conv2d_93[0][0]']
activation_85 (Activation) (None, 8, 8, 320) 0 ['batch_normalization_85[0][0]']
mixed9_1 (Concatenate) (None, 8, 8, 768) 0 ['activation_87[0][0]',
'activation_88[0][0]']
concatenate_1 (Concatenate) (None, 8, 8, 768) 0 ['activation_91[0][0]',
'activation_92[0][0]']
activation_93 (Activation) (None, 8, 8, 192) 0 ['batch_normalization_93[0][0]']
mixed10 (Concatenate) (None, 8, 8, 2048) 0 ['activation_85[0][0]',
'mixed9_1[0][0]',
'concatenate_1[0][0]',
'activation_93[0][0]']
global_average_pooling2d (Global Average Pooling2D) (None, 2048) 0 ['mixed10[0][0]']
dropout (Dropout) (None, 2048) 0 ['global_average_pooling2d[0][0]']
dense (Dense) (None, 4) 8196 ['dropout[0][0]']
```

Total params: 21,810,980
Trainable params: 21,776,548
Non-trainable params: 34,432

參數優化 (1/2)

- 以三種CNN網路架構 (EfficientNetB5, InceptionV3, DenseNet121) , 預訓練時得到的最佳參數模型為基礎 , 針對我們選擇的四種參數 , 去做參數優化的實驗

因子	說明	水準1	水準2	水準3
A	Dropout	0.4	0.5	0.6
B	Optimizer	Adam	AdaDelta	Adagrad
C	Batch Size	2	4	8
D	Data Augmentation	無	水平翻轉	垂直翻轉

- 利用了實驗設計中的田口方法 , 有效減少調整參數的總次數 , 並獲得相同的結果。
- 選擇了上述所提到的四項參數作為四個因子 , 並使用三水準 , 應用L9直交表來幫助參數優化。
- 每個實驗皆使用 **30 epoch**

參數優化(2/2)

L9 實驗設計參數組合:

實驗	Dropout	Optimizer	Batch Size	Data augmentation
