Machine Learning Final Report: The Nature Conservancy Fisheries Monitoring

Yinhao Xiao*, Juru Huang†, Szuling Chen†
*Department of Computer Science, The George Washington University
†Data Science, The George Washington University

***Abstract*—The conservancy of nature fishery is important these days. In the Western and Central Pacific, where 60 percent of the worlds tuna is caught, illegal, unreported, and unregulated fishing practices are threatening marine ecosystems, global seafood supplies and local livelihoods. There is a very big problem is that we know very little about what happens on the fisher's boats, and many commercial fishing companies dont have enough money to place an observer on board to monitor the kinds and amounts of the fish by caught, in particlulary by catching tortoise and sharks. Currently, the best solution of this problem is to invest in a monitoring solution that puts GPS cameras on boats with a machine-learning software to track and review the daily catch. Using this way, although these electronic monitoring systems work well and are ready for wider deployment, the amount of raw data produced is cumbersome and expensive to process manually.**

The Nature Conservancy, an enviornment non-profit company, is working with local, regional and global partners to more easily identify and count fish caught on the boat to preserve this fishery for the future. The nature consevancy invite the kaggle company to automatically detect and classify species of tunas, sharks and more that fishing boats catch. The Kaglle name is "The Nature Consevancy Fisheries Monitoring" which collects ideas and models for automatic monitoring, detection and recognition of fisheries activies.

In this report, we demonstrated the details of the process how we finishing the competition. We have tried at leat 3 models, and have found that one of them works particually well. Finally, we ranked 48 out of 2,293 teams(2%) for the first round data and 224 out of 2,293(10%) for the second round. of the worlds tuna is caught, illegal, unreported, and unregulated fishing practices are threatening marine ecosystems, global seafood supplies and local livelihoods. The Nature Conservancy, an enviornment non-profit company, is working with local, regional and global partners to preserve this fishery for the future

I. Introduction

People are fantasy about eating fish these days. Fishery brings a lot more economical benefits than ever before. However, illegal or excessive manhunt can cause sever ecological inbalance to the wild nature. Having said that, the need for monitoring fishery in order to prevent illegal hunting is right at the corner. Kaggle has hosted a competition named "The Nature Conservancy Fisheries Monitoring" [1].

Nearly half of the world depends on seafood for their main source of protein. In the Western and Central Pacific, where 60% of the worlds tuna is caught, illegal, unreported, and unregulated fishing practices are threatening marine ecosystems, global seafood supplies and local livelihoods. The Nature Conservancy is working with local, regional and global partners to preserve this fishery for the future.

Currently, the Conservancy is looking to the future by using cameras to dramatically scale the monitoring of fishing activities to fill critical science and compliance monitoring data gaps. Although these electronic monitoring systems work well and are ready for wider deployment, the amount of raw data produced is cumbersome and expensive to process manually.

The Conservancy is inviting the Kaggle community to develop algorithms to automatically detect and classify species of tunas, sharks and more that fishing boats catch, which will accelerate the video review process. Faster review and more reliable data will enable countries to reallocate human capital to management and enforcement activities which will have a positive impact on conservation and our planet.

Machine learning has the ability to transform what we know about our oceans and how we manage them. You can be part of the solution.

There are 6 kinds of fishes we need to classify. ALB(Albacore tuna), BET(Bigeye tuna), DOL(Dolphinfish, Mahi Mahi), LAG(Opah, Moonfish), Shark(Various: Silky, Shortfin Mako), YFT (Yellowfin tuna). However, we have 8 classifications in total, besides the previous 6 types of fishes, there are 2 more which are No Fish and Other Fish. (No Fish: There is no fish in the whole image, Other Fish: There is fish in the image but the fish does not belong to the previous 6 types of fish.)

