

Machine Learning Final Project Report - Whale

NTU_r06944051_學弟隊長就交給你了retry

B04902037 顏子斌、R05922080 王鵬傑、R05944034 鄭竣元、R06944051 郭柏辰

I. Introduction/Motivation

這個題目主要目標為，透過Happy Whale's Database訓練一個model，進行鯨魚的辨識。即給定一張含有鯨魚尾巴的圖片，Model必須預測出鯨魚的種類，最多可以標注5種鯨魚種類，若沒出現在training data中的label則標注為new_whale。

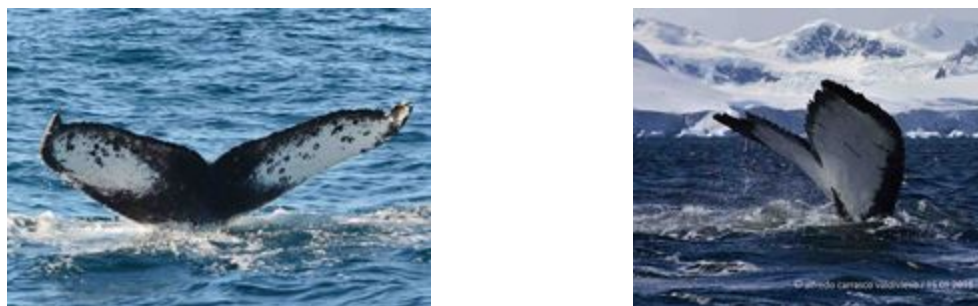


Fig. 1. Training Data 示意圖

此問題十分困難，由於資料集存在以下問題：

- A. 圖片大小不一，有的圖片解析度很高，有的圖片則很差。
- B. 圖片含有灰階圖及彩色兩種。
- C. 圖片方向不一，有些圖片有經過rotation。
- D. 訓練資料集非常不平衡，不同種類的鯨魚，資料量差異很大。如Fig1所示，有2000多類只有一張圖片。

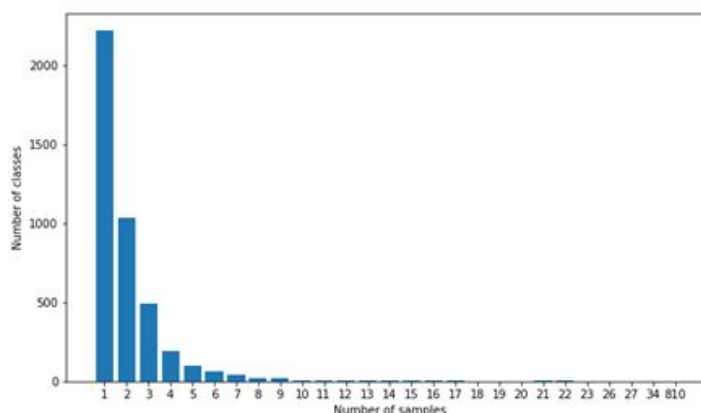


Fig. 2. Number of categories by images in the training set

主要問題為訓練資料集非常不平衡，大多數種類只有一張圖片。此種特色與face recognition很相似。在face recognition中，每個人都視為一類，而每一類都不會有大

量的samples，甚至只有一個sample，故我們嘗試使用應用在face recognition的方法—Triplet Network，來解決這個問題。

II. Data Preprocessing/Feature Engineering

A. Bounding box regression

為了獲得高品質的training data，我們希望所有圖片都能只專注於鯨魚尾巴的部分，因此我們打算訓練一個CNN來做bounding box regression，其訓練集則是採用Kaggle discussion所提供的599張人工標記好的座標[5]。

我們接著使用訓練好的model預測所有圖片的bounding box。由於599張圖片大小都不一樣，所以先將599張圖片利用linear scaling取得圖片大小為256x256所對應的bounding box座標，並將linear scaling後的bounding box座標當作training data。

