# DEEP NEURAL NETWORKS FOR SIMULTANEOUSLY LOCALIZING FISHES AND RECOGNIZING THEIR SPECIES IN UNDERWATER VIDEOS

*Roshan Pati*

# DEEP NEURAL NETWORKS FOR SIMULTANEOUSLY LOCALIZING FISHES AND RECOGNIZING THEIR SPECIES IN UNDERWATER VIDEOS

*Thesis submitted to*
*Indian Institute of Technology Kharagpur*
*for the award of the degree*

*of*

**Bachelor of Technology**
***in***
**Electrical Engineering**

*by*

**Roshan Pati**
**(13EE10062)**

*Under the guidance of*

**Dr. Debdoot Sheet**


**DEPARTMENT OF ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR**
**MAY 2017**

# DECLARATION

I certify that

**a.** The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.

**b.** The work has not been submitted to any other Institute for any degree or diploma.

**c.** I have followed the guidelines provided by the Institute in writing the thesis.

**d.** I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

**e.** Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

**f.** Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Roshan Pati

# Certificate

This is to certify that the B.Tech project entitled "***Deep Neural Networks for Simultaneously Localizing Fishes and Recognizing their Species in Underwater Videos***", which has been submitted by **Roshan Pati(13EE10062))**, is a bonafide work carried out by him under our supervision and guidance. It is worthy of consideration for the partial requirement for the award of the degree Bachelor of Technology in Electrical Engineering by the Indian Institute of Technology Kharagpur.

**Dr. Debdoot Sheet**
**Department of Electrical Engineering**
**IIT Kharagpur**

# Acknowledgment

I am immensely grateful to **Prof. Debdoot Sheet** for giving me this opportunity to work on this exciting project, and for his constant help and support, without which it would not have been possible to progress productively. His constant guidance and productive discussions with me have gone a long way in giving the right shape to my work. I am very thankful to him for his resourceful inputs and help during the process. I will always be obliged to him for showing faith in me to do this work.

I would also like to thank the members of **Signal and Image Processing Laboratory**[1], Department of Electrical Engineering, IIT Kharagpur for providing me with the resources and aid me in learning the deep learning frameworks and implementing them in my work. I also extend my heartfelt thanks to my classmates, who have encouraged and supported me through extensive discussions, to carry out my work.

Lastly, I would like to thank my parents and family who have always been there to guide and encourage me through every aspect of life. There is little that could have been accomplished without them.

[1] http://www.ee.iitkgp.ernet.in/kliv/

# Contents

**List of Figures** viii

**List of Tables** ix

**List of Abbreviations** x

**List of Symbols** xi

**Abstract** xii

**1 Introduction** 1

**2 Prior Work,Objectives and Datase** 5

2.1 Related Work . . . . . 5

2.2 Scope and Objectives . . . . . 6

2.3 Dataset . . . . . 6

**3 Baseline Method(Manual patch extraction and classification** 9

3.1 Methodology . . . . . 9

3.2 Training . . . . . 10

3.3 Evaluation . . . . . 11

**4 Joint Network Model** 12

4.1 Methodology . . . . . 12

4.2 Region Proposal Network . . . . . 12
4.3 Anchor Boxes . . . . . 13
4.4 Loss Function . . . . . 14
4.5 Training . . . . . 16
4.6 Feature Sharing . . . . . 16
4.7 Evaluation . . . . . 16

**5 Conclusion and Appendix** **18**
5.1 Summary . . . . . 18

**Bibliography** **24**

**Curriculum vitae** **26**

# List of Figures


1.1 Classification Pipeline . . . . . 4

1.2 Image Pyramid . . . . . 4

2.1 The sample images obtained from Fish4Knowledge dataset. . . . . 8

3.1 The deep CNN architecture for classification. . . . . 10

3.2 Confusion Matrix . . . . . 11

3.3 Evolution of loss function and accuracy for the classification network . 11

5.1 correctly predicting the class(8) of the fish with high class score . . . . 19

5.2 correctly predicting the class(9) of the fish with high class score . . . . 20

5.3 correctly predicting one out of two fishes present in the image . . . . . 21

5.4 correctly predicting the class but high region proposal error . . . . . 22

5.5 Incorrect Class prediction . . . . . 23

# List of Tables

2.1 Fish images present in training set . . . . . 7

# List of Abbreviations

| | |
|---|---|
| AE | Autoencoder |
| CNN | Convolutional Neural Network |
| RPN | Region Proposal Network |
| RCNN | Region-based Convolutional Neural Network |
| CRF | Conditional Random Field |
| EDM | Euclidean Distance Method |
| ANN | Artificial Neural Network |
| SIFT | Scale Invariant Feature Transform |
| PCA | Principal Component Analysis |
| BoVW | Bag of Vector Words |

