

國立清華大學

智慧化企業整合

Intelligent Integration of Enterprise

Project 2

腦部腫瘤 MRI 影像之分類模型建立

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

目錄

一、背景介紹.....	3
(一) 背景與動機.....	3
(二) 研究目的.....	3
(三) 問題描述.....	4
(四) 資料集介紹.....	4
二、研究方法.....	4
(一) 方法選擇.....	4
(二) 遷移學習(Transfer learning).....	7
三、個案研究.....	8
(一) 資料介紹.....	8
(二) 資料前處理.....	8
(三) 模型建立與訓練.....	9
(四) 參數優化.....	10
(五) 實驗設計結果.....	11
四、結論.....	13
(一) 貢獻	
(二) 侷限性	
(三) 適用性	
(四) 未來展望	
五、參考資料	

圖目錄

圖 1 腦腫瘤 MRI 影像.....	3
圖 2 CNN 網路架構初選結果.....	5
圖 3 EfficientNetB5 網路架構.....	6
圖 4 InceptionV3 網路架構.....	6
圖 5 DenseNet121 網路架構.....	7
圖 6 應用 resize 使圖片大小統一.....	8
圖 7 label 轉碼程式.....	9
圖 8 水平與垂直翻轉 Data augmentation 程式.....	9
圖 9 重新分割訓練與資料集.....	9
圖 10 分割訓練與驗證集.....	9
圖 11 模型建立.....	10
圖 12 學習率調整.....	10

表目錄

表 1 5W1H.....	4
表 2 各類別圖片數量表.....	8
表 3 實驗因子及水準.....	10
表 4 L9 實驗設計參數組合.....	11
表 5 EfficientNetB5 實驗設計結果.....	11
表 6 EfficientNetB5 最佳組合.....	11
表 7 InceptionV3 實驗設計結果.....	12
表 8 InceptionV3 最佳組合.....	12
表 9 DenseNet121 實驗設計結果.....	12
表 10 DenseNet121 最佳組合.....	13

一、背景介紹

(一) 背景與動機

腦腫瘤被認為是攻擊性的疾病之一，佔所有原發性中樞神經系統 (CNS) 腫瘤的 85% 至 90%。每年約有 11,700 人被診斷出患有腦癌。患有腦癌或中樞神經系統腫瘤的人之 5 年生存率，男性約為 34%，女性約為 36%。

腦腫瘤分為：良性腫瘤、惡性腫瘤、垂體瘤等。不同類型的腦腫瘤應實施對應的治療、計劃和準確的診斷，以提高患者的預期壽命。而現今檢測腦腫瘤的最佳技術是磁共振造影 (MRI)，通過掃描產生大量的圖像數據。這些影像皆由放射科醫師進行人工檢查及判斷。

但是腦腫瘤非常複雜，其大小和位置有很多種不同及異常情況。這使完全了解腫瘤的性質非常困難。此外，MRI 分析需要專業的神經外科醫生。有些發展中國家或醫療水平較低的國家缺乏有經驗的醫生和缺乏對腦腫瘤的了解，使得從 MRI 判斷腫瘤報告變得非常具有挑戰性和耗時。

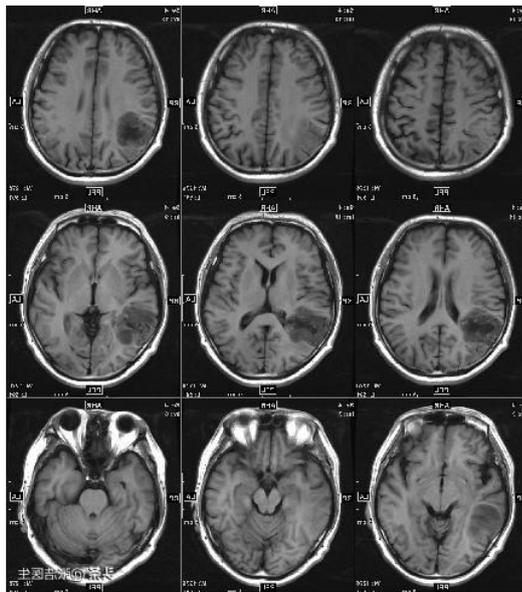


圖 1 腦腫瘤 MRI 影像

(二) 研究目的

由於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，因此本研究透過建立腦部腫瘤 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。