CNN model如下：

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 256, 256, 64)	1792
conv2d_2 (Conv2D)	(None, 256, 256, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 64)	0
conv2d_3 (Conv2D)	(None, 128, 128, 128)	73856
conv2d_4 (Conv2D)	(None, 128, 128, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 128)	0
conv2d_5 (Conv2D)	(None, 64, 64, 256)	295168
conv2d_6 (Conv2D)	(None, 64, 64, 256)	590080
conv2d_7 (Conv2D)	(None, 64, 64, 256)	590080
max_pooling2d_3 (MaxPooling2D)	(None, 32, 32, 256)	0
conv2d_8 (Conv2D)	(None, 32, 32, 512)	1180160
conv2d_9 (Conv2D)	(None, 32, 32, 512)	2359808
conv2d_10 (Conv2D)	(None, 32, 32, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 16, 16, 512)	0
conv2d_11 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_12 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_13 (Conv2D)	(None, 16, 16, 512)	2359808
max_pooling2d_5 (MaxPooling2D)	(None, 8, 8, 512)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_1 (Dense)	(None, 4096)	134221824
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 4)	16388
Total params: 165,734,212		
Trainable params: 165,734,212		
Non-trainable params: 0		

Fig. 3. Summary of bounding box CNN model

訓練完後，用此model預測bounding box結果如下：



Original

Bounding box

Fig. 4. bounding box裁切training data示意圖

B. Data augmentation

我們將裁切好的image進行data augmentation，隨機執行rotation、shift、zoom、greyscale操作，來提升model的抗噪性與應對部分training data不足的狀況。

III. Model Description

我們所選用的model的想法啟發自[1]所使用的架構，該架構使用[3]所提出根據multi-class labeled dataset進行metric embedding的Triplet Network，架構圖如下：

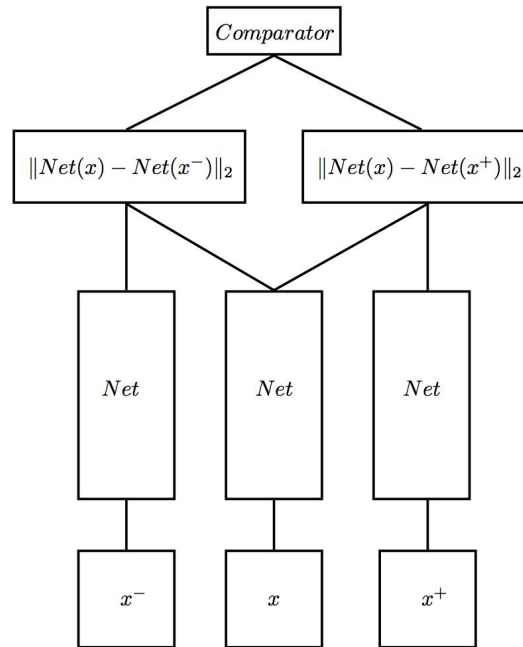


Fig. 5. Triplet network structure

如圖中所示，Triplet Network由三個feed-forward network所組成，input一次提供 x^- 、 x 、 x^+ ，其中 x^- 代表與 x 不同類的圖片， x^+ 代表與 x 同類的圖片。中間的Net負責將圖片進行embedding，將三張圖片分別做embedding後再計算他們之間的距離，最後根據Comparator的比較結果來判斷，而Comparator的實作參考自[4]。基於上述的架構，我

們讓圖片經過中間的feed-forward network後，再經過一層的MaxPooling和dense layer得到image embedding的結果。

A. Model 1: Triplet Network using ResNet50 as image embedding model

這個model中套用了ResNet50作為Triplet Network中間的Net，embedding的dense layer的dimension為1500。

B. Model 2: Ensemble of multiple Triplet Networks

在Model 1的結果開始遇到瓶頸後，改成使用像是VGG19, Xception, InceptionV3等其他經典model來作embedding，最後靠著整合不同model做ensemble後才得到了更多突破。

我們進行ensemble的方式是讓每個model對training set和testing set做embedding後取得各自的embedding results，透過sklearn套件中的NearestNeighbor取出離各筆test image的embedding result最近的10筆training data的embedding results當作可能的candidates；但如果new_whale不在這10筆candidates之中的話就再取估計值加入prediction中，這樣一來若是原本的10個candidates之中有的在NearestNeighbor的演算法中計算出的距離過遠的話，就會被new_whale所取代，這是我們因應無法透過training出來的model來判別new_whale的做法。最後把各個model所選出的10個candidates做voting，最高票的5個就是對該筆test data的prediction，這就是我們最後得出最好結果的model。