# Abstract

THE goal of the fish identification task is to identify fish occurrences in underwater video sequences. While manual analysis of underwater videos performed by human operators is the current practice, it is highly impractical and is a hectic job to continuously analyze the large amount of underwater videos. These have immense application in marine biology for understanding the marine ecosystem . Nevertheless, the development of automatic video analysis tools is challenging because of the complexities of underwater video recordings in terms of the variability of illumination, background models, optical scattering in media, etc. factors that often degrade the video quality. Because of these challenges, traditional machine learning algorithms often fail to learn the complex features present in such images. Hence We have used the state of the art technology of using Convolutional Neural Networks(CNNs) for this purpose. Two approaches have been made in this regard. With the first approach(the baseline method) the task is divided into two parts. In the first part a CNN is trained to give the Region of Interests(RoIs) of the fishes present in the video frames. Then the fish image patches are extracted from these RoIs. These image patches are then passed through another pre-trained CNN classifier that produces the classes of the fishes. In the second method a joint network comprising of a region proposal network and a detection network with shared layers is trained to localize and classify the fishes present in the video frames.We have used the Fish4Knowledge dataset that consists of 93 under-water videos each spanning for an average of 50-60 seconds. The accuracy the joint network over fishes of 14 classes achieved is 86.2%.

***Keywords:*** convolutional neural networks, Region Proposal Networks, transfer learning, visual features, Underwater Fish Species Recognition

# 1 Introduction

THE typical usage scenario of automated underwater video analysis tools is to support marine biologists in studying thoroughly the marine ecosystem and fish biodiversity. Also, scuba divers, marine stakeholders and other marine practitioners may benefit greatly from this kind of tools. It would also efficiently substitute the traditional techniques used for studying fish populations, such as casting nets or human manned photography which affects the fish behaviour to a large extent

Major work that have been done in this field uses traditional image processing methods like feature extraction, patch encoding, pooling followed by the traditional machine learning algorithms like support vector machine classifiers (Farrell et al., 2011; Khan et al., 2011). Our approach is to utilize the state of the art Convolution Neural Networks (CNNs) to extract features and to classify fish species. Convolutional networks were inspired by biological processes (Matsugu et al., 2003) and are very similar to ordinary neural networks. They are made up of neurons that have learnable weights and biases. ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations.

CNNs are currently one of the most successful feature learning methods in performing various computer vision tasks. (LeCun et al., 1998) proposed to use a CNN architecture, referred to as LeNet to address the image classification problem on the

digit classification on MNIST database . They also showed that the features learnt by the CNN dramatically improve the classification results compared to the state-of-the-art handcrafted features like FVs.

CNNs are difficult to train as they typically contain a high number of unknowns. As an example, the LeNet architecture contains over 30 million parameters and about $25,000$ neurons. This requires to have a high computational power and a huge amount of annotated data to train the networks. (Girshick et al., 2014) showed that a new network can be learnt despite the scarcity of data by performing transfer learning. They proposed to use a pre-trained model as initialization and fine-tune it to obtain a new network. Is is shown that fine-tuned network yielded a significant performance boost for object recognition task, despite being fine-tuned on a network trained for image classification. The architecture of a CNN is designed to take advantage of the 2D structure of an input image and hence predict the class the image belongs to. The major advantages of CNNs over the traditional machine learning algorithms is that CNNs with their local connections and tied weights followed by some form of pooling and fully connected layers are more efficient and achieve better results for complex applications with a huge amount of data than the traditional algorithms.

Recent advances in object localization and detection are based on the success of region proposal methods (e.g.(Uijlings et al., 2013)) and region-based convolutional neural networks (RCNNs). The cost of originally computationally expensive region-based CNNs has been significantly reduced thanks to sharing convolutions across proposals (He et al., 2014; R.Girshick, 2013). The latest model, Fast R-CNN (R.Girshick, 2013), achieves almost real-time rates using deep networks (Simonyan and Zisserman, 2015), when not considering the time taken by the proposal network. However the time spent on proposal networks increases the overall computational time to a large extent. Hence, now proposals are the test-time computational bottleneck in state-of-the-art detection systems.

Region proposal methods typically rely on inexpensive features and economical inference schemes. Selective Search (Uijlings et al., 2013), one of the most popular

methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks (R.Girshick, 2013), Selective Search is an order of magnitude slower, at 2 seconds per image in a CPU implementation. EdgeBoxes (Zitnick and Dollar, 2014) currently provides the best tradeoff between proposal quality and speed, at 0.2 seconds per image. Nevertheless, the region proposal step still consumes as much running time as the detection network. This is because fast region-based CNNs take advantage of GPUs, while the region proposal methods used in research are implemented on the CPU, making such runtime comparisons inequitable. An obvious way to accelerate proposal computation is to reimplement it for the GPU. This may be an effective engineering solution, but re-implementation ignores the down-stream detection network and therefore misses important opportunities for sharing computation.