For our project, we have tried multiple models:

- 2-step (Fish detection+recognition) method with R-CNN (Region based CNN).
  - For detection: Sliding window + Revised O-NET; For recognition: Revised Zeiler&Fergus Net
  - For detection: Region of Interest(ROI) such as Selective search + Revised O-NET; For recognition: VGG-16/Revised Zeiler&Fergus Net
  - For detection: Bounding box regression with O-NET and Zeiler and Fergus Net; For recognition: VGG-16 and Zeiler&Fergus Net
- Whole image prediction
  - Zeiler&Fergus Net
  - VGG-16

TABLE I: The Revised O-Net Structure. FC represents "Fully Connected Layer"

Layer	Input Size	Output Size	Kernel Size
Conv1	$48 \times 48 \times 3$	$48 \times 48 \times 32$	$3 \times 3 \times 3$
Pool1	$48 \times 48 \times 32$	$16 \times 16 \times 32$	$3 \times 3 \times 32$
Conv2	$16 \times 16 \times 32$	$16 \times 16 \times 64$	$3 \times 3 \times 32$
Pool2	$16 \times 16 \times 64$	$6 \times 6 \times 64$	3×3
Conv3	$6 \times 6 \times 64$	$6 \times 6 \times 64$	3×3
Pool3	$6 \times 6 \times 64$	$3 \times 3 \times 64$	2×2
Conv4	$3 \times 3 \times 64$	$3 \times 3 \times 128$	2×2
FC1	1152	256	
Dropout			
FC2	256	2	

The reasons why we did not use more sophisticated or complicated DNN structures, e.g., GoogLeNET and Resnet are both because time-comsumptions of those method are really high and we do not have a strong hardware-capable(strong GPU clusterings) device to support these methods.

II. Approach

In this section, we are going to demonstrate our methods in details. As shown in Section I, we leverage 5 models in total. Among all these five models, we actually leverage 3 CNN models in total. The revised O-Net, revised Zeiler&Fergus Net and VGG-16. The structure O-Net was first proposed by Zhang et.al., [**?**]. We tested the original model, out loss function does not decrease after certain iterations. Therefore, we need to revise the structure of the network a little bit to embrace our own problem. After several trials, we came up with a revised O-Net model that fits our problem the most. Its structure is shown as Table I. Zeiler&Fergus Net were first proposed by Zeiler et.al. [2] and was widely used in different learning tasks. Later Schroff et.al., [3] revised the network structure by adding one 1×1 convolutions inspired by the work done by Lin et.al., [4]. Based on the revision of [3], we further revised the network structure to better serve our problem. Our revised Zeiler&Fergus Net is shown as Table II. VGG-16 was first proposed by Simonyan et.al., [5]. We did not change the structure of VGG-16 since we use VGG-16 as a fine tune model, that is, we re-train part of the weights instead of training the whole model from scratch. As a matter of fact, we re-train the weights and bias of all the fully connecte layers as well as the last convolution layer.

A. Fish Detection + Fish Recognition with R-CNN

The most straightforward way to complete the competition will be performing fish detection fist, then classifying the selected fish region.

Sliding Window Method Sliding window methods are widely used in object detection. The basic mechanism is let a fixed-size window to brute-force scan the whole image and detect which block of the target object is located. One problem is determin the size of the window which should not be too small nor too large. We observed that most of the images have the size of 1280×960, and most of the time, the fish occupies 10% of the image. Therefore, we set the window size to be

TABLE II: The Revised Zeiler&Fergus Net

Layer	Input Size	Output Size	Kernel
Conv1	$220 \times 220 \times 3$	$110 \times 110 \times 64$	$7 \times 7 \times 3$
Pool1	$110 \times 110 \times 64$	$55 \times 55 \times 64$	$3 \times 3 \times 64$
Batch Norm			
Conv2a	$55 \times 55 \times 64$	$55 \times 55 \times 64$	$1 \times 1 \times 64$
Conv2	$55 \times 55 \times 64$	$55 \times 55 \times 128$	$3 \times 3 \times 64$
Batch Norm			
Pool2	$55 \times 55 \times 128$	$28 \times 28 \times 128$	$3 \times 3 \times 128$
Conv3a	$28 \times 28 \times 128$	$28 \times 28 \times 128$	$1 \times 1 \times 28$
Conv3	$28 \times 28 \times 128$	$28 \times 28 \times 256$	$3 \times 3 \times 128$
Batch Norm			
Conv4a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256$
Conv4	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256$
Batch Norm			
Conv5a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256$
Conv5	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256$
Batch Norm			
Conv6a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256$
Conv6	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256$
Pool6	$14 \times 14 \times 256$	$7 \times 7 \times 256$	$3 \times 3 \times 256$
Batch Norm			
FC1	12544	128	
Dropour			
Fc2	128	7	

