Report
Humpback Whale Identification Challenge
r06922086_feijai

1. Introduction & Motivation

這次的工作是藉由座頭鯨的尾巴辨認出現在是哪種座頭鯨。

訓練資料的影像：9850 張彩色的鯨魚尾巴照片，總共有 4251 類別（4250 種類別 + new whale）。

測試資料的影像：15610 張彩色的鯨魚尾巴照片

評估的方法：MAP@5

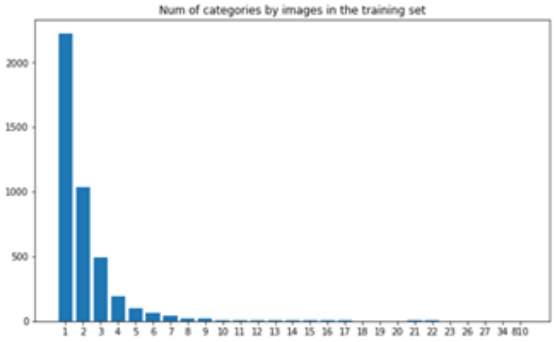
送出的預測每筆測項可以包含五種類別。

1.1. Data Analysis

首先我們先分析我們這次所拿到的資料。

將所有訓練資料中的依類別的數量建表，橫軸為影像的數量，縱軸為類別的數量。(圖一)

圖一



圖二

從圖中可以觀察到訓練資料中，大約有兩千多種種類，只出現過一張圖片，然然後約一千多種種類，只出現過兩張圖片，接著便快速的下降。這情況代表我們在訓練時，大部分的種類，我們的模型只會看過一次。

而有一種類別的影像特別的多，發現是 new whale 總共有 810 張照片。

再來我們觀察影像本身，影像有彩色也有灰階的圖片，解析度大部分為 1000 x 500。圖片雖然幾乎都是以鯨魚尾巴為主體，但還是有部分影像有把周圍的景色也囊括進去，包括水花、海洋、天空等等。因此我們想把這些背景去除掉，盡量專注於鯨魚尾巴的影像上。(圖二)

我們將所有圖片進行了分析，發現訓練資料與測試資料中，都有重複的影像在資料中。

由上述對資料的觀察，我們歸納出五點：

1. 這個問題本身屬於一個影像的分類問題，所以我們可以考慮使用 CNN 的模型來進行訓練與預測。

2. 基於這個資料集的特性，大部分的種類的影像的訓練資料不足，再加上每條鯨魚尾巴都非常相似。所以我們有參考他人的討論可以將這個問題對應到人臉的辨識問題上：每張人臉的非常相似，但是人臉的資料往往也只出現過一次就必須進行預測。

3. new whale 這個種類數量很多，並且很可能代表不同的鯨魚。因此我們推測這個種類在訓練時會影響模型的準確度，但是同時很高比例的資料是 new whale，所以最後在預測時，希望能盡量將 new whale 加到預測結果中。

4. 圖片的環境可能會在訓練時造成影響，所以希望在資料前處理時，把背景切除，只保留鯨魚尾巴的部分。

5. 有大量重複的影像出現，這部分除了影響訓練結果之外，也可以用在預測資料當中，我們將會進行實驗，觀察重複影像所造成的影響。

2. Data Preprocessing

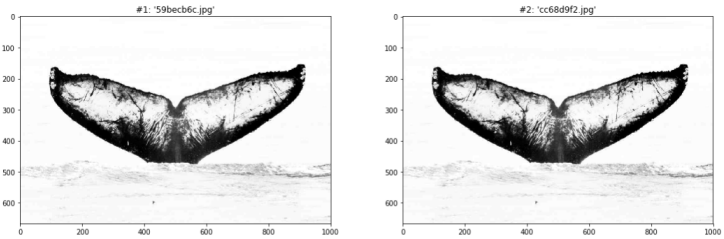
2.1. Duplicate Image

影像中，我們發現總共有 1841 張一模一樣的圖片出現在訓練與預測資料當中(圖三)，另外訓練資料中，也有 781 張重複的影像 [5]。為了處理這樣的問題，我們採取兩個作法。