We implement the region proposal network using a deep CNN which leads to an elegant and effective solution where proposal computation is nearly cost-free given the detection network's computation.We use such Region Proposal Network(RPN) that share convolutional layers with state-of-the-art object detection networks (R.Girshick, 2013; He et al., 2014). By sharing convolutions attest-time, the cost of proposal computation is small (e.g., 10ms per image). The idea behind is that the convolutional feature maps used by region-based detectors, like Fast RCNN, can also be used for generating region proposals. On top of these convolutional features, we make a RPN by adding a few additional convolutional layers that simultaneously regress region bounds and objectness scores at each location on a regular grid. The RPN is basically a fully convolutional network that can be trained for generating region proposals.

We use RPNs to predict region proposals with a wide range of scales and aspect ratios. In contrast to prevalent methods (Sermanet et al., 2014; Felzenszwalb et al., 2010; He et al., 2014; R.Girshick, 2013), that use pyramids of images Figure 1.2(a) or pyramids of filters Figure 1.2(b), we use anchor boxes that act as references at multiple scales and aspect ratios. The scheme ia a pyramid of regression references Figure 1.2(c) , which avoids enumerating images or filters of multiple scales or aspect ratios. This model performs well when trained and tested using single-scale images and

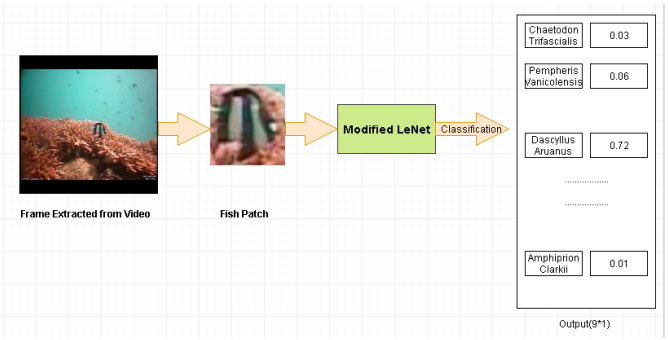



Figure 1.1: Classification Pipeline

thus increases running speed. To unify RPNs with Fast R-CNN (R.Girshick, 2013) object detection networks, we use an alternating training scheme that alternates between fine-tuning for the region proposal task and then fine-tuning for object detection, while keeping the proposals fixed.

For the baseline classification problem, the fish image patches are extracted from the image frames manually. Further these patches are passed through the pre-trained modified version of LeNet model to obtain the class of the fish species.


(a) "fig1a" (b) "fig 1b" (c) "fig 1c"

Figure 1.2: Image Pyramid

# 2 Prior Work,Objectives and Datase

## 2.1 Related Work

Several techniques have been specifically devised to deal with fine-grained recognition (Branson et al., 2010; Farrell et al., 2011; Khan et al., 2011; Wah et al., 2011; Yao et al., 2012) mostly focusing on the discovery of visual features which are more discriminative at the subordinate level. In (Khan et al., 2011), a combination of visual cues extracted from training images is used to build discriminative compound words. In (Kuan et al., 2012), image patches are considered as discriminative attributes and a Conditional Random Field (CRF) framework is used to learn the attributes on a training set with humans in the loop.

Dense-sampling techniques have also been explored which decompose an image into patches and then extract low-level features from these regions to identify fine image statistics. In (Yao and Li, 2010) the authors introduce Grouplet, a set of generative local dense features, which work reliably for human activity recognition. As follow-up, the same authors, in (A.Khosla et al., 2014), improved the performance of their former method by fusing global and local information and using a random forest based approach to extract dense spatial information. The limitation of these methods is their efficiency, as dense sampling feature spaces are often huge and increase many-fold when employing multiple image patches of arbitrary size. In addition, they are not able to operate with low-quality images, where the finer details necessary to pick the subtle differences between fine-grained object classes are missing.

Several other works in this field include that of (Saitoh et al., 2016) who adopted the technique of extracting the feature points manually and then used a combination of geometric features and BoVW model to predict the fish classes. In 2013 (Pornpanomchai et al., 2013) adopted shape and texture based fish recognition in which they proposed two methods of Euclidean Distance Method (EDM) and Artificial Neural Network (ANN). (Matai et al., 2012) employed the methods of Principal Component Analysis (PCA) and Scale Invariant Feature Transform (SIFT) for this recognition purpose.

## 2.2 Scope and Objectives

The aim of this thesis is to develop a deep neural network architecture for classifying the fishes present in under-water videos. The following objectives have been set to achieve this objectives:

1. Classification of the fish images present in under-water videos by first extracting the fish image patches manually and then passing those through a pre-trained CNN to predict the class.