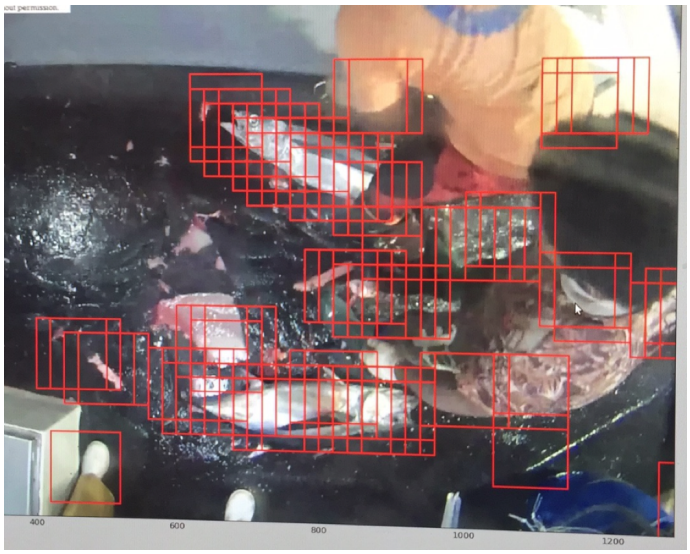
100×100. For each input window block, we resize the block to 48×48 in order to feed into our revised O-Net. Then if there is no fish in the block, O-Net outputs $[0, 1]$, otherwise, outputs $[1, 0]$. As for training, we cropped all the fishes as well as randomly cropped some regions that do not have fish and feed them to train. We uses the softmax cross entropy as our loss function shown as Equation 1. As for recognition, we resize the block that contains the fish to 220×220 and feed into Zeiler&Fergus Net, and outputs 8 classifications training with the softmax loss.

$$\frac{1}{N}\sum_{i}(y_i \cdot (-log(\frac{x_i}{\sum_j x_j})) + (1 - y_i) \cdot (-log(1 - \frac{x_i}{\sum_j x_j}))) \quad (1)$$

Selective Search Method Selective search, as one of the most popular region proposal mechanism, was first proposed by Uijlings et.al., [6]. Compared to sliding window, this method proposes more reasonable region candidates by segmenting the image and grouping the parts based on the sum of color, texture and fill similarities. Rest of the steps are very similar to what we did for sliding window method. Even though the method may have more reasonable proposals, it can sometimes miss the fish part or wrongfully group fish with other objects. Moreoever, it is highly computational expensive with the running time of $O(n^3)$.

Bounding Box Regression Bounding box regression is one of the widely used technique to retrieve the target region. By doing bounding box regression, we feed an image to our desired networks, in our case, we used both O-Net and Zeiler&Fergus Net. Both networks outputs 4 valuse, top-x

(a) Sliding Window O-Net Fish Detectio Example 1



(b) Sliding Window O-Net Fish Detectio Example 1

Fig. 1: Sliding Window Fish Detection

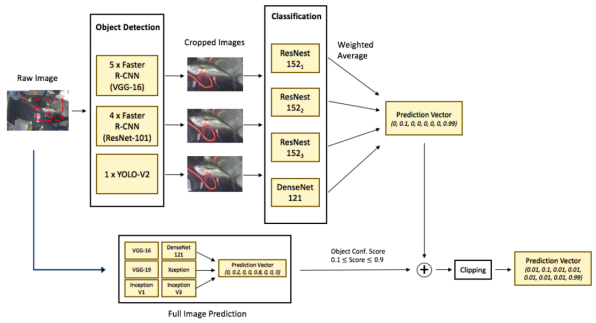


Fig. 2: Felix Yu's Model.

coordinate, top-y coordinate, width and height of the fish region. For training, we define the loss function to be L^2 loss shown as equation 2

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|^2 \tag{2}$$

B. Whole Image Prediction

In the whole image prediction, we used the Zeiler&Fergus Net and VGG-16 to directly classify the fish types based on the whole image. This may sound bizarre, however, we indeed noticed that there is a correlation between boats and the fish. In this method, we again employ the softmax cross entropy in equation 1 as our loss function.