1. 將重複的訓練資料移除，避免模型專注在重複的影像。

2. 實驗將最後測試資料中有出現在訓練資料中的標記換為訓練資料的標記，觀察對預測值的影響。

圖三



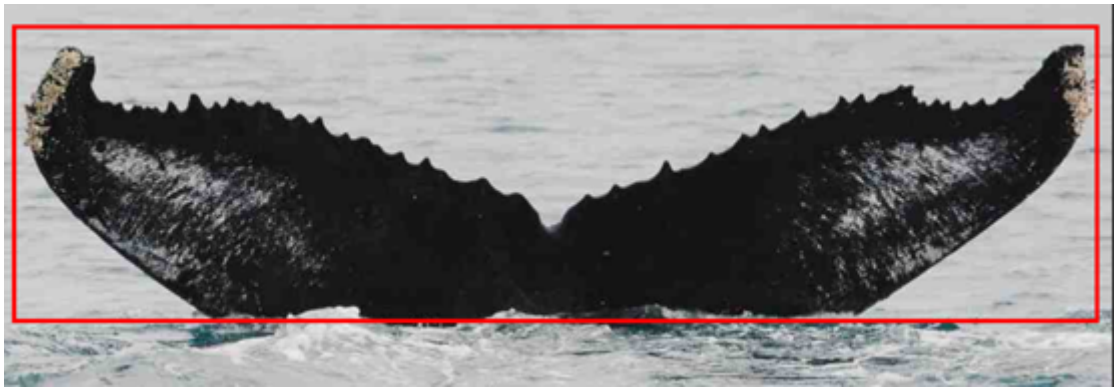
2.2. Bounding Box

雖然大部分的影像已經有很好的裁切，只保留鯨魚尾巴的部分，但還是存在著不少圖片，包含了背景的環境像是水花、天空、海洋等等。

為了去除這切除這部分的影像，我們想設計一個監督式學習的模型，去協助我們找出最逼近鯨魚尾巴的座標點，然後依據這些座標點裁切出我們要的影像。

首先我們取 1000 張圖片進行標記，繪出最接近鯨魚尾巴的方框。(圖四)然後將上下的水平分量跟左右的垂直分量記錄下來。然後建一個 CNN 的模型，輸入一張影像後，預測出上述的四個值。

圖四

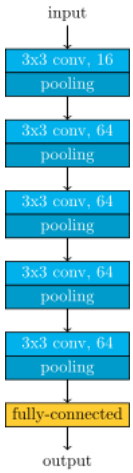


以下是我們 Bouding Box 訓練資料的說明：
我們的輸入為我們標註的影像，會調整為 256 x 128 的灰階影像，然後輸出則是我們標記的四個數值(如表一)。正規化處理則是將每張影像的 pixel 值除以 255 ，輸出值則是做標準差正規劃。

表一

Image	Bottom	Top	Right	Left
00022e1a.jpg	36	666	63	374
000466c4.jpg	231	580	303	442
00087b01.jpg	11	1037	25	360

圖五



接下來是我們的訓練模型：
我們參考了另一個鯨魚預測比賽中第一名所用的 Bounding Box 模型 [4]，
輸入之後接著有五層的捲積層，每層捲積層都採用 3x3 kernel size， 池化層則採用 2x2 kernel size，flatten 後接兩層 256 units 全連接層(圖五)。
loss function: mean-squared error
optimizer: adam

