

# 隊名：劉 X 鵬程萬里

隊員：

電信碩一 詹鈞皓 R06942141

電機碩一 劉兆鵬 R07921052

生機碩二 王凱陞 R06631035

題目：

## Human Protein Atlas Image Classification

### 壹、 Problem study:

針對我們所選的題目，由於資料為 512x512x4，不論是進行 image augmentation 或是 training，都會相當消耗資源，因此必須以適當的 batchsize 進行資料的批次處理，接著在圖片的讀取上，提供的資料是將圖片的 4 個 channel 分開存取，因此在讀檔的時候需要將 4 個 channel 的圖片進行疊圖，而此題目最有趣也是最具挑戰性的部份，即是在一張圖片中可能會有多個 labels 存在，且 28 種 labels 有一定程度的 Imbalance 存在，因此在 problem study 上我們將著墨在以下 3 點，先在 kaggle kernels 中參考一些有效的方法，並進行相關的 research，以達到更佳的结果。

#### 1. data Imbalance

由於資料的 data imbalance 頗為嚴重，在訓練資料中懸殊的 labels 量會導致 training model 的時候，會因為過度偏向數據較多的一方，而沒有完整得考量到所有 labels 的分佈情形，此時若 testing data 的資料分佈不太相同，則會導致預測結果變得很差，而 kaggle kernel 中，較為常見的方法有針對少數 labels 的圖片進行 image augmentation 或是使用不同的 loss function，同時選擇正確的 evaluation matrices 也會對於結果有很大的影響 [4]，針對 data imbalance 的探討，Rushi Longadge, etc. [5]整合了一些常用的處理方式，under sampling 和 oversampling 都是值得嘗試的作法，同時亦可以將 under sampling 和 image augmentation 一起使用，將較為少數類別的資料增加，此外 focal loss [6]的作法適用於 data imbalance 的 model training，透過調整計算權重的方式，改善過度專注於訓練多數類別資料的狀況。

## 2. multilabels case

在傳統的物件分類中，主要都為找出圖片中機率最大的物件為何，而在輸出分類層的部分則會透過 softmax activation function 將各類別所輸出的值根據各類別中所佔的比例轉換為機率值，之後在預測時只需要透過 argmax 則能夠選出最大機率的類別則能夠找出圖片中的物件。然而，在 multilabel case 中，使用 softmax 的話則會導致圖片中可能有出現該類別但因為轉換成機率後沒有超過原先制定的 threshold 則會忽略掉此類別。所以在 multilabels case 中，我們會將最後一層的 activation function 轉變成使用 sigmoid [7]，將各類別所輸出的值透過 sigmoid 轉換成 0 與 1 之間的值，接著只要將各類別所輸出的跟去我們所定義的 threshold 來判定此圖片中是否包含到此物件。另外，近年來也常常有研究者開始往不同的 multilabel model 研究，如 [8] 則透過全卷積層模型進行圖片的特徵萃取工程，而有別於以往所使用的 sigmoid activation function 將值轉換至 0 與 1 之間的方法，此方法在最後一層時使用 1x1 的 inception unit，將模型的最後一層的輸出 channel 投射到與類別數目相同，也就是將原始圖片中的各類別，透過全卷積層將圖片中各物件都投射到各自的 feature map 中，接著只要透過最後一層中各類別的 feature map 計算各類別的分數，則能夠辨識出此圖片中是否包含著以下的物件。另外，也有研究者使用使用 sub-concept [9] 的方式來辨別物件中有包含哪些類別，主要方法為使用全卷積層將圖片的特徵進行萃取，接著使用圖片經過卷積層後所降維的 feature map 經由 sub-concept model 將 feature map 進行維度轉換。假設圖片中就是包含著固定數量的物件，透過 sub-concept model 能夠對每一個假設的物件進行分數的計算，進而找出圖片中有包含哪些類別。

## 3. how to define threshold

由於上文提到，我們所訓練出的 model 會面臨 multilabels, multiclass 的挑戰，而為了要提升 classifier 的精確度，要如何制定精確的 threshold 就是一門學問。最簡單制定 threshold 的方法毫無疑問就是將所有類別的 threshold 統一設為 0.5，超過 threshold 的機率就判定為 True，反之則為 False，但在這個競賽中，data imbalance 的影響甚為巨大，這將會使這簡單的機制失去效用，因為在 imbalance 的資料組中，有較不常出現的類別是不太可能超過 threshold 為 0.5 的這堵高牆，這會導致某一不常出現的類別都不出現的情況，所以我們從 Kaggle 上的 kernel 著手研究，再找出討論在 multilabel 和 data imbalance 的情況下如何選出 threshold 的論文，進