(三) 問題描述

表 1 5W1H

What	腦腫瘤 MRI 人工閱片複雜耗時，需依賴醫生的經驗，有機會因經驗不足而辨識錯誤
Why	透過深度學習的模型，建立腦部腫瘤 MRI 影像的分類模型，提供醫生作診斷參考根據，減少閱片時間，並降低人工辨識錯誤機會。
Where	醫院、放射科
When	病患有腦腫瘤疑慮需進行 MRI 影像分析時
Who	病患、病患家屬、放射科醫生
How	卷積神經網絡(CNN)模型訓練

(四) 資料集介紹

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)。

資料集分為 train 及 test，前者資料集包含 2870 筆資料，而後者測試資料集包含 394 筆資料，合共 3264 筆數據。

二、 研究方法

(一) 方法選擇

本研究使用 CNN 對腦部腫瘤圖片進行辨識與分類。現今 CNN 的網路架構眾多，且在實務領域中各有各的長處。因此我們先將多種在 ImageNet 圖片資料集預訓練過的 CNN 網路，套用在本次研究的圖片資料集中，並選出三個精確度高且所花時間短的 CNN 網路架構來進行最終的比較。

CNN 網路架構初選結果如圖 2：

CNN網路架構名稱
EfficientNetB0
EfficientNetB1
EfficientNetB2
EfficientNetB3
EfficientNetB4
EfficientNetB5
EfficientNetB6
EfficientNetB7
DenseNet121
DenseNet169
DenseNet201
InceptionResNetV2
InceptionV3
VGG16
VGG19
Xception
MobileNet
MobileNetV2
MobileNetV3Large
MobileNetV3Small
NASNetMobile
ResNet50
ResNet50V2
ResNet101
ResNet101V2
ResNet152
ResNet152V2

圖 2 CNN 網路架構初選結果

因此最終我們選擇 EfficientNetB5、DenseNet121 與 InceptionV3 這三個網路作為本次研究的 CNN 分類網路架構。

(1) EfficientNetB5

EfficientNet 是 Google 所提出的一個新的基準網路。它應用了新的網路縮放方法，利用複合係數均勻地縮放網路深度(網路層數)、寬度(通道數)與圖像解析度(輸入圖片大小)，使其能在固定的資源限制下，達到更好的精確度。而在 ImageNet 中的圖片分類任務上，EfficientNet 也超越熱門的 CNN 網路像是 ResNet、DenseNet、ResNeXt 等，達到最高的 Top-1 精確度。

作者使用自動化模型結構搜尋法(Neural Architecture Search)，產生一個 Baseline Model，稱為 EfficientNetB0(B 代表 Baseline，0 代表 N=0)。接著使用複合式模型縮放，以 N=1 到 7 分別產生了 7 個擴充模型，命名為 EfficientNetB1 到 EfficientNetB7。EfficientNetB5 網路架構如圖 3：