2.3. Refined Bounding Box

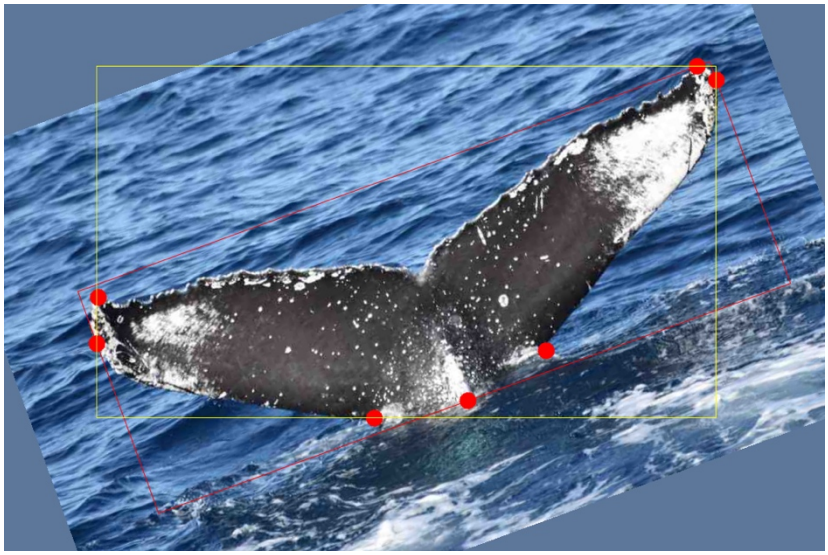
上個禮拜，Martin Piotte 在 Humpback Whale Identification Challenge 中發表了一篇 kernel [3] ，分享他如何實作 Bounding Box ，我們觀察的作法後，發現他的做法更加細緻，有兩點我們覺得特別可以借鏡的：

1. 他們的標記是直接標記在鯨魚的尾部邊緣，再去計算出整個 Bounding Box 的邊界值。

2. 採用 affine transformation 來增加訓練的資料量。

於是我們便進行了實驗，採用他們的標記來進行訓練，並用了 affine transformation 來產生新的訓練資料，affine transformation 僅用了旋轉正負二十度(圖六)。

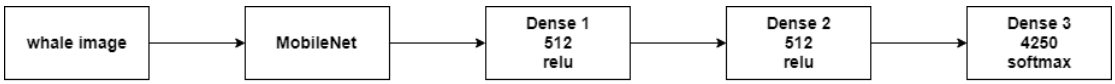
圖六



3. Model Description

我們的模型架構：Input 是 1 張鯨魚圖片，先經過 1 個 MobileNet，再經過 2 層 Dense，最後 Output 成 4250 種鯨魚類別的機率(圖七)，注意這邊是把 new_whale 從 training 的流程中移除，所以 Output 只有 4250 種類別的機率。

圖七



圖八

Type / Stride
Conv / s2
Conv dw / s1
Conv / s1
Conv dw / s2
Conv / s1
Conv dw / s1
Conv / s1
Conv dw / s2
Conv / s1
Conv dw / s1
Conv / s1
Conv dw / s2
Conv / s1
5× Conv dw / s1 Conv / s1
Conv dw / s2
Conv / s1
Conv dw / s2
Conv / s1
Avg Pool / s1
FC / s1
Softmax / s1

我們採用 MobileNet 的結構(圖八)，但我們只保留 CNN 和 pooling 的部分，去除 FC layers 改用我們自己定義的 FC layers

Dense 1 及 Dense 2 有包含 BatchNormalization 及 Dropout(0.6)，

Dense 3 只有 BatchNormalization。

Optimizer 是 Adam，loss function 是 categorical crossentropy。

訓練時使用來自 ImageNet 的 pretrained weights 作為 MobileNet 的 initial weights 並且不會凍結 weights，FC layers 則是用 glorot normal initialization。除此之外也加入 data augmentation，用到了水平翻轉、旋轉、垂直移動、水平移動，利用產生器在訓練同時一邊生成新圖一邊訓練。

在 output 後我們列出前 4 個機率最高的類別，最後再將 new_whale 放在第一順位，得到共 5 個類別，作為我們單一 model 的答案。
除此之外，我們會再 ensemble 來提高我們的成績。這次改為每個單一 model 要列出前 20 個機率最高的類別做為他們的答案，每個 model 再給予自己的答案權重分數，從機率最高的第一名得 20 分到機率最低的第 20 名得 1 分。每個 model 都做完後將大家的答案混合起來去算每種鯨魚的平均權重分數，取出最高的前 4 種類別，再將 new_whale 放在平均權重分數 19 的鯨魚前面一個順位，假如真的 4 種鯨魚的平均權重分數都在 19 分以上，那還是會將 new_whale 放在最後順位，以保證 new_whale 一定在答案之中，以上即完成 ensemble。