2. Classifying the fishes directly form the video frames by jointly optimizing both the Region Proposal Network and the Classification Network using faster Region Based Convolutional Neural Network(RCNN)technique.

## 2.3 Dataset

The dataset used consists of several underwater video sequences collected by NCHC in Taiwan and used in the Fish4Knowledge project . The training set is built up of 20 manually annotated videos, a list of 15 fish species and a set of about 30,000 sample imagesFigure 2.1 to support learning of fish appearance models. The test set contains, instead, 73 underwater videos fully labelled.vEach underwater video spans an average of 50-60 seconds. Out of the 15 species top 9 species were selected for experimental purpose as these species have more than 90% share of the total image set.

For the RPN, frames were extracted from the underwater videos at the rate of 25 frames per second. 2983 frames were considered for training purpose while 1941 frames were used for testing. The region masks for the training images were generated using the ground truth xml file provided along with the dataset.

The classification network was trained by using the sample images. A total of 17240 images of size $32 \times 32$ pixels were used for training.The fishes in the image belonged to one of the nine classes. Labels of 1 to 9 were given to these images according to the class of fish present in it Table 2.1. For the testing purpose a total of 11319 fish image patches extracted from the video frames were used. The evaluation was focussed on making the correct prediction of the fish present in the images.

Table 2.1: Fish images present in training set

| Fish Species Name | Label | Number of Images |
|---|---|---|
| Chaetodon Trifascialis | 1 | 2705 |
| Pempheris Vanicolensis | 2 | 1995 |
| Hemigymnus Melapterus | 3 | 365 |
| Dascyllus Reticulatus | 4 | 3365 |
| Dascyllus Aruanus | 5 | 1727 |
| Chromis Crysura | 6 | 645 |
| Amphiprion Clarkii | 7 | 976 |
| Plectrogly Phidododon Dickii | 8 | 1499 |
| Chaetodon Lunulatus | 9 | 2963 |

Figure 2.1: The sample images obtained from Fish4Knowledge dataset.

# 3 Baseline Method(Manual patch extraction and classification

## 3.1 Methodology

The baseline is a slightly modified version of LeNet model available in caffe zoo repository. The network consists of a pair of convolutional layers followed by pooling layersFigure 3.1.The output of the convolutional layer for an input **z** is given as

$$f(i,j) = \sum_{k,l} w(k,l)\, z(i-k, j-l)$$

where, $\mathbf{w}(.)$ is the convolution kernel learnt by the CNN. This layer extracts meaningful features from the input that aids in learning the hypothesis $H(.)$ . Each convolution layer is associated with a specific number of kernels (N), size of the kernel ($k \times k$), and stride of the kernel(S)Figure 3.1. The number of features extracted for each input corresponds to the number of filters.

Pooling layers are introduced to reduce the spatial dimension of the representations learnt by the CNN. This effectively reduces the number of parameters to be tuned in the network thereby controlling overfitting. In the architecture used, max pooling is employed. Like convolutional kernels, pooling kernels are also associated with a size and stride which dictates the reduction of dimension. As the name suggests, the neurons in FCN are connected to all the activations of the previous layer. The output softmax

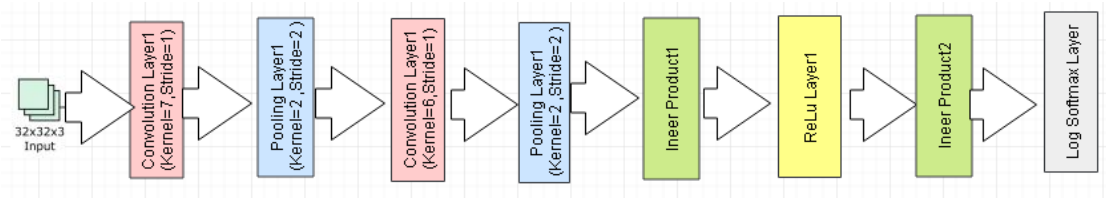



Figure 3.1: The deep CNN architecture for classification.

layer is given as

$$p(y = x|z) = \frac{e^{z^T w_k}}{\sum_{k=1}^{K} e^{z^T w_j}} \tag{3.1}$$

where x is the class label, **w** is the weights learnt by the network and **z** is the input.