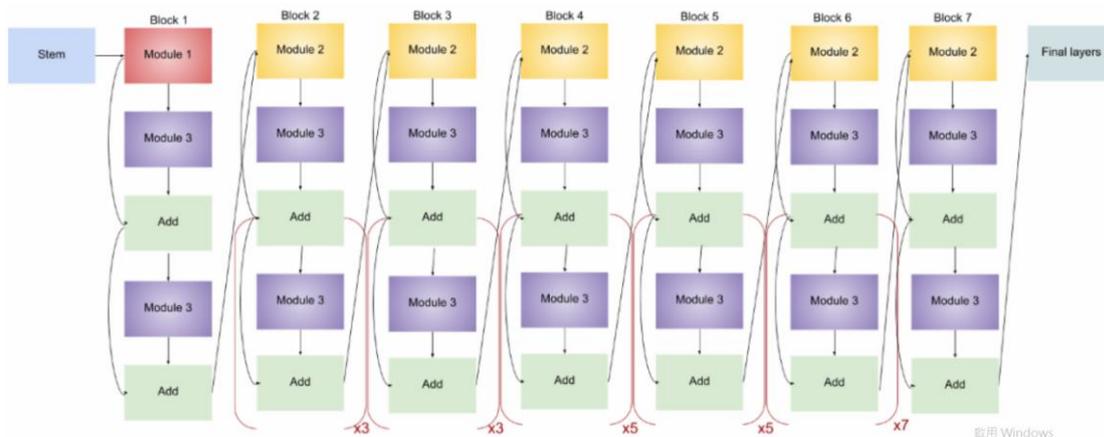


圖 3 EfficientNetB5 網路架構

(參考資料：<https://cloud.tencent.com/developer/article/1701843>)

(2) InceptionV3

GoLeNet 最大的特點就是使用了 Inception 模塊，它的目的是設計一種具有優良局部拓撲結構的網絡，即對輸入圖像並行地執行多個卷積運算或池化操作，並將所有輸出結果拼接為一個非常深的特徵圖。

InceptionV3 一個最重要的改進是卷積分解(Factorization)，將 7x7 卷積分解成兩個一維的卷積串聯(1x7 和 7x1)，3x3 卷積分解為兩個一維的卷積串聯(1x3 和 3x1)，這樣既可以加速計算，又可使網絡深度進一步增加，增加了網絡的非線性即為每增加一層都要進行 ReLU。另外，網絡輸入從 224x224 變為 299x299。其他也增加了輔助分類器(BatchNorm)、標籤平滑和 RMSProp 優化器。InceptionV3 網路架構如圖 4：

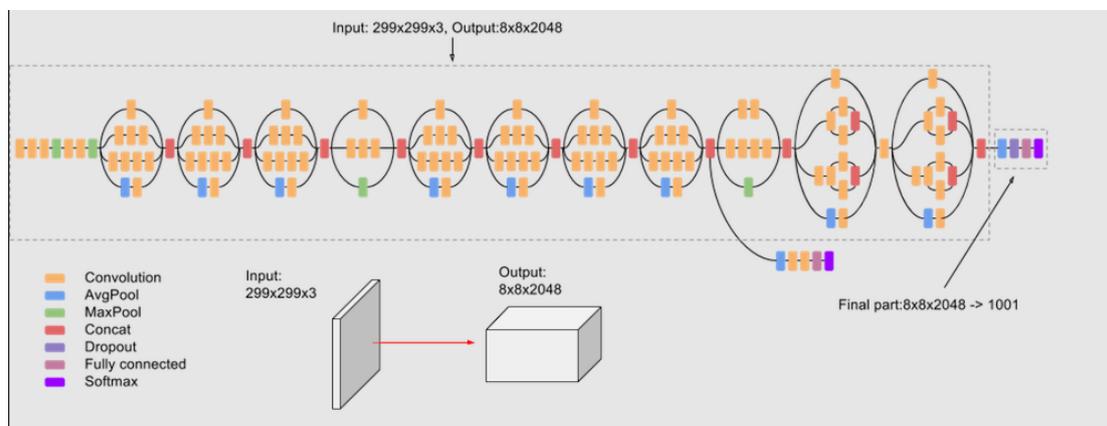


圖 4 InceptionV3 網路架構

(參考資料：<https://paperswithcode.com/method/inception-v3>)

(3) DenseNet121

DenseNet 提出了一個更激進的密集連接機制：即互相連接所有的層，具體來說就是每個層都會接受其前面所有層作為其額外的輸入。另一大特色是通過特徵在 channel 上的連接來實現特徵重用(feature reuse)。這些特點有別於以往的神經網絡，不從網路深度與寬度著手，而是由特徵的角度去考慮，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，並達到更好的準確率，讓 DenseNet 在參數和計算成本更少的情形下實現更優的性能。