另外我們也參考了[2]裡所提到的 Triplet loss，Triplet loss 基本上是從 training data 中 sample 三個 items，其中兩個 items 是同一個 class，另一個是不同 class。並在 training 的過程中使得相同的 class 在 embedding 中會更靠近，不同 class 則會拉遠。Triplet loss 也有使用在 Google 的 FaceNet 中[8]。Triplet loss 主要是用在 One-shot learning，因為在這個 task 中，有許多鯨魚的影像只有一張的情形，因此這個 task 算是 One-shot learning。Model structure 用 ResNet50，並用 pre-trained weight，embedding dimension 為 50 維。Predict 時，我們利用之前 train 的 model 將 testing data 投影至 embedding space，並用 kNN 的方式找出離 testing data 最近的 5 個點，並用 0.1 當作 new_whale 的 threshold，也就是說，若 kNN 找出的點與 testing data Euclidean distance 小於 0.1 時，就將 new_whale 插入 predict 的順位。

4. Experiment and Discussion

4.1. Bounding Box

4.1.1. Data augmentation 的影響

我們利用 data preprocessing 中提到的 bounding box model architecture，並比較有無 data augmentation 後 bounding box 結果。我們利用 affine transformation 做 data augmentation，並只做 rotation，我們分別做了從-20 度到+20 度間隔 10 度的 augmentation，意即影像角度為-20, -10, 0, +10, +20，以及從-10 度到+10 度間隔 5 度的 augmentation，意即影像角度為-10, -5, 0, +5, +10，因此 augmentation 皆為 5 倍的 training data。training data 共有 1200 筆，augmentation 後為 6000 筆，10%的 training set 當作 validation set，皆 train 500 個 epochs，並用 validation loss 最低的 model 進行比較。各 model 資訊如下表：

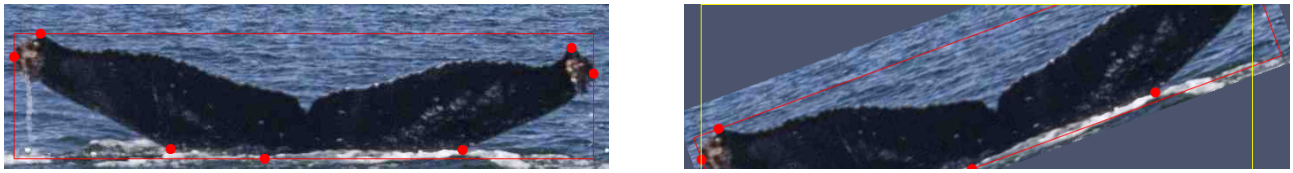
model 名稱	Train set 大小	Validation set 大小	Training loss	Validation loss
w/o_aug	1080	120	0.003974	0.004545
w/_aug_10_degree	5400	600	0.001195	0.000955
w/_aug_20_degree	5400	600	0.001540	0.001205

以下比較 bounding box 的結果

(由左至右依序為：原圖、w/o_aug、w/_aug_10_degree、w/_aug_20_degree)

0fb4c4dd.jpg

此影像 w/_aug 的結果比 w/o_aug 的結果好，w/o_aug 的影像完全沒抓到鯨魚尾巴的位置，而旋轉 20 度也稍好於 10 度。可能是因為 rotation 後的影像變多都會使鯨魚尾巴的位置旋轉至角落的位置，如下圖，因此在 predict 位於角落的鯨魚尾巴便能辨識出來。