The output of the second pooling layer is passed through two fully connected layers. The last fully connected connected layer of 10 neurons from the original LeNet architecture is removed and added a new layer of 9 neurons. The network is trained using stochastic gradient descent method. The loss function used is a sigmoid cross entropy log function given as

$$E = \frac{-1}{n} \sum_{n=1}^{N} [p_n \log \hat{p}_n + (1 - p_n) \log(1 - \hat{p}_n)] \tag{3.2}$$

## 3.2 Training

For the classification network the weights and biases are randomly initialized with the network being fine tuned for 10000 iterations with 64 images in a batch. The learning rate is initialized to $10^{-2}$ and a step decaying policy for the learning rate is used. A momentum of 0.9 and a gamma value of 0.1 are chosen for the training purpose. The network was trained on the same GPU as that of the RPN. The training process took 0.7 seconds for 100 iterations,i.e. completing in roughly 12 minutes.

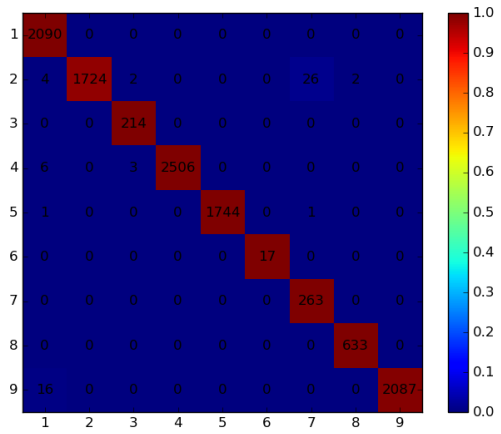



Figure 3.2: Confusion Matrix

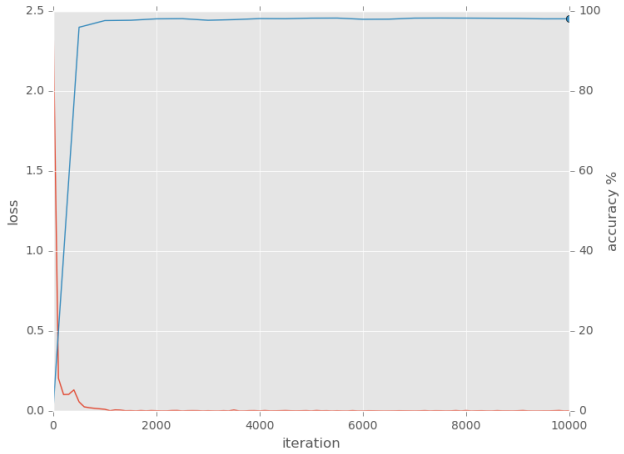



Figure 3.3: Evolution of loss function and accuracy for the classification network

## 3.3 Evaluation

For the classification network an accuracy of 99.5% is obtained. The evolution of loss and the accuracy is given in Figure 3.3. To check over-fitting 4 fold cross-validation is used and the average cross-validation accuracy of 98.4% is obtained. The confusion matrix involving the nine classes is given in Figure 3.2

# 4 Joint Network Model

## 4.1 Methodology

In this approach a joint network is used that is made up of two modules. The first one is a fully convolutional network called the Region Proposal Network (RPN) that proposes regions and the other is the image recognition network that uses the proposed regions. The entire system is a single unified network for object detection. The novelty of this method is the sharing of a few layers of convolutional networks between these two modules and the joint training method.

## 4.2 Region Proposal Network

A region proposal network takes an image as the input and outputs a collection of rectangular object proposals each with its own objectness score. A fully convolutional network is used to model this. The aim is to share computation with an object detection network.Both the nets are assumed to share a common set of convolutional layers.The VGG net is used as the detection network and 13 convolutional layers are shared by both the networks.

To generate object proposals we slide a small network over the convolutional feature map output of the last shared convolutional layer of the detection network. This network takes a $3 \times 3$ spatial window of the convolutional feature map and outputs a lower dimensional feature (256-d). This feature is fed into two parallel fully connected

layers-a regression(reg) layer and a classification(cls) layer. To generate multiple region proposal at a single location we used the concepts of anchor boxes of different sizes and aspect ratios and are centred at the sliding window. The number of anchors or in other words the maximum possible region proposals at a window location is denoted as k. Hence the classification layer has 2k outputs containing the estimated probability of a proposal being an object or not an object and the regression layer has 4k outputs encoding the coordinated of k boxes.

## 4.3 Anchor Boxes

At each sliding-window location, we simultaneously predict multiple region proposals, where the number of maximum possible proposals for each location is denoted as k. So the reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal4. The k proposals are parameterized relative to k reference boxes, which we call anchors. An anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio (Figure 3, left). By default we use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position. For a convolutional feature map of a size $W \times H$ (typically 2,400), there are WHk anchors in total.