1	0.4	Adam	2	無
2	0.4	AdaDelta	4	水平翻轉
3	0.4	Adagrad	8	垂直翻轉
4	0.5	Adam	4	垂直翻轉
5	0.5	AdaDelta	8	無
6	0.5	Adagrad	2	水平翻轉
7	0.6	Adam	8	水平翻轉
8	0.6	AdaDelta	2	垂直翻轉
9	0.6	Adagrad	4	無

實驗設計：EfficientNetB5

實驗	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
1	0.4	Adam	2	無	0.95918
2	0.4	AdaDelta	4	水平翻轉	0.89796
3	0.4	Adagrad	8	垂直翻轉	0.98129
4	0.5	Adam	4	垂直翻轉	0.98810
5	0.5	AdaDelta	8	無	0.78571
6	0.5	Adagrad	2	水平翻轉	0.86224
7	0.6	Adam	8	水平翻轉	0.98469
8	0.6	AdaDelta	2	垂直翻轉	0.86054
9	0.6	Adagrad	4	無	0.93197



Minitab結果：EfficientNetB5

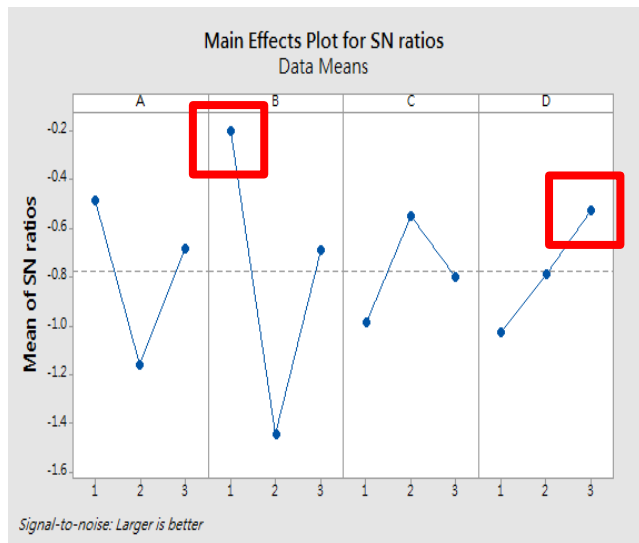
因子	說明	水準1	水準2	水準3
A	Dropout	0.4	0.5	0.6
B	Optimizer	Adam	AdaDelta	Adagrad
C	Batch Size	2	4	8
D	Data Augmentation	無	水平翻轉	垂直翻轉

Response Table for Signal to Noise Ratios
Larger is better

Level	A	B	C	D
1	-0.4870	-0.2000	-0.9847	-1.0229
2	-1.1621	-1.4447	-0.5503	-0.7854
3	-0.6835	-0.6878	-0.7976	-0.5242
Delta	0.6751	1.2447	0.4344	0.4987
Rank	2	1	4	3

Response Table for Means

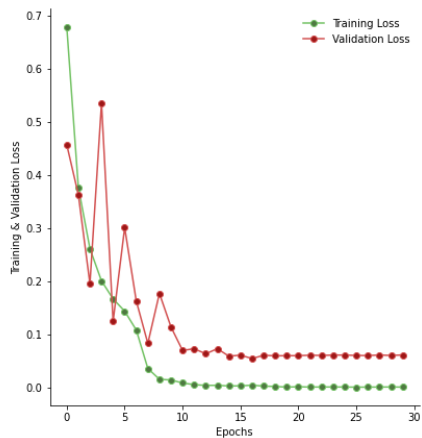
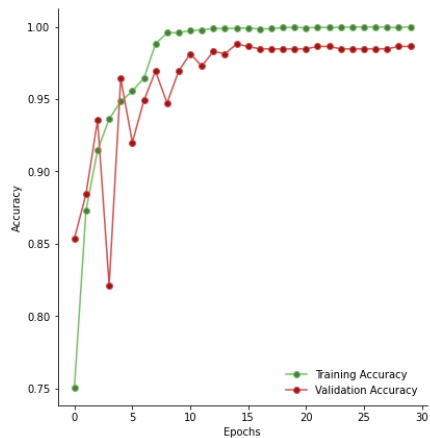
Level	A	B	C	D
1	0.9461	0.9773	0.8940	0.8923
2	0.8787	0.8481	0.9393	0.9150
3	0.9257	0.9252	0.9172	0.9433
Delta	0.0675	0.1293	0.0454	0.0510
Rank	2	1	4	3