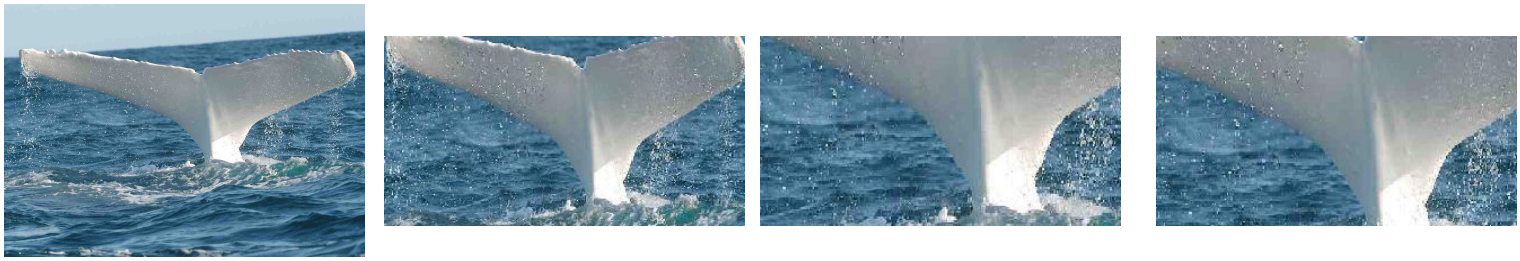
2d3967c1.jpg

在此影像中反而旋轉 10 度的結果最好，在 w/o_aug 影像中，海水還是佔大部分，而旋轉 20 度的影像中，尾巴靠近身體的部分被裁切掉了。

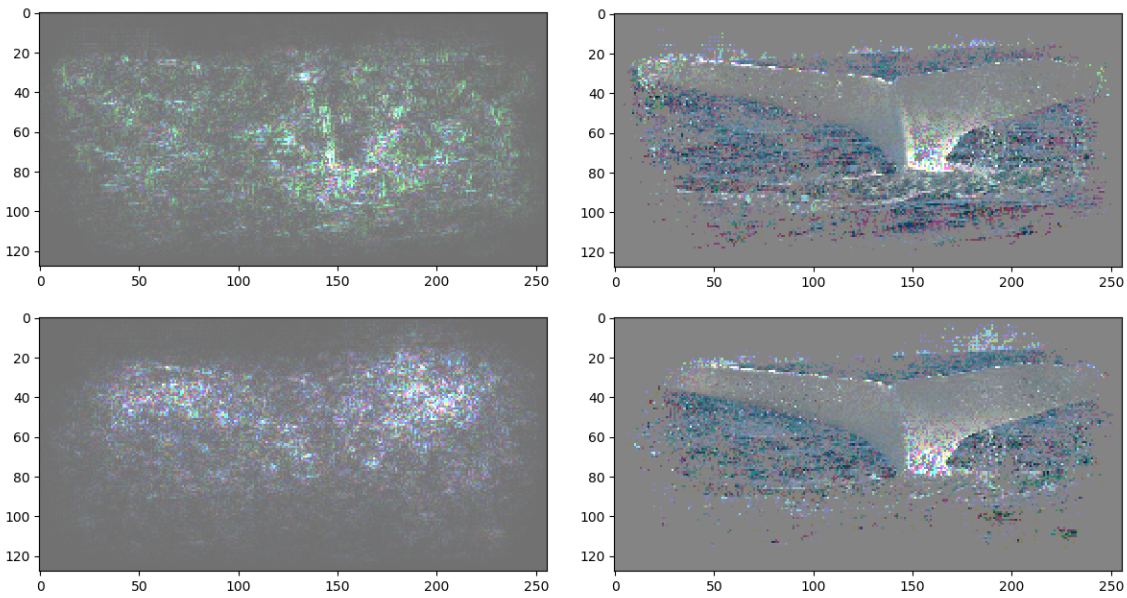
2d3967c1.jpg

此影像同樣是旋轉 10 度的結果最好，從上述兩個例子可以看到，當原圖的鯨魚尾巴在正中央時，w/o_aug 雖然沒辦法準確框出尾巴位置，但至少尾巴還在影像中。但旋轉 20 度的尾巴反而都有部分被裁切掉。可能是因為當每張影像都旋轉做 augmentation 時，旋轉的影像即佔了 4/5 的 training data，導致 model fit 到歪斜的影像，在一般的影像便失去準確度。