An important property of the approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. If one translates an object in an image, the proposal should translate and the same function is able to predict the proposal in either location. This translation-invariant property is guaranteed by the method. Comparing with other methods, the MultiBox method (Szegedy, Reed, Erhan and Anguelov, n.d.) uses k-means to generate 800 anchors, which are not translation invariant. So MultiBox does not guarantee that the same proposal is generated if an object is translated. The translation-invariant property also reduces the model size. MultiBox has a $(4+1) \times 800$ -dimensional fully-connected output layer, whereas our method has a $(4+2) \times 9$-dimensional convolutional output

layer in the case of k = 9 anchors. As a result, the output layer has $2.8 \times 10^4$ parameters (for VGG-16), two orders of magnitude fewer than MultiBox output layer that has $6.1 \times 10^6$ parameters for GoogleNet (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan and Rabinovich, n.d.) in MultiBox (Szegedy, Reed, Erhan and Anguelov, n.d.).

We use a novel scheme for addressing multiple scales (and aspect ratios). There have been two popular ways for multi-scale predictions. The first way is based on image/feature pyramids, e.g., in DPM (Felzenszwalb et al., 2010) and CNN based methods (Sermanet et al., 2014; He et al., 2014; R.Girshick, 2013). The images are resized at multiple scales, and feature maps (HOG (Felzenszwalb et al., 2010) or deep convolutional features (Sermanet et al., 2014; He et al., 2014; R.Girshick, 2013)) are computed for each scale . This way is often useful but is time-consuming. The second way is to use sliding windows of multiple scales (and/or aspect ratios) on the feature maps. For example, in DPM (Felzenszwalb et al., 2010), models of different aspect ratios are trained separately using different filter sizes (such as $5 \times 7$ and $7 \times 5$). If this way is used to address multiple scales, it can be thought of as a pyramid of filters (Figure 1(b)). The second way is usually adopted jointly with the first way . As a comparison, the anchor-based method is built on a pyramid of anchors, which is more cost-efficient. Our method classifies and regresses bounding boxes with reference to anchor boxes of multiple scales and aspect ratios. It only relies on images and feature maps of a single scale, and uses filters (sliding windows on the feature map) of a single size. Because of this multi-scale design based on anchors, we can simply use the convolutional features computed on a single-scale image, as is also done by the Fast R-CNN detecter . The design of multiscale anchors is a key component for sharing features without extra cost for addressing scales.

## 4.4 Loss Function

For training RPNs, we assign a binary class label (of being an object or not) to each anchor. We assign a positive label to two kinds of anchors: (i) the anchor/anchors with

the highest Intersection-over- Union (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. Note that a single ground-truth be thought of as bounding-box assign positive labels to multiple anchors. Usually the second condition is sufficient to determine the positive samples; but still the first condition is adopted for the reason that in some rare cases the second condition may find no positive sample. A negative label is assigned to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective.

Our loss function for an image is defined as:

$$L(p_i,t_i) = \frac{1}{N_{cls}}\sum_i [L_{cls}(p_i,p_i^*)] + \lambda \frac{1}{N_{reg}}\sum_i [p_i^* L_{reg}(t_i,t_i^*)] \tag{4.1}$$

Here, $i$ is the index of an anchor in a mini-batch and $p_i$ is the predicted probability of anchor i being an object. The ground-truth label $p_i^*$ is 1 if the anchor i is positive, and is 0 if the anchor is negative. $t_i$ is a vector representing the 4 parameterized coordinates of the predicted bounding box, and $t_i^*$ is that of the i ground-truth box associated with a positive anchor. The classification loss $L_{cls}$ is log loss over two classes (object vs. not object). For the regression loss, we use $L_{reg}(t_i;t_i^*) = R(t_i - t_i^*)$ where R is the robust loss function (smooth $L_1$). The term $p_i^* L_{reg}$ means the regression loss is activated only for positive anchors $(p_i^* = 1)$ and is disabled otherwise $(p_i^* = 0)$.The outputs of the cls and reg layers consist of $p_i$ and $t_i$ respectively.

The two terms are normalized by $N_cls$ and $N_reg$ and weighted by a balancing parameter . In our current implementation , the cls term is normalized by the mini-batch size (i.e., $N_cls = 256$) and the reg term is normalized by the number of anchor locations (i.e., $N_reg\ 2,400$). By default $\lambda$ is set as 10, and thus both cls and reg terms are roughly equally weighted.

## 4.5 Training

The RPN can be trained end-to-end by back- propagation and stochastic gradient descent (SGD). The 'image-centric 'sampling strategy from is followed to train this network. Each mini-batch arises from a single image that contains many positive and negative example anchors. It is possible to optimize for the loss functions of all anchors, but this will bias towards negative samples as they are dominate. Instead, 256 anchors are randomly sampled in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:1. If there are fewer than 128 positive samples in an image, the mini-batch is padded with negative ones.

all new layers are randomly initialized by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. All other layers (i.e., the shared convolutional layers) are initialized by pretraining a model for ImageNet classification (Russakovsky et al., n.d.), as is standard practice (Girshick et al., 2014). A learning rate of 0.001, a momentum of 0.9 and a weight decay of 0.0005 are used. The implementation uses Caffe (Jia et al., n.d.).

