

# Final Project - Human Protein Atlas Image Classification

隊名：NTU\_r06521504\_隊名我想想

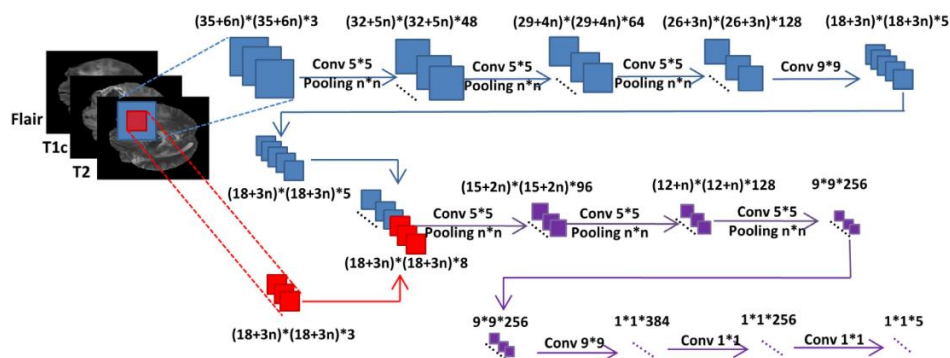
隊員：陳譽仁 土木所交通組 R06521504

趙浩雅 土木所交通組 R06521511

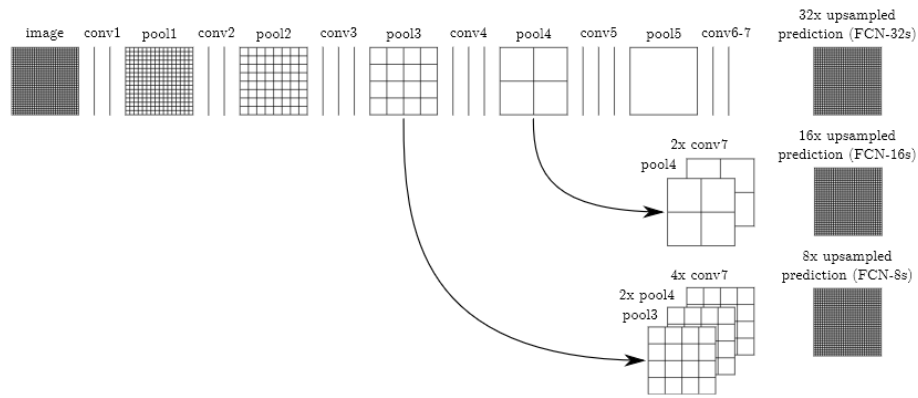
## Introduction & Motivation

近幾年，深度學習成為機器學習中最受重視的一區，而藉由此技術完成的影像辨識系統被廣泛的運用在各個領域；其中，藉由影像辨識的系統，可輔助醫療判斷細胞狀況，藉以進行醫療診斷之輔助。考量到台灣未來 AI 的潛在性，本次報告我們便選擇相關題目，利用 CNN 架構去進行圖像辨識，並用 ensemble 的架構去加強模型。在建立模型部分，本次報告參考兩個方面，分別為 paper 與競賽上參賽者分享的 Kernel。

a. Paper 方面，本報告參考了 Zhao et al., 2018 的研究[1]，本篇論文是對大腦腫瘤進行辨識與圖像的分區(segmentation)，在模式比較底層的部分引用了 Fully Convolutional Neural Network 所衍伸的架構，在模式中的一些部份疊上前幾層的結果，如下圖。



雖然分區不是本報告所選主題要做的事情，但是因為該主題是要辨識散布在圖片上的一些特徵，其中的架構或許可以拿來參考。因此，參考了 FCN 的文章[2]，這個架構將模式較後面的部分 upsampling，再與前面 maxpooling 前的卷積層結果相加以保留一些特徵在圖片上的位置資訊。



b. 初期參考的 Kernel 則是來自 Kaggle 上的 Kernel[3]，其中包含完整的程式碼，其模式為數個卷積層的疊加，其中一個卷積層有四個平行的卷積層，各自使用不同的 kernel 大小，應該是為了方便取出不同大小的特徵並疊加在一起。本報告所使用的模式架構則以[4]作者後續教學文使用的模式架構為基礎，建立出本報告所使用的模式。

## Data Preprocessing \ Feature Engineering

目前讀取資料的方法、f1 score 的計算、Data generator 參考本競賽其他參加者在 Kernel 分享的程式碼[3]，對圖片資料先用 np.stack 的方式將同一細胞之紅、綠、藍、黃圖連接，以便在之後 training 過程讀取特徵值。為了增強數據，本報告用 imgaug 的套件將圖片進行平移、縮放、錯切等動作並加入少量的噪音，由於擔心其數據會造成過大偏差，未使用像素平移。考慮到有部分種類數量過少，會在訓練過程中被忽略或著是被切除，在最開始切資料的時候我們另外進行篩選，避免出現未訓練該項目的狀況。