行研究。

在研究 Kaggle 上的 kernel 後，定義 threshold 的方法有很多種，像是上述提到最簡單的：定義全部類別的 threshold 為 0.5，或是利用 validaion data 來去微調，測出在哪種 threshold 下所得到的 validation score 會最高。除此之外，另一種方法則是去調整每個類別的 threshold[3]，可以根據每個類別出現機率的分布來制定，又或是單獨依照 R, G, B, Y 四個 channel 分別制定 threshold 再 ensemble 起來得出最終的 threshold 數值。

然而，在 multilabes classifier 下制定 threshold 的領域中，已經有眾多學者做過研究，其中以 Binary Method[1]表現得最好，在多個類別的情況下去針對每個類別設計一個 decision function，並根據此 function 來制定最終的 threshold。此外，也有許多文獻記載 threshold selectiong 的議題與比較[2]，他們羅列了近年來學者大多使用甚麼樣的演算法制定 multilabels 的 threshold，並且自己準備許多 multilabels 的資料來做實驗，實驗中會定義好使用的演算法，驗證標準來去分析 threshold 的好壞，在經過一連串的 survey 之後，就可以挑選出最符合我們這個問題的演算法，制定出最好的 threshold。

## 貳、 Proposed method:

在前文中提到了幾個已知的重點問題，我們擇擇透過查詢 kaggle kernel 的方式找到已經測試且有效的作法進行參考，並將可行的方法進行實作，以下我們將目前所實作的方法和預期使用的方法分為四項進行說明：

### 1. data Imbalance 處理方式

目前我們所上傳的模型還未進行 data imbalance 的處理，目前有嘗試加入 focal loss 的功能，但尚未在 validation data 上得到理想的效果，接著我們將會嘗試 under sampling 的方式，由於少數類別的資料量明顯是不足的，因此加也將上 image augmentation 增加少數類別資料的數量，藉此增加可以訓練的資料量，同時我們也將測試是否可以透過多次 resample 的資料進行訓練並進行 ensemble，藉此減少 data imbalance 所造成的影響。

### 2. data preprocessing

在圖片的預先處理部份，目前加入了基本的 image augmemtation，如平移、旋轉，在圖片的尺寸上最一開始我們採用 256 x 256 的圖片大小，但經過測試之後發現 512 x 512 的大小會有較好的表現，除此之外我們將參考了 Alexander Liao [10]所

做的一些 image augmentation 的結果，加入較能產生不同色彩空間的圖片進行預處理，增加資料的數量和彌補資料不平衡的狀況。

除了上述所提到的資料不平衡的問題以外，我們也發現在圖片中有些圖片經過疊合之後會有過暗的問題，對此我們預計將把這些照片挑出，目前計劃透過照片的亮度平均值進行挑選，並將挑選出過暗的照片透過 histogram equalization 或其他明暗度、對比度的調整方式進行調整，藉此將圖片的影像品質盡可能地做到一致。

### 3. Model

目前我們所採用的模型為基本的 Convolutional neural network 架構，而有別於一般的 Batch normalization 的部份，我們是針對每一層 channel 的部份進行 normalization 而非對 batch 維度進行，是因為我們認為此次的圖片為採用 4 個 channel 作為輸入，然而這四個 channel 的資訊並非傳統的光源顏色資訊，而是透過染料顏色所呈現的資訊，所以在原始資料中，圖片的各項 channel 是沒有相關性的，故我們會採用對 channel 的部份先做 normalization 的步驟。此外，因為對於四個顏色的圖片是能夠對於不同物件偵測而成，故我們在模型中也會先分別對四個 channel 的顏色進行 convolutional 的運算，之後再將各 channel 所計算後的 feature map concatenate 起來繼續往下訓練。而目前此作法在 kaggle 上能夠 0.320 的分數。雖然沒有到很高的分數，但是在沒有做任何前處理以及 pretrained model 的情形下，也許目前的模型是朝正確的方向進行。

另外，根據 kaggle 上的 kernel 所教學，使用 pretrained model 的方法能夠獲得更高的成績，原因是因為使用 pretrained model 的 weight 是已經學習過無數張圖片的特徵，所以將之轉換到此任務上也能夠使模型叫快收斂，然而，因為這次的圖片與 pretrained model 所預先訓練的圖片資訊相差甚遠，故在平常一般的 transfer learning 都會將 pretrained weight 設定為 non-trainable 的，但是因為在這次比賽中的圖片較為不同，故我們會將 pretrained weight 設定為可以計算訓練的。使用 pretrained 的目的是只為了讓模型能夠有好的初始值，而透過新的圖片訓練學習讓模型能夠更加的專注在這次比賽中圖片的特徵架構。