## 4.6 Feature Sharing

Both RPN and the detection network, trained independently, will modify their convolutional layers in different ways. We therefore need to develop a technique that allows for sharing convolutional layers between the two networks, rather than learning two separate networks. For this RPN is first trained, and the proposals are used to train the detection network. The network tuned by the detection network is then used to initialize RPN, and this process is iterated.

## 4.7 Evaluation

The weights of the VGG net model are retrained by fine tuning it as per the requirements. Then both the region proposal network and the classification network were

trained alternatively for 100,000 iterations. The dataset was split 60:40 (out of each 100 images 60 were used for training and 40 for testing). On a GPU of 2Gigabytes of memory each 100 iterations took roughly a minute, hence the entire training process taking around 17 hours for completion. After testing an accuracy of 86% across 14 classes was achieved.Some of the sample output along with their class scores are given in fig Figure 5.1 .

# 5 Conclusion and Appendix

## 5.1 Summary

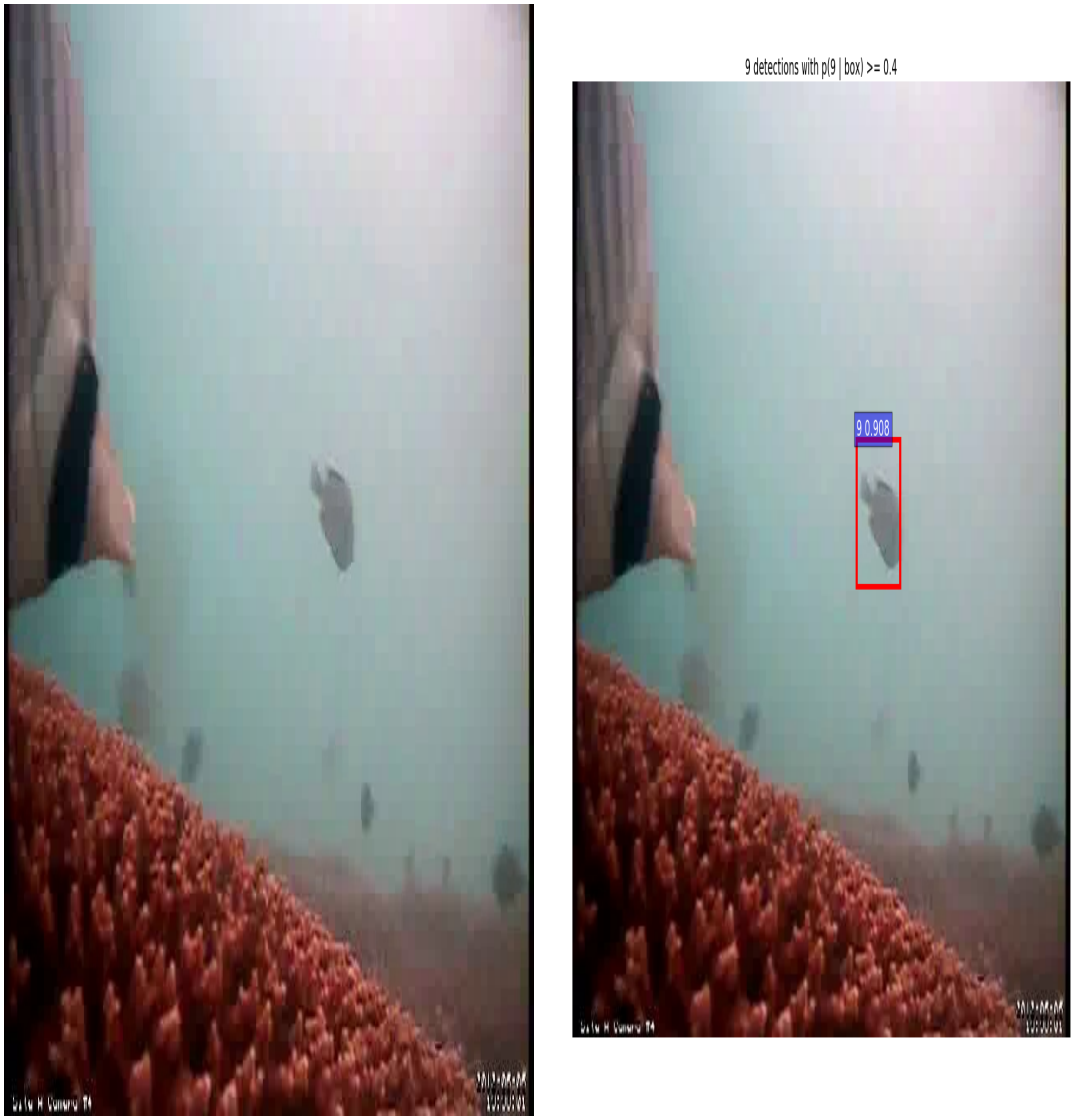
An overall accuracyof over 98% was obtained in the baseline classification method. However it involves manual extraction of cropped fish patches from the image frames. In the main method of joint optimization of a Region Proposal Network(RPN) and a classification network an accuracy of this is obtained .The training errors (both the classification and the regression errors) were well below this after 100,000 iterations. In the final demo fish detection took only around 0.7 seconds for each image frame.