## Model Description

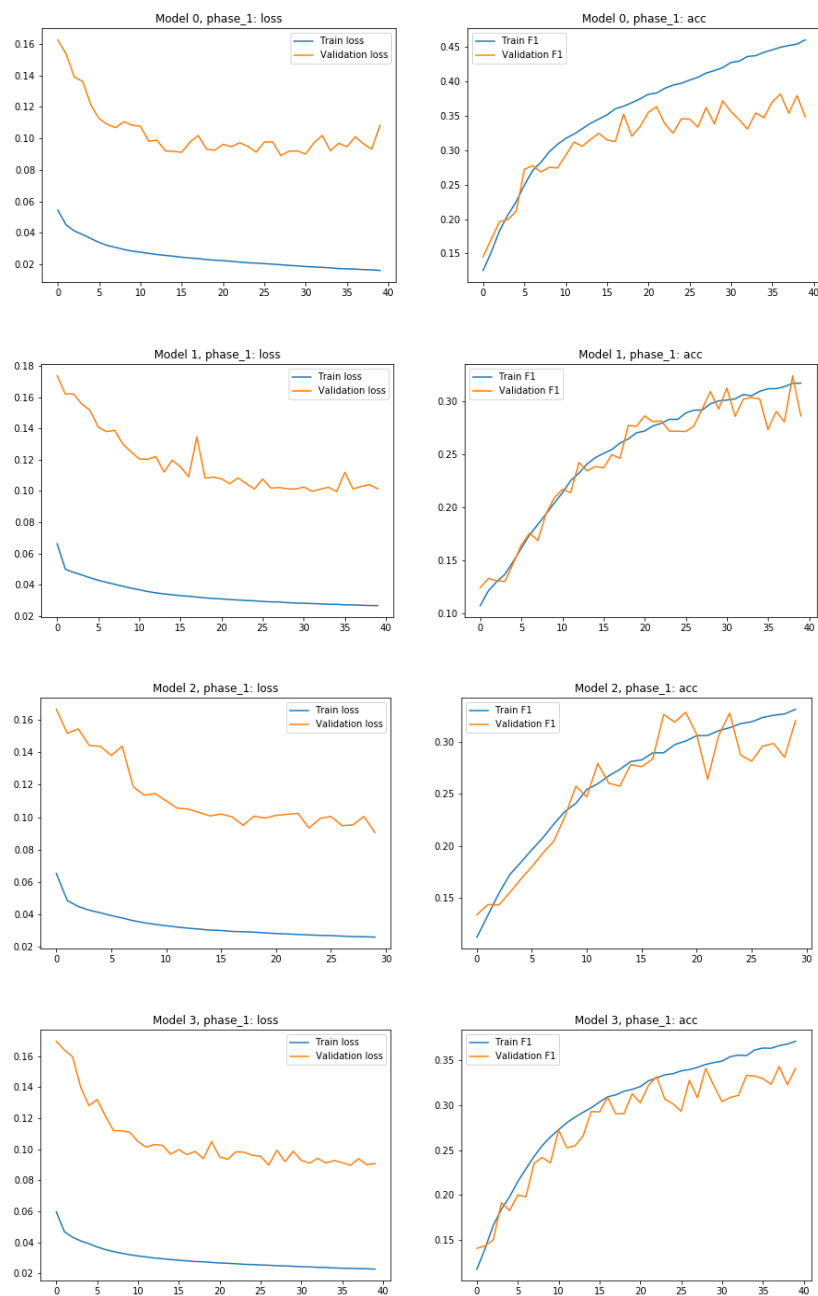
本報告於開始先建立一個 CNN+DNN 的基礎模型，其架構如下圖所示：

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 256, 256, 4)	0	
batch_normalization_1 (BatchNor	(None, 256, 256, 4)	16	input_1[0][0]
conv2d_1 (Conv2D)	(None, 254, 254, 32)	1184	batch_normalization_1[0][0]
batch_normalization_2 (BatchNor	(None, 254, 254, 32)	128	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 127, 127, 32)	0	batch_normalization_2[0][0]