IV. Experiment and Discussion

A. Experiments on Model 1 (ResNet50)

1. Kaggle Kernel Triplet Model

在本次final project中，一開始我們是先嘗試使用在該競賽Kernel裡面其他參賽者所提出的Triplet model source code [1]。在對該份code進行修正與拿掉一些不必要的部分後，丟下去train與predict所得到的結果為0.38492，剛好略為超過simple baseline的標準。

2. 更換Embedding Model

在原本Kernel的source code中是採用ResNet50當作embedding的model。一個自然的想法就是更換embedding的model說不定效果會更好，所以我們另外嘗試了Keras 2.0.8所提供的InceptionV3與VGG19 model，結果如Fig. 6所示。我們發現新的model的效果並沒有比較好，所以繼續沿用原本的ResNet50當作embedding的model。

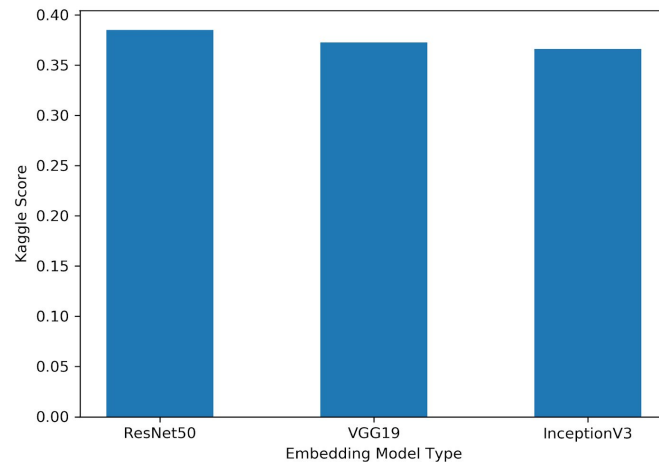


Fig. 6. Kaggle score of different embedding models

3. Training Data特性利用

根據前面的討論，我們可以發現本次final project的training data的分佈極度不平均，有相當高比例的training data的label皆為new_whale，所以我們希望能利用這個特性加強embedding的結果。由於我們的model的概念上是希望將各圖片embedding的結果在相同whale fluke時差異越小越好，在不同whale fluke時差異越大越好，所以我們嘗試將被標為new_whale的每筆training data都視為是一隻全新的whale，然後丟下去train和predict。不過結果顯示這樣效果反而變差，分數下降至0.37749。

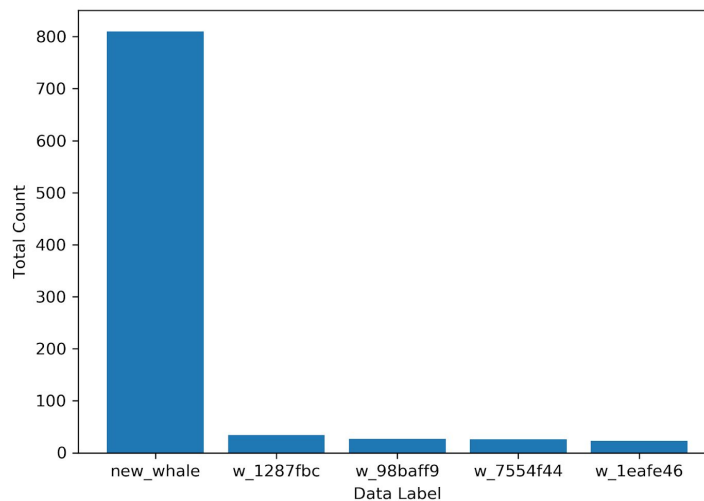


Fig. 7. Total counts of training data for each data label (showing only the highest five)

4. More Data Augmentation

根據前面的討論，我們可以發現有相當多的whale fluke只在training data中出現過一次，而這有可能會造成雜訊對我們model的影響上升，因此我們決定嘗試對training data做data augmentation來增加model的抗噪性。在原先kernel的code裡面對於data augmentation僅實作了機率性的左右翻轉，而我們在參考另一篇kernel上面的所提供的方法[2]後，另外增加了旋轉、平移、放大與轉灰階等作法，丟下去train與predict的結果有些微的進步，達到0.38685。