7fdde4d6.jpg



此影像是一個特別的例子，沒有做 augmentation 的結果竟然最好，因為全白的鯨魚尾巴不常見，dataset 裡可能沒幾條全白的尾巴。只能猜測是做 augmentation 的 model overfitting 了，導致在碰到沒見過的尾巴時失去辨識能力。我們分別做了此影像在 w/o_aug 及 w/_aug_20 的 saliency maps 以及 filter visualization。上面 2 張為 w/o_aug、下面 2 張為 w/_aug_20。

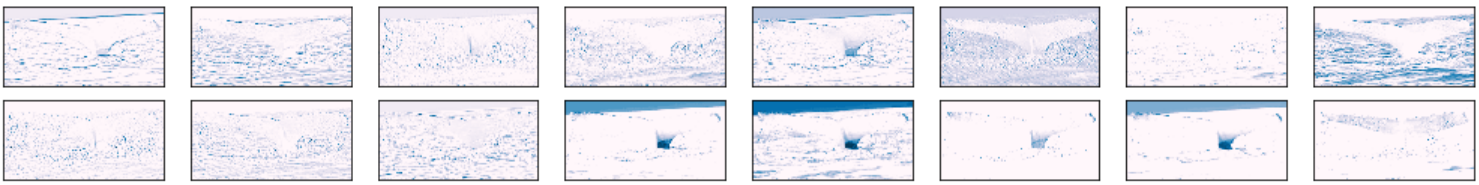


可以發現 w/o_aug 看到尾巴的範圍稍微較大。

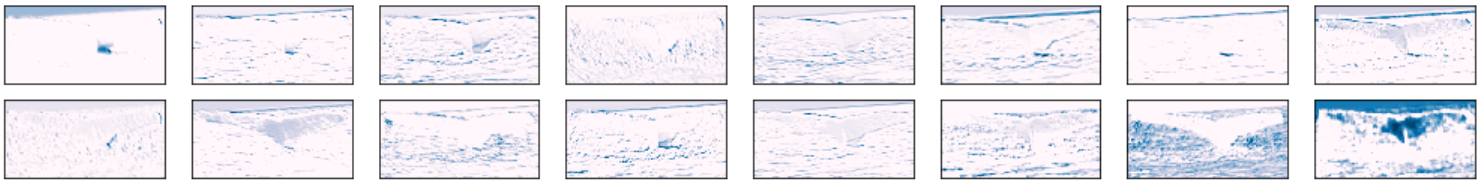
接下來比較 filters 所看的的內容，觀察後發現 CNN 前三層的 output 差異不大，前三層只附上 w/o_aug 的 output。前三層主要捕捉細部的 texture，第一層 layer 主要捕捉海洋的區域以及天空的區域，值得注意的是捕捉天空的 filter 同時也抓到尾巴根部的區域，可能是天空的 filter 捕捉一整片相同顏色的 texture，而在這一個影像中尾巴根部的區域

