Right Whale Recognition

Meng-Jiun Chiou, Li-An Yang, Kai-Sheng Yang
0110761, 0116314, 0116040
National Chiao Tung University

***Abstract*—In order to deal with computer vision and data mining techniques, we participate the Right Whale Recognition competition on Kaggle. After the introduction and some demonstration of several approaches adopted in this proposal, we move on to deliver the correspondent result and illustrate thoroughly about the entire paradigm.**

1. Introduction

This competition cope with the identification of endangered right whales in North Atlantic. In the description of the contest, it says: "*With fewer than 500 North Atlantic right whales left in the world's oceans, knowing the health and status of each whale is integral to the efforts of researchers working to protect the species from extinction.*" Therefore, this is not only a computer vision contest but also charitably meaningful.

For these endangered right whales, they will be labelled for tracking purpose. In this contest, Kaggle [1] and co-organizer NOAA (National Oceanic and Atmospheric Administration) provide datasets including both training data and test data. The former consists of about 4,500 different photos and the latter consists of about 5,500. Each photo consists of single right whale (or maybe head or body). What we have to do is answering the corresponding right whale's label in test data. In other word, to find out "who" this right whale is in the photos.

There are mainly two challenges to take care of in the competition. First, we have to locate the face part of the right whale to distinguish individuals. Secondly, after we acquire the location of the right whale in each photo, we can move on to train an image classifier that could be used to give the classification result of test data.

2. Whale face localization

In this chapter, we discuss some approaches of right whale location that may be used in our experiment. There're mainly three approaches we are going to take: 1) cascade object detector using Viola-Jones algorithm 2) R-CNN (Regions with Convolutional Neural Network Features) and 3) artificial labeling.

2.1. Cascade object detector using Viola-Jones algorithm

The cascade object detector uses the Viola-Jones algorithm [2] to detect people's faces, noses, eyes, mouth, or upper body. It could be easily implemented by using "Training Image Labeler App" in Matlab to train a custom classifier with this cascade object.

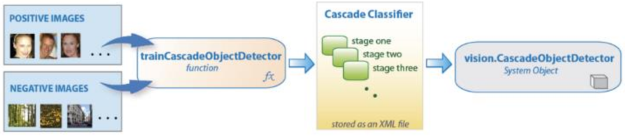


Figure 1: Process of cascade object detector

Because of its ease of use, we give it a try first. What we do is labeling some training samples and cropping negative samples, and use them to train a cascade object with tuning parameter such as the feature type (LBP, Haar or HOG), the minimum true positive rate in each stage (default: 0.995), the false positive rate (the fraction of negative training samples incorrectly classified as positive samples in each stage, default: 0.5) and the number of cascade stages (default: 20).

After training the classifier, we use it to detect the test sample to localize the right whale. However the result shows that the false positive rate is too high, and there is no confidence score for each detection, so we can't tune a threshold to remove those false positives. Plus, the detection appears to be sensitive to lighting condition, and the right whale photos photographed during aerial surveys would also cause some variation. Hence, we decided to move on to the second approach called R-CNN.

2.2. R-CNN

CNN (Convolutional Neural Network) is a type of feed-forward artificial neural network where the individual

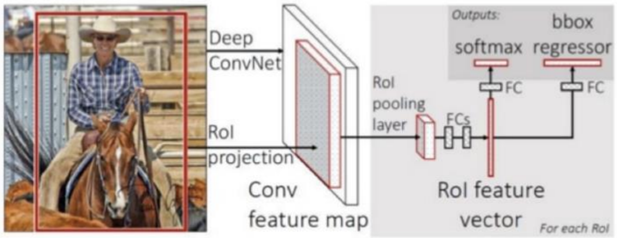


Figure. 2: Fast R-CNN projects ROIs onto feature map generated by pre-trained deep network and passes through pooling and fully-connected layers.

neurons are tiled in such a way that they respond to overlapping regions in the visual field. [3] Based on CNN, R-CNN [4] is a state-of-the-art visual object detection system that combines bottom-up region proposals with rich features computed by a convolutional neural network. Unlike the previous best results, R-CNN achieves this performance without using contextual rescoring or an ensemble of feature types.