5. Embedding Dimension

在原kernel的source code裡面，embedding dimension是設定為50。因為embedding的效果實際上受embedding dimension的影響相當大，所以我們決定嘗試不同的embedding dimension並觀察其效果。在做過一點實驗後結果如Fig. 8. 所示，我們發現如果將dimension提高到約1500的時候效果最好，可以達到0.40822的成績。

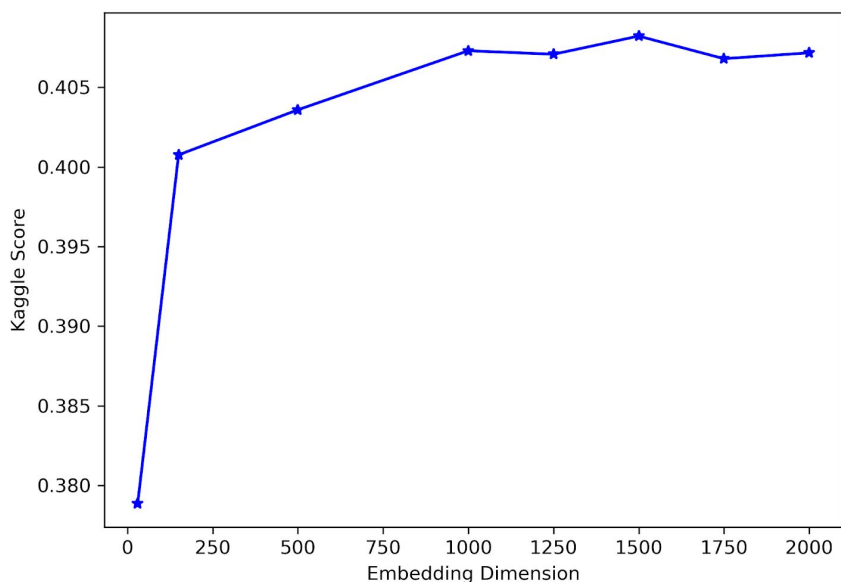


Fig. 8. Embedding dimension experiment result

B. Experiment on Model 2 (Ensemble)

在經過對model 1的實驗後，我們發現分數有卡在0.40左右而上不去的情況，因此我們決定尋求助教與其他組同學的幫助，而其他組同學與助教都建議我們可以試ensemble learning，所以我們就繼續朝這個方向做。我們嘗試撤換embedding model，train出另外三個分別使用InceptionV3、Xception與VGG19的model，然後與原本

的ResNet50一起predict並將結果做voting，上傳評分後發現效果相當好(0.46369)，所以就沒有繼續做實驗延伸下去。

V. Conclusion

本次Final project的Whale Identification其最大的困難點在於training data的資料有許多缺陷，例如每張圖的尾巴大小與圖片的比例不同、許多種類的鯨魚只有一張圖片等等，所以我們透過data augmentation來擴充training data，還有過去課堂上學過的CNN做bounding box regression來鎖定鯨魚尾巴在圖片中的位置，進而讓training可以更集中在尾巴的部分做特徵辨識，進而提高辨識的準確度。

而在辨識圖片的model上，我們使用現成的知名Image Recognition的架構來做image embedding，再使用[3]所proposed的Triplet Network，藉由它根據multi-class labeled dataset進行metric embedding的傑出效果，就可以根據判別test image和已知的labeled data的相似度來進行預測。將幾個不同model的預測結果進行ensemble後得到了0.46369的Accuracy。

References

- [1] CVxTz, Beating the baseline - Keras (lb 0.38),
<https://www.kaggle.com/CVxTz/beating-the-baseline-keras-lb-0-38>
- [2] Lex Toumbourou, Humpback Whale ID: Data and Aug Exploration,
<https://www.kaggle.com/lextoumbourou/humpback-whale-id-data-and-aug-exploration>
- [3] E. Hoffer and N. Ailon. Deep metric learning using triplet network. Similarity-Based Pattern Recognition, 2015.
- [4] maciejkula, Recommendations in Keras using triplet loss,
https://github.com/maciejkula/triplet_recommendations_keras
- [5] lisa needs braces, Bounding Box Labels,
<https://www.kaggle.com/c/whale-categorization-playground/discussion/57108>