也符合這樣的 texture。在第二、三層 layers 已經能捕捉到鯨魚尾巴的位置了。

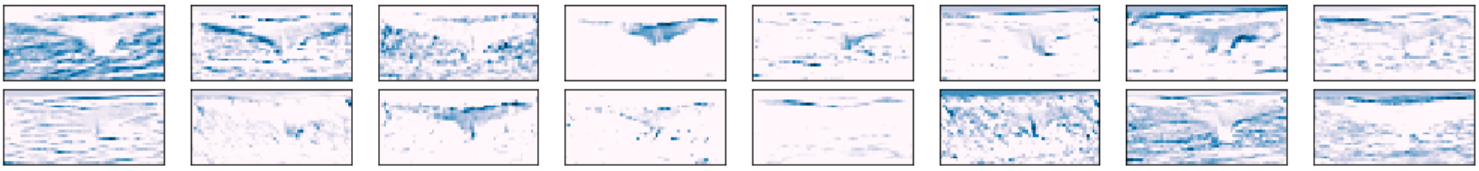
w/o_aug_layer_0



w/o_aug_layer_1

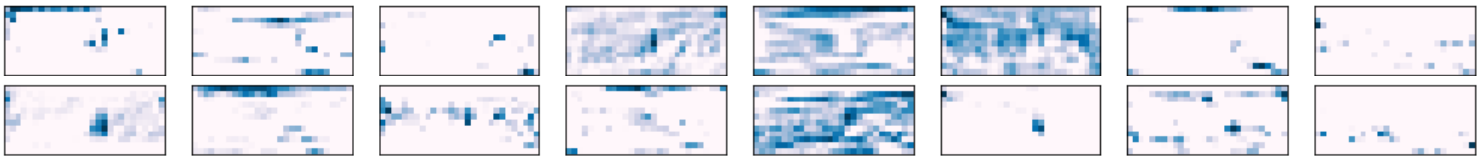


w/o_aug_layer_2

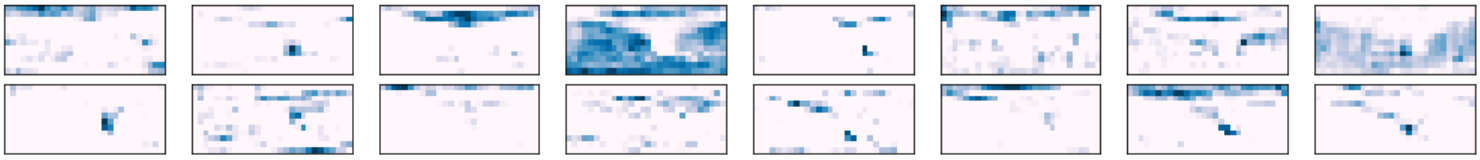


以下比較後二層的 output。這兩層已經沒有在捕捉細部 texture，而是關注在尾巴周圍的一些特徵，特別的是兩個 model 都有許多 filters 關注在尾巴根部的區域，可見尾巴根部是 bounding box 重要的 feature。

w/o_aug_layer_3

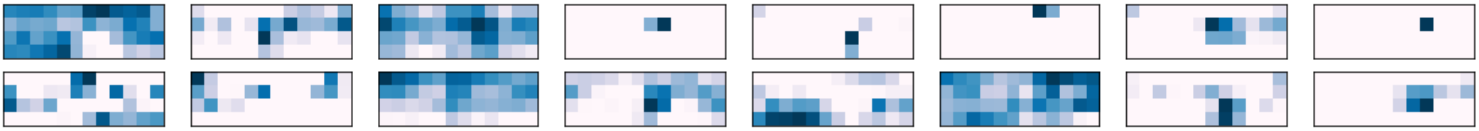


w/_aug_20_layer_3

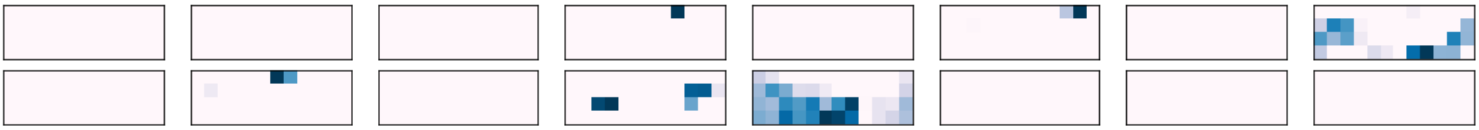


最後一層 CNN layer 的 output 中，明顯看出 w/_aug_20 多數的 filters 都是 unactivated，而 w/o_aug 的 filters 已經能把尾巴周圍的 features 給捕捉出來，因此結果比較好。