dropout_1 (Dropout)	(None, 127, 127, 32)	0	max_pooling2d_1[0][0]
batch_normalization_3 (BatchNor	(None, 127, 127, 32)	128	dropout_1[0][0]
conv2d_2 (Conv2D)	(None, 63, 63, 64)	18496	batch_normalization_3[0][0]
batch_normalization_4 (BatchNor	(None, 63, 63, 64)	256	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 61, 61, 64)	36928	batch_normalization_4[0][0]
batch_normalization_5 (BatchNor	(None, 61, 61, 64)	256	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 59, 59, 64)	36928	batch_normalization_5[0][0]
batch_normalization_6 (BatchNor	(None, 59, 59, 64)	256	conv2d_4[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 29, 29, 64)	0	batch_normalization_6[0][0]
dropout_2 (Dropout)	(None, 29, 29, 64)	0	max_pooling2d_2[0][0]
batch_normalization_7 (BatchNor	(None, 29, 29, 64)	256	dropout_2[0][0]
conv2d_5 (Conv2D)	(None, 27, 27, 128)	73856	batch_normalization_7[0][0]
batch_normalization_8 (BatchNor	(None, 27, 27, 128)	512	conv2d_5[0][0]
conv2d_6 (Conv2D)	(None, 25, 25, 128)	147584	batch_normalization_8[0][0]
batch_normalization_9 (BatchNor	(None, 25, 25, 128)	512	conv2d_6[0][0]
conv2d_7 (Conv2D)	(None, 23, 23, 128)	147584	batch_normalization_9[0][0]
dropout_3 (Dropout)	(None, 23, 23, 128)	0	conv2d_7[0][0]
global_average_pooling2d_1 (Glo	(None, 32)	0	dropout_1[0][0]
global_average_pooling2d_2 (Glo	(None, 64)	0	dropout_2[0][0]
global_average_pooling2d_3 (Glo	(None, 128)	0	dropout_3[0][0]
concatenate_1 (Concatenate)	(None, 224)	0	global_average_pooling2d_1[0][0] global_average_pooling2d_2[0][0] global_average_pooling2d_3[0][0]
batch_normalization_10 (BatchNo	(None, 224)	896	concatenate_1[0][0]
dense_1 (Dense)	(None, 256)	57600	batch_normalization_10[0][0]
dropout_4 (Dropout)	(None, 256)	0	dense_1[0][0]
batch_normalization_11 (BatchNo	(None, 256)	1024	dropout_4[0][0]
dense_2 (Dense)	(None, 256)	65792	batch_normalization_11[0][0]
dropout_5 (Dropout)	(None, 256)	0	dense_2[0][0]
dense_3 (Dense)	(None, 28)	7196	dropout_5[0][0]
activation_1 (Activation)	(None, 28)	0	dense_3[0][0]

Total params: 597,388    Trainable params: 595,268    Non-trainable params: 2,120

並以此模型架構做參數調整、包含將 kernelsize 設定與其他 dropout 的數值調整，另外建置出八個基礎模型，進行 40 次 epoch 訓練。下圖為模型在訓練過程中 lose 與 F1 數值變化：