It is achieved by classifying deep convolutional network features computed on each image region proposal (~2k), which means it should compute the deep convolutional network at the same times. It requires the input of R-CNN to be the determined size. As a result, the repeated computation makes the detection slow. To address this problem, Girshick et al. [5] propose Fast R-CNN which relaxes the restriction by projecting region proposal to a feature map created by pre-trained deep convolutional network (e.g. VGG16) and take advantage of multi-task loss technique. Projected proposal on feature map then passes pooling layers, fully-connected layers and finally output prediction boxes by solving regression problems and cross entropy (i.e. probability) by softmax classifier.

We implemented Fast R-CNN which is fine-tuned on VGG-16 convolution network [6] with 2,679 human-labelled test images. With the threshold set to 0.7, out of the total of 6925 images, 6196 images are detected with 652 images needed to be revised (false positive), meaning that this network achieve at an accuracy of 0.805.

2.3. Artificial labeling

No matter which approach we uses to detect the location of whale face, the accuracy must not be 100%; therefore, we still need to revise those wrong detections by ourselves.

While automatic labeling can make our job more easily, labeling all of the training sample by people may produce highly precise classifiers. As a result, we consider labeling all of the training sample (about 4,500 photos) by ourselves if needed. Additionally, considering orientation alignment would make the result of classification better, we have tried doing so by calculating the gradient and the rotation angle. The work is achieved by first acquiring the slope of the whale's moving direction by figuring out the linear regression of the pixels located on the whale. After that, we try to rotate the image according to the slope by the correspondent angle. However, the accuracy is less than half and deemed too low to use, so we eventually surrender to rotating and cropping all test samples by hand. Noted that we still come up with a decent recognition methodologies using Fast-RCNN despite that considering the final efficiency, the manual way is still better and therefore adopted. (Fortunately, someone put rotation and location information of train data in public online, so we are able to do it via simple programing.)

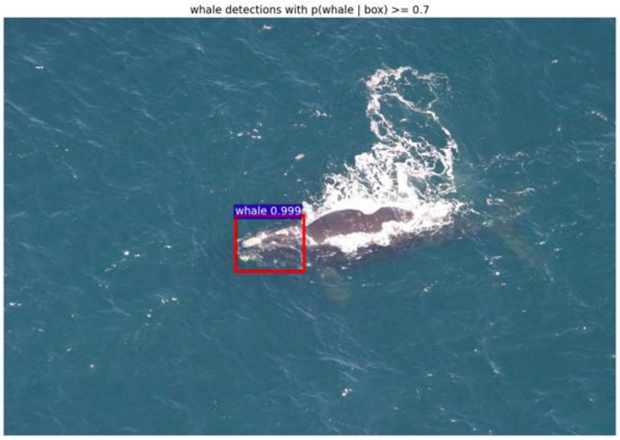


Figure. 3: Demo of Fast R-CNN. Fine-tuned on VGG-16 network with 2,679 human-labelled images.

3. Whale Face Classification

There're a variety of approaches to implement image classification. Traditionally, image classification consists of three parts: 1) SIFT [7] (feature extraction) 2) Bag-of-features [8] (representation model) and 3) Classification. Beyond the traditional process, we have tried more than a few deep learning metrics to improve our classification results, including the original 4) CNN work used in whale face recognition phase, 5) VGG16, MATLAB neural network 6), Neon, 7), and 8) GoogLeNet, some of which didn't show up as a satisfying adoption of methodologies. Hence, we will only illustrate some of the most effective works from these frameworks.

3.1. SIFT

In order to describe the images in an apparent way,

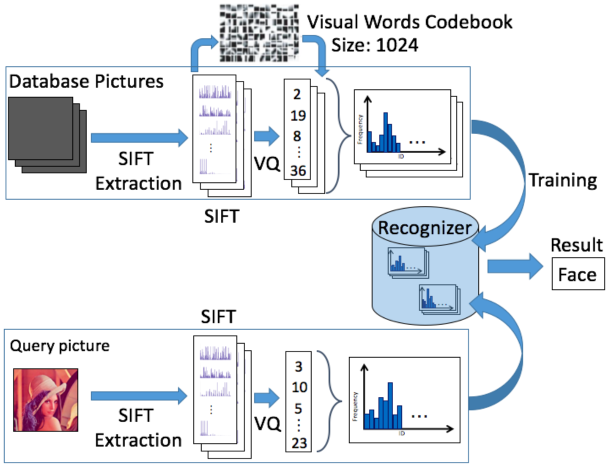


Figure. 4: Common process of image classification [11]

feature extraction is used in most solutions. SIFT (Scale-Invariant Feature Transform) is used for extracting scale-invariant keypoints and computes its descriptor to form distinctive image features. In our experiment, SIFT is mainly computed in two steps: 1) Keypoints Detection: For standard SIFT, images across various scales are convolved with Gaussian filter, and keypoints are detected by Difference of Gaussian (DoG). 2) Feature Description: The direction and magnitude of gradient are computed for every pixel in a neighboring region of the detected keypoint. Finally, the neighboring regions are divided into 16 blocks of 4×4 pixel, each with 8 bin orientation vector, which form an SIFT of 128-dimensional vector.