DenseNet121 網路架構如圖 5：

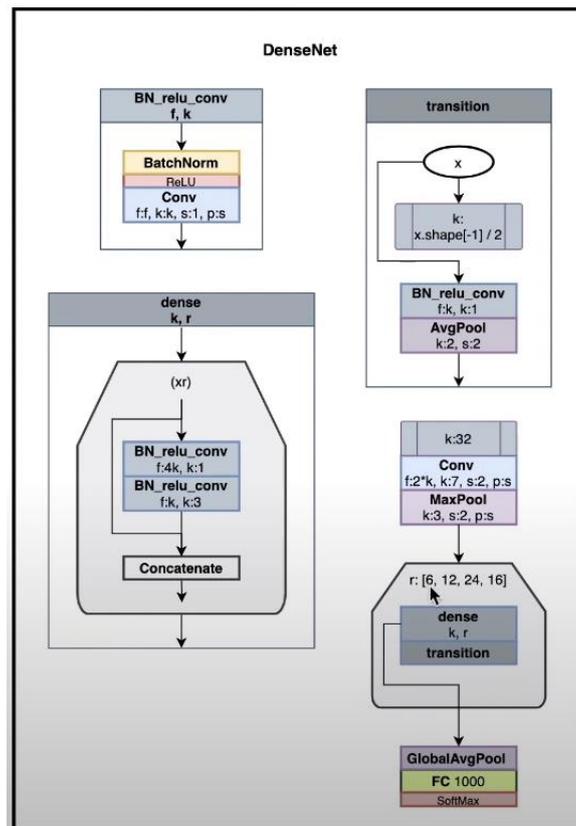


圖 5 DenseNet121 網路架構

(參考資料：<https://www.youtube.com/watch?v=3ZPJyknZolE>)

(五) 遷移學習(Transfer learning)

遷移學習(Transfer Learning)是一種機器學習方法，是把一個領域(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。且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。

本研究初選時所選擇的 CNN 網路結構，皆已在 ImageNet 圖片資料集中進

行預訓練，因此大幅提升本研究的訓練效率。

三、個案研究

(一) 資料介紹

本次研究所使用的 Dataset 在 Kaggle 上所取得，為腦部腫瘤的 MRI 圖片。總共有四種類別，分別為 glioma tumor(膠質瘤)、meningioma tumor(腦膜瘤)、no tumor(無腫瘤)與 pituitary tumor(垂體瘤)。其原檔案有 2870 張訓練集圖片與 394 張測試集圖片檔(.jpg)，皆為有標籤資料。表 2 為各類別圖片數量表：

表 2 各類別圖片數量表

腫瘤類別	訓練集圖片數量	測試集圖片數量
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

(二) 資料前處理

Step 1 :

由於我們非專業醫界人士，無法判別腦腫瘤分類是否正確。因此我們人工檢視是否有空白或模糊 MRI 圖片於其中，最終確定圖片集皆為清晰腦部 MRI 照片。

Step 2 :

由於原本的圖片大小不一，且大部分 CNN 網路架構皆要求 input 圖片大小須為 224x224。因此如圖 6，我們利用 resize 手法將圖片統一大小。最後再將 label 利用 one hot encoding 的方式，使其能對應到各個圖片。轉碼方式如圖 7：