然而一般所採用的 pretrained model 都為使用生活中的圖片內容，如 VOC, ImageNet 等圖片集的圖片都是為 3 個 channel (rgb)的圖片，但是我們這次所使用的輸入圖片為 4 個 channel，故直接採用 pretrained model 來訓練是不可行的。故我們將使用 Resnet34 作為 pretrained model，但我們會將此 model 的第一層 CNN layer 的 weight 取出來，並將此層的 CNN 中的 previous layer 改為 4 個 channel，

並將原先 pretrained 的 rgb weight 設定至新的 CNN layer 中，但因為 pretrained weight 只有三個通道，故於新的 CNN 的 Y channel 中我們會將 pretrained weight 中的 r 與 g 做平均後放入至新的 CNN layer 中的 Y channel，使得此 Y channel 也能夠在好的初始值下進行訓練。

#### 4. threshold learning

在這個競賽中當我們要做 predict 時，往往會需要用到 Test Time Augmentation(TTA)[10]的技術，他就是在 Validation data 或是 testing data 也隨機作 augmentation 的技術，像是旋轉圖片、翻轉圖片等等，使用這項技術有助於提高我們預測的精確度，既然對預測精確度有影響，當然對於 threshold 的制定也有很大的影響。

我們這組在對訂定哪組 threshold 有許多研究，嘗試了統一所有類別的 threshold 數值為 0.5，因為是最簡單的制定策略，所以在 Leaderboard 上僅得到 F1-macro 0.27 的分數，再這種策略上，我們還可以浮動制定 threshold 的數值並不一定只是 0.5 為分水嶺，但是因為這個競賽要預測的是 multilabels，所以我們覺得對於每項 class 都定義一個專屬於他們的 threshold 是比較合乎邏輯的，所以而我們目前使用制定 threshold 的方法是依照每一個類別的分佈情況來制定的，出現愈多次的類別，相對地 threshold 數值就要比較高，比較難跨越，而出現較少次的類別，就給定較小的 threshold 數值，在應用了這個制定策略後，我們在 Leaderboard 上的 F1-macro 分數從 0.27 進步到了 0.32，代表我們制定的這個策略是有用的。

然而，除了制定靜態的 threshold 之外，我們應該也要能學習動態制定 threshold 的方法，像是由 K. Nigam 等人所提到的 Binary relevant method，或是按照 validation score 來去動態調配出最適合的 Threshold 數值。

#### 參、Reference:

1. K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, <https://link.springer.com/article/10.1023/A:1007692713085>
2. Rong-En Fan and Chih-Jen Lin\*, A Study on Threshold Selection for Multi-label Classification, <https://www.csie.ntu.edu.tw/~cjlin/papers/threshold.pdf>
3. Iafoss, pretrained ResNet34 with RGBY (0.460 public LB), <https://www.kaggle.com/iafoss/pretrained-resnet34-with-rgbby-0-460-public-lb>

4. Michal Haltuf. Input preprocessing for fast data loading,  
<https://www.kaggle.com/rejpalcz/best-loss-function-for-f1-score-metric>
5. Rushi Longadge, Snehalata Dongre, Class Imbalance Problem in Data Mining Review, <https://arxiv.org/pdf/1305.1707v1.pdf>
6. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár.  
Focal Loss for Dense Object Detection  
<https://arxiv.org/pdf/1708.02002.pdf>
7. Andre F. T. Martins, Ramon F. Astudillo.  
From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification.  
<http://proceedings.mlr.press/v48/martins16.pdf>
8. Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, Jianbin Jiao.  
Weakly Supervised Instance Segmentation using Class Peak Response.  
<https://arxiv.org/abs/1804.00880>
9. Ji Feng, Zhi-Hua Zhou.  
Deep MIML Network.  
[https://cs.nju.edu.cn/zhoush/zhoush.files/publication/aaai17deepMIML.p  
df](https://cs.nju.edu.cn/zhoush/zhoush.files/publication/aaai17deepMIML.pdf)
10. Alexander Liao, Image Augmentation Demo with albumentation  
[https://www.kaggle.com/alexanderliao/image-augmentation-demo-  
with-albumentation](https://www.kaggle.com/alexanderliao/image-augmentation-demo-with-albumentation)