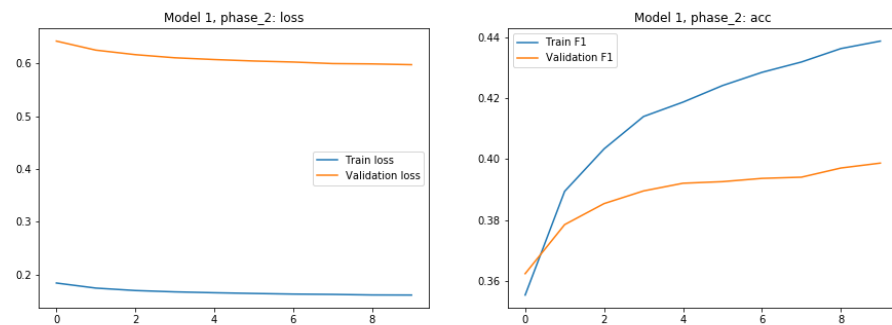




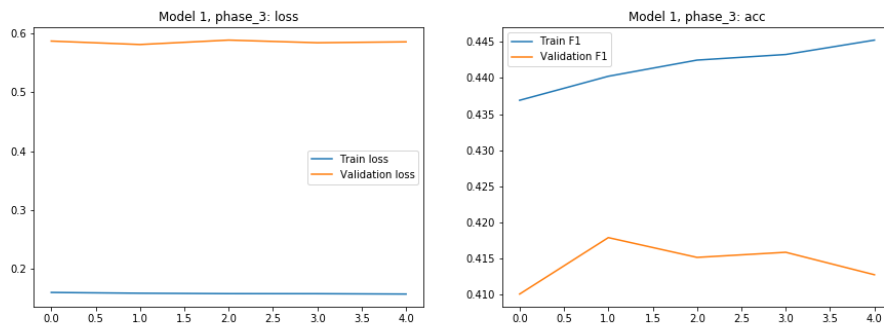
圖一、Base model 0-4

Train lose、Validation lose、Train F1、Validation F1 變化過程

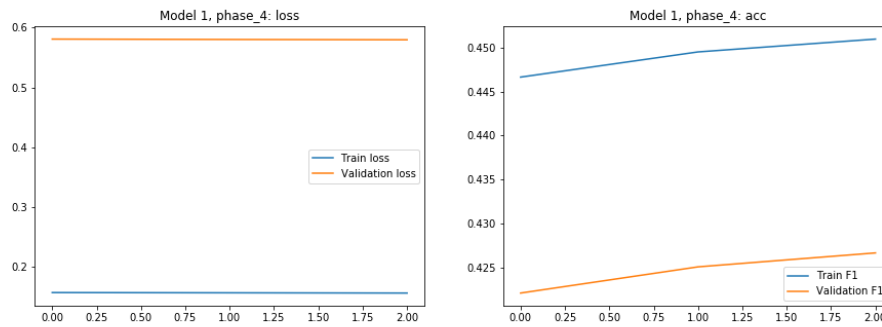
從圖片中可發現，原本的模型在經過四十次 epochs 後 Validation F1 大約落在 0.3-0.35 之間，而且有些還有上升趨勢。所以本報告後續用 pretrain model 用微調(Fine-tune)的方式，對 CNN 與 DNN 部分進行重新訓練，另外產生 24 個 model，兩者過程分別如下：



圖二 Base model 1 pretrain 後用 CNN 繼續訓練 10 個 epochs 後  
Train lose、Validation lose、Train F1、Validation F1 變化過程



圖三 Base model 1 pretrain+CNN 後用 DNN 繼續訓練 4 個 epochs 後  
Train lose、Validation lose、Train F1、Validation F1 變化過程



圖四 Base model 1 pretrain+CNN+DNN 後用 DNN 繼續訓練 2 個 epochs 後  
Train lose、Validation lose、Train F1、Validation F1 變化過程

從圖片趨勢中可以發現，後續訓練已經有一些 overfitting 的現象，不過其 validation F1 依舊有些許上升。最後本報告依照試驗後採用 model 0, model 1, model 2, model 3, model 4 與 model 6，權重部分先是參考 Validation F1 的數值（約在 0.42-0.47 之間）後並進行平移(-0.34)，最後依照得到的權重比用 ensemble 的方式進行預測。在 kaggle 上得到 public score 0.448 與 private score 0.423 的成績。

25 submissions for NTU\_r06521504\_隊名我想想 Sort by Most recent

**All** Successful Selected

Submission and Description	Private Score	Public Score	Use for Final Score
<a href="#">4channels_cnn_from_scratch.csv</a> a few seconds ago by Yu-Jen Chen <a href="#">add submission details</a>	0.423	0.448	<input type="checkbox"/>

## Experiment and Discussion

在最後 ensemble 的部分，從之前測試中可發現平移權重大小其實對最後數據相差極高。以 0.34 與 0.36 做平移值，其相差的 public score 可以差到 0.02。

另外，我們也曾經將 32 個模型中前期末經過 CNN 與 DNN 的模型一併放入 ensemble，在權重都為 1 的狀況下，雖然 public score 並沒有比較高 (0.428)，但是後來 private score 出來後可發現其數值較其他模型的 private score 要高(0.428)，推測此部分是因為權重未調整到最佳數據，如果進行調整，或許可以得到更高的分數。

## Conclusion

本次報告我們先用 CNN 與 DNN 的模型進行訓練，並使用 ensemble 的方式加強模型強度，最終得到不錯的成果。

## Reference

1. X. Zhao, Wu, Y., Song, G., Li, Z., Zhang, Y., and Fan, Y., "A deep learning model integrating FCNNs and CRFs for brain tumor segmentation," *Medical image analysis*, 43, 98-111, 2018
2. Jonathan Long, Evan Shelhamer, Trevor Darrell; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440
3. Michal Haltuf, "CNN 128x128x4, Keras from scratch [LB 0.328]," *kaggle.com*, Oct. 30, 2018. [Online]. Available: [https://www.kaggle.com/rejpalcz/cnn-128x128x4-keras-from-scratch-lb-0-328?fbclid=IwAR2SbpFcM0WhZfs7l9xelukyE6e9bMoUKX\\_kjxo3fUhaFeS2UUM6QQ2LpT8](https://www.kaggle.com/rejpalcz/cnn-128x128x4-keras-from-scratch-lb-0-328?fbclid=IwAR2SbpFcM0WhZfs7l9xelukyE6e9bMoUKX_kjxo3fUhaFeS2UUM6QQ2LpT8). [Accessed Dec. 14, 2018].
4. Michal Haltuf, "CNN 128x128x4, Keras from scratch [LB 0.328]," *kaggle.com*, Nov. 4, 2018. [Online]. Available: <https://www.kaggle.com/rejpalcz/gapnet-pl-lb-0-385>. [Accessed Jan. 13, 2019].