3.2. Bag-of-features

Bag-of-features is a popular representation model, usually used to represent a single picture. First, feature extraction (e.g. SIFT) is implemented. Next, extracted features are used to train a visual words codebook through K-means clustering, then the nearest visual word is assigned to the extracted features from the visual words codebook by vector quantization. All assigned visual words of a single picture form a high dimensional (dependent on number of visual words) histogram vector.

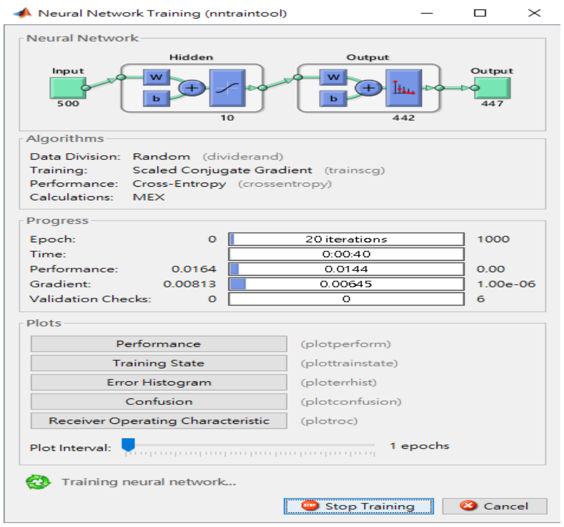
3.3. Classification

Such vectors described above can be trained through classifiers (e.g., Support Vector Machine) and be used to predict the class of queried pictures. Also, queried pictures are represented in the same way as database pictures (i.e. histogram vectors here). In this step, the accuracy of prediction will be evaluated across different sizes of training and table-forming database.

In our implementation, we have tried bag of sift and tuned some parameter like the size of the visual words codebook (up to 3000), and step size of SIFT. Unfortunately, the result is not remarkable regardless of the size we chose, so we consider turning to CNN for visual recognition instead.

3.4. CNN for Visual Recognition

Recently, the development in neural network (aka deep learning) can be used to deal with visual recognition tasks such as object classification, localization and detection. By means of neural network, one can implement these tasks even without having computer vision knowledge to convert a pretty good result. Also, there're some framework open to everybody. By means of *Caffe* [9], one can easily access to these models [10] using the framework. Next, we introduce the VGG16 model with the input requirements and its usage.



Algorithms

Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Cross-Entropy (crossentropy)
Calculations: MEX

Progress

Epoch:	0	20 iterations	1000
Time:		0:00:40	
Performance:	0.0164	0.0144	0.00
Gradient:	0.00813	0.00645	1.00e-06
Validation Checks:	0	0	6

Figure 5: The ongoing neural network training condition using the GUI interface in MATLAB

3.5. VGG16

According to the documentation, VGG16 increases the depth to 16 weight layers, which is substantially deeper than what has been used in the prior-art. To reduce the number of parameters in such a deep network, a very small filter sized at 3*3 is used in all of the convolutional layers (convolution stride is set to 1). This results in the team (VGG) getting only 7.5% top-5 classification error on the validation set.

The first step of constructing our fine-tuned model on the VGG16 pre-trained model is to resize all training images to the size of 224*224*3. We have used the train_val.prototxt given by Andrej Karpathy, but he didn't give the blobs_lr on each layer. We then set blobs_lr to 1 and 2, and set bolbs_lr 11 and 12 in the final layer. During the training phase, we can only set test batch size to 10 because the gpu will be out of memory if set to be greater than 10. Training VGG16 is much more time-consuming than all the other models we've tried, we only ran 20000 iterations and got the top-1 accuracy at 38.1% and top-5 accuracy at 54% on the validation set.