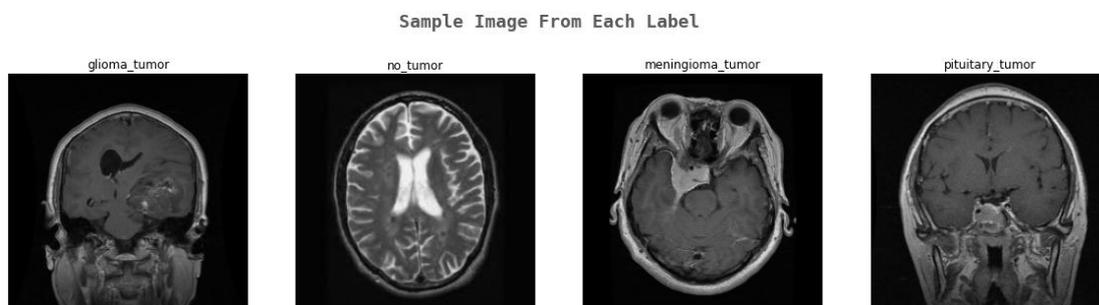


圖 6 應用 resize 使圖片大小統一

```
[ ] 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) # 將label轉成one hot encoding的形式

    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)
```

圖 7 label 轉碼程式

Step 3 :

為了避免訓練圖片資料過少，使用水平與垂直翻轉的 Data augmentation 方式，將資料量擴張一倍。其程式碼如圖 8：

```
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)
```

圖 8 水平與垂直翻轉 Data augmentation 程式

Step 4 :

我們認為原先測試集圖片總數過多，且沒額外劃分驗證集圖片資料。因此我們先將訓練集與測試集圖片合併，隨機打亂後再按照訓練與測試集 9:1 的比例重新劃分。如圖 9：

```
[ ] #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)
```

圖 9 重新分割訓練與資料集

最後再將訓練集以 9:1 分割為訓練與驗證集。如圖 10：

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

圖 10 分割訓練與驗證集

(三) 模型建立與訓練

本研究利用 TensorFlow 開源軟體庫進行訓練，其提供許多已在 ImageNet 圖

片資料集中預訓練過的 CNN 網路架構。因此我們除了引入網路外，並再最後加入 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，也能避免過擬合；最後再利用 Softmax 函數確保能更有效的分類。模型建立的程式碼如圖 11：

```
[ ] effnet = tf.keras.applications.densenet.DenseNet121(weights='imagenet', include_top=False, input_shape=(image_size, image_size, 3))
model = effnet.output
model = tf.keras.layers.GlobalAveragePooling2D()(model)
model = tf.keras.layers.Dropout(rate=0.5)(model)
model = tf.keras.layers.Dense(4, activation='softmax')(model)
model = tf.keras.models.Model(inputs=effnet.input, outputs = model)
```

圖 11 模型建立

除了使用 Optimizer 自動調整學習率與其他相關參數外，為了加快訓練速度，與避免學習率過大導致無法收斂，我們還應用了 ReduceLRonPlateau 函數。只要在連續幾個 epoch 中，驗證集的 accuracy 皆未提升，當前的學習率便會乘上一個小於 1 的倍數，幫助訓練。相關程式碼如圖 12：

```
[ ] # 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)
```

圖 12 學習率調整

(四) 參數優化

三種 CNN 網路架構除了預訓練時所獲得的最佳模型參數外，其餘的參數皆是使用相同的數據，還有超參數優化的空間。希望可以在有限的實驗次數中找到最佳模型參數來達到最高的準確率，節省時間以及減少人為調整的因素。挑選出的因子以及水準如表 3：

表 3 實驗因子及水準

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

我們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。我們選擇了上述所提到的四項參數作為四個因子，並使用三個水準，應用 L9 實驗設計參數組合來幫助參數優化。針對每次實驗時都統一使用 30 epoch 進行訓練，實驗設計參數組合詳情如表 4：

表 4 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

(五) 實驗設計結果

1. EfficientNetB5

表 5 EfficientNetB5 實驗設計結果

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

表 5 為 EfficientNetB5 的實驗組合及結果，準確度最高的為實驗 4，準確度達到了 0.9881，將這 9 次實驗結果透過統計分析後可以得到各個因子對於準確率的影響程度，我們又從中挑選出了各個因子中最好的水準組合來做測試，希望可以得到更好的結果，最佳組合為表 6，準確度為 0.98。