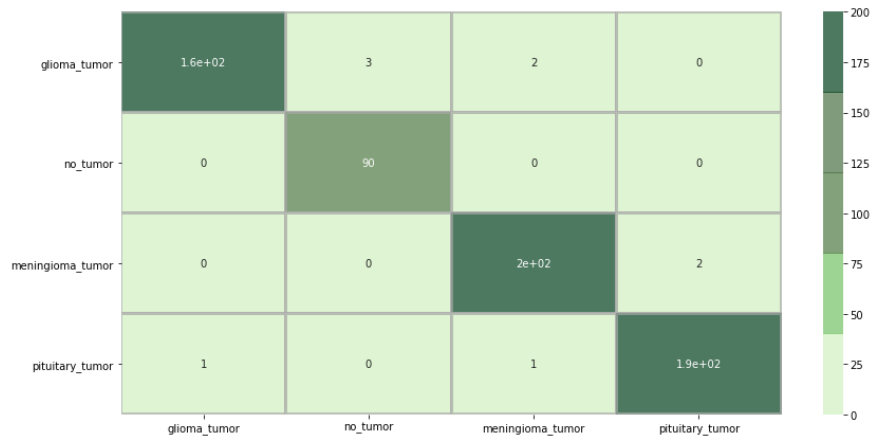
進一步調參：EfficientNetB5

實驗	備註	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
10	最佳組合	0.4	Adam	4	垂直翻轉	0.98

Epochs vs. Training and Validation Accuracy/Loss



Heatmap of the Confusion Matrix



實驗設計：InceptionV3

實驗	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
1	0.4	Adam	2	無	0.87075
2	0.4	AdaDelta	4	水平翻轉	0.95918
3	0.4	Adagrad	8	垂直翻轉	0.97279
4	0.5	Adam	4	垂直翻轉	0.96769
5	0.5	AdaDelta	8	無	0.90816
6	0.5	Adagrad	2	水平翻轉	0.98639
7	0.6	Adam	8	水平翻轉	0.9989
8	0.6	AdaDelta	2	垂直翻轉	0.88095
9	0.6	Adagrad	4	無	0.97619



Minitab結果：InceptionV3

因子	說明	水準1	水準2	水準3
A	Dropout	0.4	0.5	0.6
B	Optimizer	Adam	AdaDelta	Adagrad
C	Batch Size	2	4	8
D	Data Augmentation	無	水平翻轉	垂直翻轉

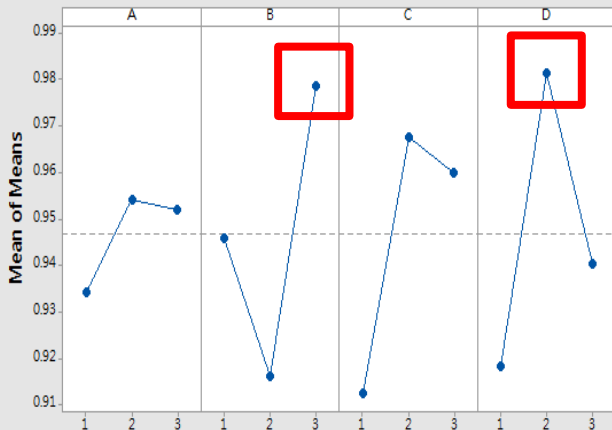
Response Table for Signal to Noise Ratios
Larger is better

Level	A	B	C	D
1	-0.6012	-0.4990	-0.8074	-0.7494
2	-0.4137	-0.7666	-0.2855	-0.1635
3	-0.4399	-0.1893	-0.3620	-0.5420
Delta	0.1876	0.5773	0.5218	0.5859
Rank	4	2	3	1

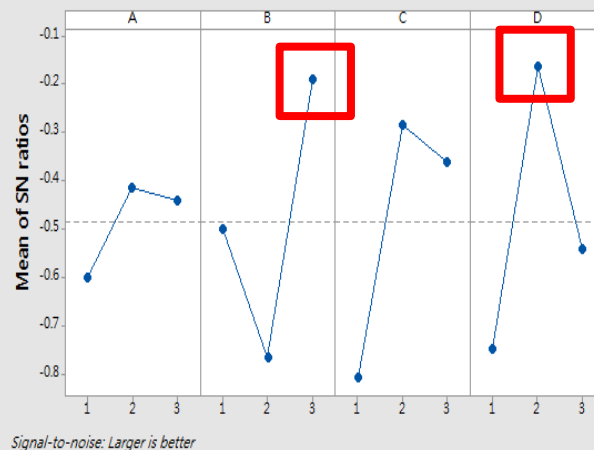
Response Table for Means

Level	A	B	C	D
1	0.9342	0.9458	0.9127	0.9184
2	0.9541	0.9161	0.9677	0.9815
3	0.9520	0.9785	0.9599	0.9405
Delta	0.0198	0.0624	0.0550	0.0631
Rank	4	2	3	1

Main Effects Plot for Means