III. Performance

In this section, we are going to show our performance of the models we build as well as our ranking in kaggle.

A. Sliding Window For Fish Detection

For O-Net based sliding window fish detection for posterior recognition task.on, performance is ok, but our performance shows that false positivity is common. False positivity could be a huge problem. We include two performances of our O-Net sliding window fish detection shown in Figure 1. As we can see, Figure 1a has false positive in some bright areas of the pictures. Figure 1b has even more false positive predictions. Therefore, we can see that sliding window method is somehow not satisfying.

B. Selective Search For Fish Detection

For O-Net based selective search fish detection for posterior recognition task.on, performance is better off than sliding window. However, due to heavy overhead, this method does not fit for second stage competition which has more than 12,000 images needed to be classified, we abandon this method as well. The performance is shown as 3

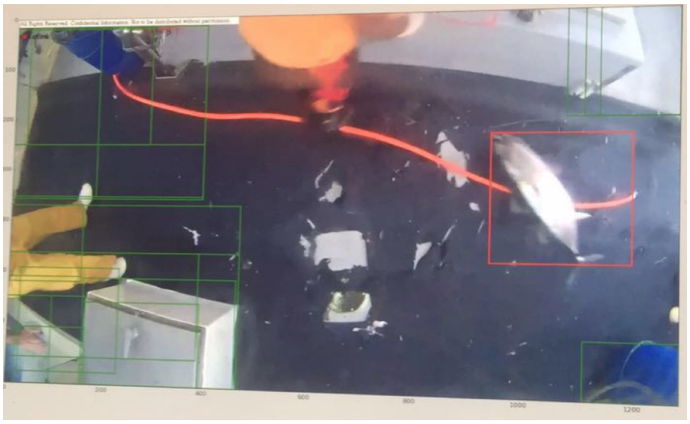
C. Whole Image Prediction

Finally we have found out that using Zeiler&Fergus Net for whole image prediction has the best performance out of all other models (including ensembled ones).

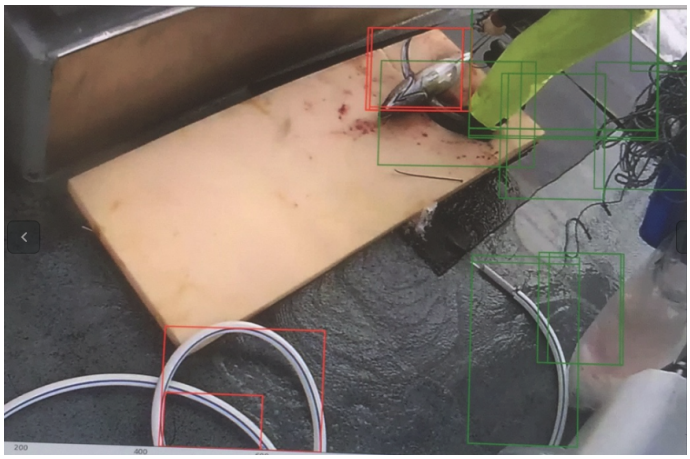
Finally, for the first round competition which has 1,000 images for classification, our rank is 48 out of 2,293 teams (top 2%). Unfortuantely, the host of the competition actually change the dataset obtained from another boat for second round competition which has more than 12,000 images. Our rank drops to 224 out of 2,293 teams (top 10%) and we won a bronze medal.

IV. Other Solutions

So far, there is only one person, Felix Yu who ranks 8th out of all teams, shares his solution [7]. For fish detection, he



(a) Green and red rectangles are proposed regions, red is detected region.



(b) Green and red rectangles are proposed regions, red is detected region.

Fig. 3: Selective Search Fish Detection

uses $5\times$Faster VGG-16 and $4\times$ Faster ResNet-101 and YOLO-v2. Then ensemble them together. Then for classification, he uses three ResNet and 1 DenseNet and ensemble. Finally he uses full image prediction for ensemble all the predictions. The model is shown as Figure 2.

REFERENCES

[1] "The nature conservancy fisheries monitoring overview," https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring, observed in May 2017.

[2] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[4] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[7] "Solution by felix yu," https://flyyufelix.github.io/2017/04/16/kaggle-nature-conservancy.html, observed in May 2017.