表 6 EfficientNetB5 最佳組合

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

2. InceptionV3

表 7 InceptionV3 實驗設計結果

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

表 7 為 InceptionV3 的實驗組合及結果，準確度最高的為實驗 7，準確度達到了 0.9989，將這 9 次實驗結果透過統計分析後可以得到各個因子對於準確率的影響程度，我們又從中挑選出了各個因子中最好的水準組合來做測試，希望可以得到更好的結果，最佳組合為表 8，準確度為 0.98。

表 8 InceptionV3 最佳組合

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

3. DenseNet121

六、表 9 DenseNet121 實驗設計結果

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

表 9 為 DenseNet121 的實驗組合及結果，準確度最高的為實驗 7，準確度

達到了 0.9864，將這 9 次實驗結果透過統計分析後可以得到各個因子對於準確率的影響程度，我們又從中挑選出了各個因子中最好的水準組合來做測試，希望可以得到更好的結果，最佳組合為表 10，準確度為 **0.9864**。

表 10 DenseNet121 最佳組合

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

四、結論

其中，EfficientNetB5、InceptionV3 以及 DenseNet121 模型在 L9 直交表四種因子九種實驗組合下分別在 Dropout/ Optimizer/ Batch Size/ Data augmentation 得到最佳準確率分別為 0.5/ Adam/ 4/ 垂直翻轉、0.6/ Adam/ 8/ 水平翻轉以及 (0.4/ Adagrad/ 8/ 垂直翻轉+ 0.6/ Adam/ 8/ 水平翻轉)。為了減少由噪聲因子引起的可變性來查看不同因子組合對準確度的影響程度，我們在 Minitab 使用訊號雜訊比 S/N ratio 進行分析，並選出最佳因子組合進行進一步調整參數來訓練模型，得出 EfficientNetB5、InceptionV3 以及 DenseNet121 模型最終準確率為 0.98、0.9991 以及 0.99。最佳因子水準組合以及準確率如表 11 及表 12：

表 11 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

表 12 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

從此次實驗情況中可見，三種 CNN 網絡架構對最終結果有顯著影響的情況為：在 Optimizer 的選擇上 Adam 以及 Adagrad 相比於 AdaDelta 對模型亦有更大的顯著效果、使用資料增強（垂直翻轉/水平翻轉）相比於原始圖像能提供更

多的學習能力，而在實驗結果中三種模型得出的最佳水準組合並非最佳的精確度，這部分推測因子之間也許存在交互作用。

總結來說，對於此次研究結果可歸納於以下四點：

(一) 貢獻

本次研究中，我們統整了各個常見的 CNN 網路架構，並一一嘗試其對腦腫瘤圖片分類的有效性為何。提供了公開可檢視的數據，讓大家能依各自在精確度或訓練速度上的需求，來選用最適合自己的網路架構。

(二) 侷限性

因本研究的圖片資料集品質較佳，訓練時並未考量到低畫質與模糊圖片的情況。因此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；數據增強的手法過於單一，有可能錯失能使精確度更加提升的有效增強手法。

(三) 適用性

由於本研究最終結果較佳，我們認為能應用於臨床實施的機率非常大。由於模型結果的高精確度，當醫生面臨腦腫瘤判斷的抉擇且時間急迫時，即可應用我們的模型快速做出判斷，再依照相關知識確認模型給出的結果是否正確。

(四) 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；也能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。

五、參考資料

(一) <https://www.kaggle.com/jaykumar1607/brain-tumor-mri-classification-tensorflow-cnn>

(二) <https://www.itread01.com/content/1530278414.html>

(三) <https://codingnote.cc/zh-hk/p/153860/>

(四) <https://medium.com/ching-i/inception-%E7%B3%BB%E5%88%97-inceptionv2-inceptionv3-93cd42054d23>

(五) http://ielab.ie.nthu.edu.tw/109_IIE_project/2/109IIE_proj2_6_word.pdf