w/o_aug_layer_4



w/_aug_20_layer_4



4.2. Classifier

4.2.1 new whale 的影響

在訓練過程中是否加入 new_whale 資料，沒有加入的話會在 output 後，強制加在預測結果中且在預測第一順位。

	Public score
有 new_whale	0.36595
沒有 new_whale	0.53043

可以看到差距十分明顯，應該是因為 new_whale 不是一個真正的鯨魚品種，new_whale 中夾雜太多種鯨魚，所以影響了 classifier 判斷能力。

4.2.2. data augmentation

data augmentation，用到了水平翻轉、旋轉、垂直移動、水平移動。

	Public score
有 data augmentation	0.53043
沒有 data augmentation	0.46935

在分數上也有顯著差異。

4.2.3. rgb or gray

原始圖片中是由 rgb 和 gray 混合組成的，比例如下表：

	gray	rgb
train	0.193048	0.193833
test	0.334957	0.278162
total	0.528005	0.471995

上圖中的比例是指該部分的圖佔所有圖片(train + test)的比例。

而下表是將所有圖片轉成 rgb 或 gray 的成績

	Public score
rgb	0.53089
gray	0.53043

從數據上來看差異不大。

4.2.4. 圖片大小

原始圖片本身就長寬大小不一，訓練需要統一長寬，且 MobileNet 要求 input 必須是正方形，因此才變形成正方形。

	Public score
128*128	0.46138
160*160	0.46935

我們是採用早期的兩組 training 結果來做比較，因為在知道圖片長寬較大表現會較好之後，training 都採用 160*160 的圖片資料，唯一與 128*128 有其他相同實驗條件的是表上這一組 160*160，為了公平起見才沒有採用其他更好表現的結果。

從表上的數據來說，兩者之間有些微差異，應該是縮小圖片時消減了一些細節。

4.2.5. 塞入標準答案的影響

有一些 testing data 的圖片與 training data 一致，因此可以直接照抄對應的類別答案。

	有照抄的 public score	沒有照抄的 public score
Sample submission	0.40954	0.32453
Our final ensemble	0.60864	0.60587

可以看到當我們的 model 越準確越強時能夠進步的幅度會更小。

5. Conclusion

最後我們 Kaggle 上的 public score 為 0.60864， 排名 5 / 494 。

我們稍微歸納了一下為什麼我們的表現還不錯：

1. 用了作 Bounding Box 的影像比起原圖來說進步很多，採用原圖時訓練有可能無法集中在鯨魚尾巴上。而被外在背景影響，使得準確度降低。

2. 訓練時除去 new whale，直到最後預測時才將 new whale 考慮進來，因為 new whale 是一個混合各種種類的類別，再訓練時，模型很容易遭受其影響，我們的實驗當中，是否有加上 new whale 類別去訓練，是差非常多的。

再來就是 new whale 的數量眾多，由 sample submission 可以得知，測試資料中每三筆資料可能就有一筆是 new whale，所以我們決定在輸出時利用 threshold 的方式加入 new whale。

3. MobileNet 這種輕量級的分類模型，在鯨魚分類的辨識上非常不錯，因為模型的參數不多，所以也不容易造成 overfitting 。

4. weight voting 的 ensemble 方式還不錯，因為我們本身是產生許多 CNN 模型去作訓練，最後讓每個模型能各自選出二十種鯨魚再透過權重挑出所有模型的大共識，由實驗可知，這樣的投票方式表現是好的。

上面幾點便是我們歸納出的總結，能在最後得到第五名，我們也非常的意外。

我們從這門課學到非常多，最後謝謝老師的教學與助教辛苦的批閱作業。

6. Reference

[1] Lex Toumbourou. Humpback Whale ID: Data and Aug Exploration. https://www.kaggle.com/lextoumbourou/humpback-whale-id-data-and-aug-exploration
[2] CVxTz. Beating the baseline – Keras (lb 0.38) .https://www.kaggle.com/CVxTz/beating-the-baseline-keras-lb-0-38
[3] Martin Piotte. Bounding box data for the whale flukes .https://www.kaggle.com/martinpiotte/bounding-box-data-for-the-whale-flukes
[4] Robert Bougucki, Marek Cygan, Machiej Kilmek, Jan Kanty Milczek, Marcin Mucha. Which whale is it, anyway? Face recognition for right whale using deep learning. https://deepsense.ai/deep-learning-right-whale-recognition-kaggle/
[5] TheGoose. Duplicate Images. https://www.kaggle.com/stehai/duplicate-images
[6] Keras Applications. https://keras.io/applications/
[7] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861
[8] FaceNet: A Unified Embedding for Face Recognition and Clustering https://arxiv.org/pdf/1503.03832.pdf