3.6. MATLAB Neural Network

Another option we have tested is the built-in neural network model in MATLAB. It is comprised of several functions with tunable parameters and a graphic user interface to allow a more convenient way to implement. As shown in figure 5 regarding the experiment, we built one or more hidden layers and expect the network to output 6925 lists that evaluates the confidence of resemblance with the 447 right whales. It turns out that the result is not very impressive at all, and the best submission scored a lot lower than using other metrics. To induct the reasons why it

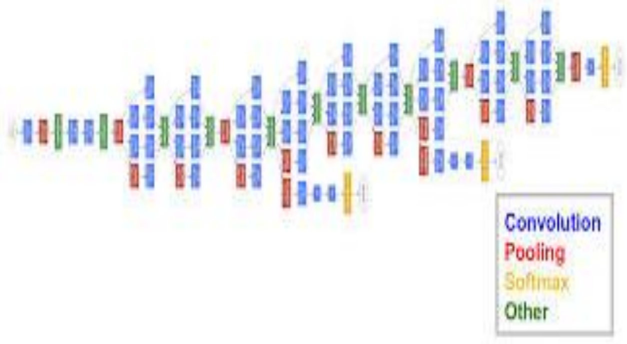


Figure 6: The structural GoogLeNet stated in the text

doesn't work that well, we think this kind of network is not able to meet our need in the classification of as many as 447 different right whales. Perhaps the deep neural network is still a better direction to work on; thus, we move on to instill other conceptual methodology into our trial.

3.7. Neon

About a month before the competition reach its deadline, someone representing an organization called Nervana posted an interesting work on the forum. The work is Neon, a convnet-based framework that process the entire right whale recognition information for you. Since then, People have work on this proposed model to further improve their rank on the leaderboard. It is said that many top ten scores came from the usage of such implementation. We however, decide to assign one of us to work on it. Sadly enough to say that the version of CUDA residing in our server could not be detected by Neon, so we could only run the procedures with CPU supported instead of GPU. It actually matters a lot as we fail to come up with a descent score before the deadline due to the excessive elapsed time. Perhaps next time when encountering such problem, we should spend more time trying to solve the pre-configuration issue rather than compromising to sacrifice the performance which lead to failure.

3.8. GoogLeNet

As an introduction, Google Inc. proposed a extremely high-performance deep neural network "GoogLeNet", which achieved a top-1 accuracy 68.7% (31.3% error) and a top-5 accuracy 88.9% (11.1% error) on the validation set of ILSVRC14. By means of a kind of inner structure named "inception", it improves not only the performance but also the ability to keep the computation budget constant.

Consequently, we decided to fine-tune GoogLeNet on the Right Whale dataset. In our training process, we took care of the parameter tuning (e.g. learning rate, step size, gamma, weight decay) and monitor the primary changes of the learning rate. While the learning rate is set to 0.01 at default, the output loss seemed to explode (from ~1 to 100~) so we have to set it smaller. Usually, setting it to 0.02~0.0001 would produce an acceptable result in the condition of GoogLeNet fine-tuned on our dataset. Also, regarding the result, softmax loss usually starts from 10 and descend to about 0.1.

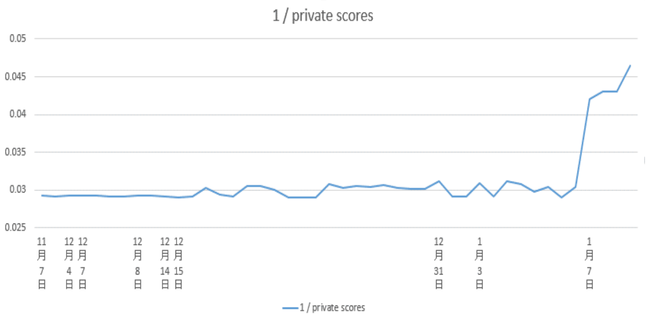
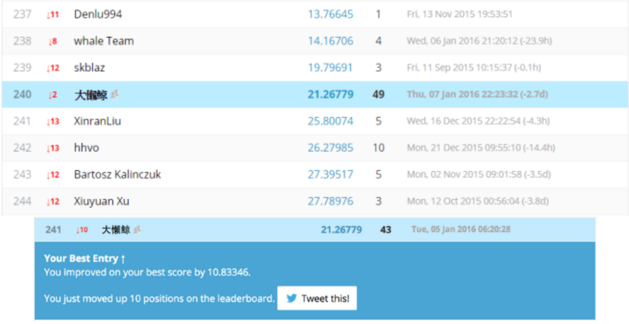


Figure 7: The Submission Graph

Firstly we achieved bad results with top-1 accuracy 12% and top-5 accuracy 25% on the validation set without the rotation of training data. Taking advantage of the tools provided by other competitors, we have come up with a better result with top-1 accuracy 39.2% and top-5 accuracy 52.1% on the validation set with 60,000 iterations. It has been the most satisfying outcome so far.