Data Means



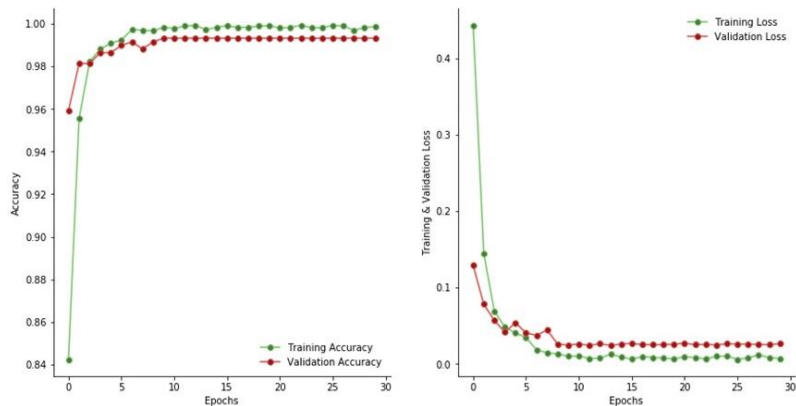
Main Effects Plot for SN ratios
Data Means



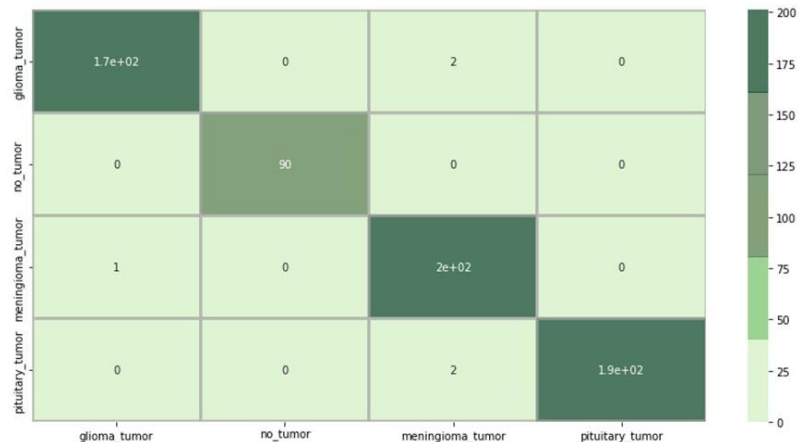
進一步調參：InceptionV3

實驗	備註	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
10	最佳組合	0.5	Adagrad	4	水平翻轉	0.9932

Epochs vs. Training and Validation Accuracy/Loss



Heatmap of the Confusion Matrix



實驗設計：DenseNet121

實驗	Dropout	Optimizer	Batch size	Data augmentation	Test Accuracy
1	0.4	Adam	2	無	0.8980
2	0.4	AdaDelta	4	水平翻轉	0.9354
3	0.4	Adagrad	8	垂直翻轉	0.9813
4	0.5	Adam	4	垂直翻轉	0.9677
5	0.5	AdaDelta	8	無	0.8912
6	0.5	Adagrad	2	水平翻轉	0.9694
7	0.6	Adam	8	水平翻轉	0.9864
8	0.6	AdaDelta	2	垂直翻轉	0.8622
9	0.6	Adagrad	4	無	0.9864



Minitab結果：DenseNet121

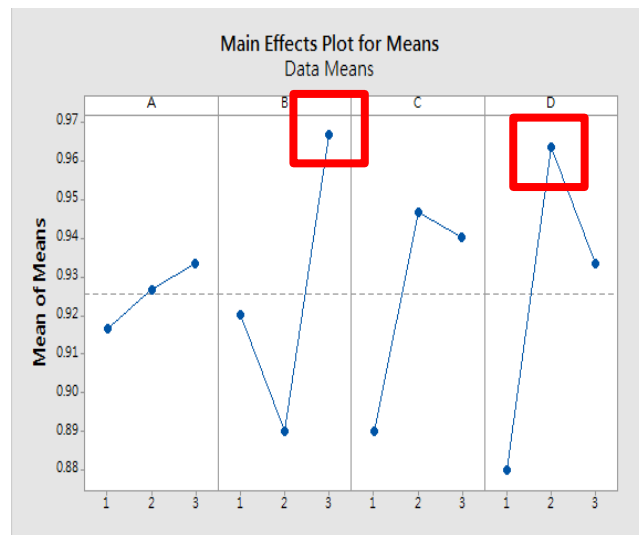
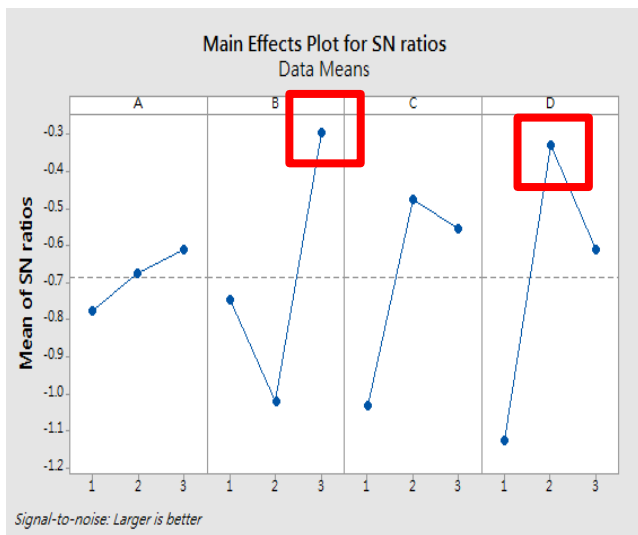
因子	説明	水準1	水準2	水準3
A	Dropout	0.4	0.5	0.6
B	Optimizer	Adam	AdaDelta	Adagrad
C	Batch Size	2	4	8
D	Data Augmentation	無	水平翻轉	垂直翻轉

Response Table for Signal to Noise Ratios
Larger is better

Level	A	B	C	D
1	-0.7771	-0.7465	-1.0309	-1.1247
2	-0.6734	-1.0190	-0.4762	-0.3258
3	-0.6102	-0.2952	-0.5537	-0.6102
Delta	0.1669	0.7238	0.5547	0.7988
Rank	4	2	3	1

Response Table for Means

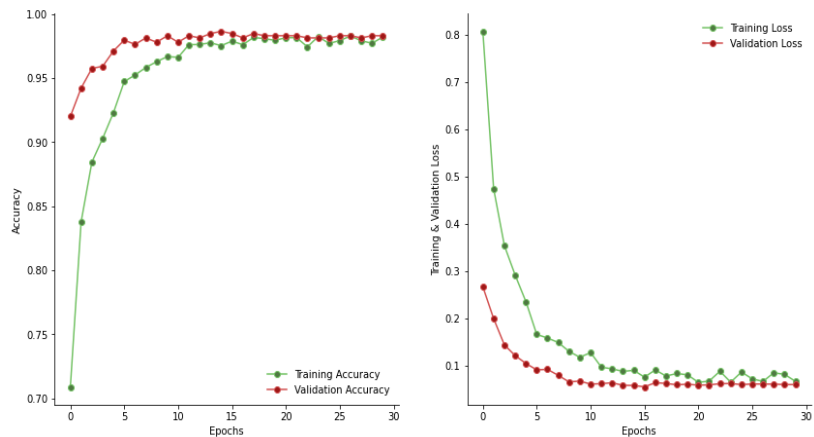
Level	A	B	C	D