4. Result

So far in the end of the competition, we have submitted multiple times using a number of methodologies illustrated above. The submission diagram along with the scores evaluated using the multi-class logarithmic class is shown in figure 7. Lower scores are considered better performance, so we attach the inverse function of our score on the y-axis. As indicated by the curve, we had made minor progress as time goes by, and many adoption of our metrics didn't convert to an effective result on the leaderboard. Unfortunately, the approaching deadline didn't provide us with more time to further develop on our work. We ended up getting a rank of 240 out of 368 teams. We have submitted 43 times with a the-lower-the-better score at around 21. Much to our disappointment, we didn't reach our expectation set up by ourselves at the beginning of the competition. However, a relatively greater leap reflecting on the score in the later stage of our implementation is



237	↓11	Denlu994	13.76645	1	Fri, 13 Nov 2015 19:53:51
238	↓8	whale Team	14.16706	4	Wed, 06 Jan 2016 21:20:12 (-23.9h)
239	↓12	skblaz	19.79691	3	Fri, 11 Sep 2015 10:15:37 (-0.1h)
240	↓2	大懶鯨	21.26779	49	Thu, 07 Jan 2016 22:23:32 (-2.7d)
241	↓13	XinranLiu	25.80074	5	Wed, 16 Dec 2015 22:22:54 (-4.3h)
242	↓13	hhvo	26.27985	10	Mon, 21 Dec 2015 09:55:10 (-14.4h)
243	↓12	Bartosz Kalinczuk	27.39517	5	Mon, 02 Nov 2015 09:01:58 (-3.5d)
244	↓12	Xiuyuan Xu	27.78976	3	Mon, 12 Oct 2015 00:56:04 (-3.8d)
241	↓10	大懶鯨	21.26779	43	Tue, 05 Jan 2016 06:20:28

Your Best Entry ↑
You improved on your best score by 10.83346.

You just moved up 10 positions on the leaderboard.

Figure 8: Final Result of the Competition

accomplished by GoogLeNet using a more precisely equivalent scale of training and testing image.

5. Conclusion

In fact, we are the only team choosing Right Whale Recognition to work on in the class. Compared to other competitions, we clearly suffer from a higher threshold of machine learning skills, especially deep learning technique that are not a part of the class materials. Besides, we are also required to acquire enough knowledge about computer vision since we have to deal with image processing and recognition most of the time during the competition. The result is unsatisfying but will push us harder to strive for improvement in the future.

6. Group Assignments

Meng-Jiun Chiou:

- Work on Fast-RCNN, SIFT,VGG16, GoogLeNet
- Implement and submit the highest score of our team (21.26).
- Provide the team with computer vision background knowledge.

Li-An Yang:

- Work on cascade object detector, MATLAB neural network, and Neon.
- Involve intensive effort in the initial survey of the competition and whale face localization.
- First author of the final report.

Kai-Sheng Yang:

- Work on the automatic image rotation and cropping using slopes.
- Utilize online resources and make the image pre-processing phase ready for classification.

References

[1] https://www.kaggle.com/c/noaa-right-whale-recognition

[2] Viola, Paul and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. Volume: 1, pp.511–518.A. Alpher. Frobnication. Journal of Foo, 12(1):234–778, 2002.

[3] https://en.wikipedia.org/wiki/Convolutional_neural_network

[4] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014. tr.pdf.

[5] Girshick, Ross, et al. "Region-based Convolutional Networks for Accurate Object Detection and Segmentation."

[6] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[7] D. G. Lowe, “Distinctive image features from scale-invariant key- points,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[8] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *Proc. 9th Int. Conf. Comput. Vis.*, 2003, vol. 2, pp. 1470–1477.

[9] http://caffe.berkeleyvision.org/

[10] http://caffe.berkeleyvision.org/model_zoo.html

[11] Meng-Jiun, Chiou, Toshihiko Yamasaki and Kiyoharu Aizawa, “A Fast Method of Visual Words Assignment of Bag-of-Features for Object Recognition”, in The 18th Meeting on Image Recognition and Understanding, 2015