1	0.9167	0.9200	0.8900	0.8800
2	0.9267	0.8900	0.9467	0.9633
3	0.9333	0.9667	0.9400	0.9333
Delta	0.0167	0.0767	0.0567	0.0833
Rank	4	2	3	1



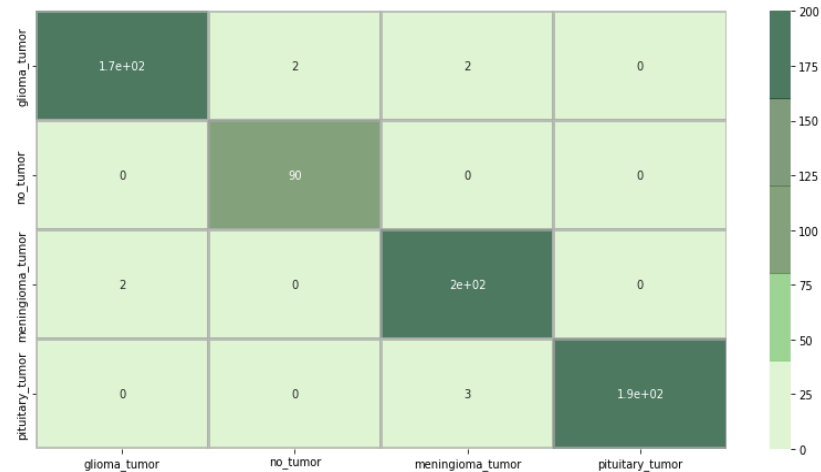
進一步調參：DenseNet121

實驗	備註	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
10	最佳組合	0.6	Adagrad	4	水平翻轉	0.9864

Epochs vs. Training and Validation Accuracy/Loss



Heatmap of the Confusion Matrix



04

結果與未來展望



結果討論

最佳水準組合

模型	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
EfficientNetB5	0.4	Adam	4	垂直翻轉	0.98
InceptionV3	0.5	Adagrad	4	水平翻轉	0.9932
DenseNet121	0.6	Adagrad	4	水平翻轉	0.9864

最佳精確度組合

模型	Dropout	Optimizer	Batch Size	Data augmentation	Test Accuracy
EfficientNetB5	0.5	Adam	4	垂直翻轉	0.9881
InceptionV3	0.6	Adam	8	水平翻轉	0.9989
DenseNet121	0.6	Adam	8	水平翻轉	0.9864
	0.6	Adagrad	4	無	0.9864

具顯著影響的因子水準：**Adam&Adagrad > AdaDelta**、進行資料增強

貢獻

- 統整各個常見的**CNN**網路架構，分別測試對腦腫瘤圖片分類的有效性。
- 提供公開可檢視的數據，讓大家可以依各自在精確度或訓練速度上的需求，選用最適合自己的網路架構。

侷限性

- 此次使用的資料集圖片品質較佳，訓練時並未考量到低畫質與模糊圖片的情況。
- 若用於測試的圖片有經過擾動，可能會導致分類精確度的下降。
- 數據增強手法過於單一，可能錯失使精確度更加提升的增強手法。



適用性

- 最終結果佳 → 可以應用於臨床實施的機率非常大
- 模型高精確度 → 快速做出判斷，提供醫生建議，醫生只須依照相關知識進行最後確認

未來展望

- 除了腦腫瘤類別的辨識，我們認為可以再進一步框出腫瘤的位置，讓醫生的判讀速度可以更加快速
- 可以將此模型應用於其餘部位的腫瘤**MRI**圖片，來確認模型的泛化性。



Thank You!