(a) "Input Image" (b) "Network Output"

Figure 5.1: correctly predicting the class(8) of the fish with high class score

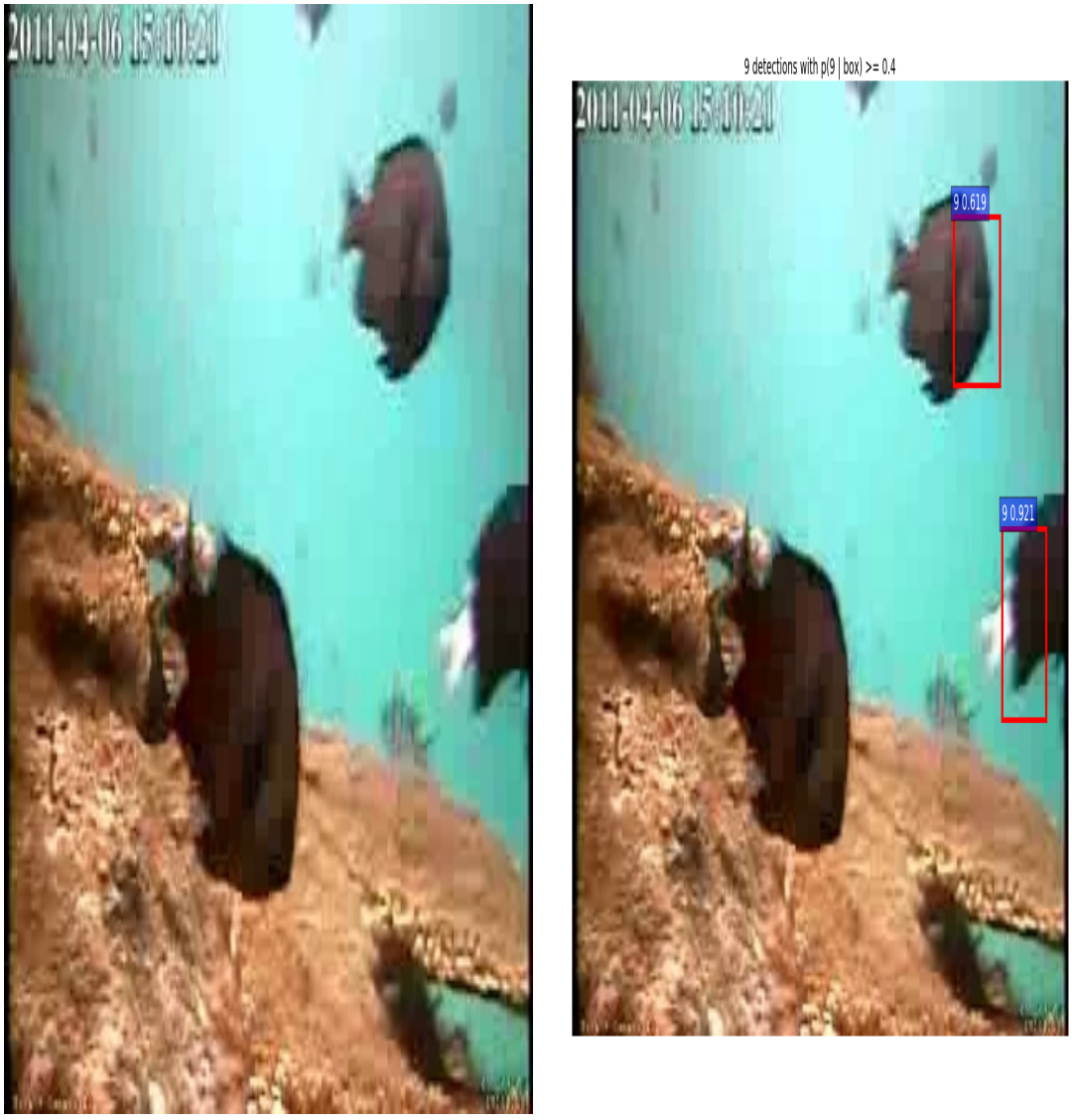

(a) "Input Image" (b) "Network Output"

Figure 5.2: correctly predicting the class(9) of the fish with high class score



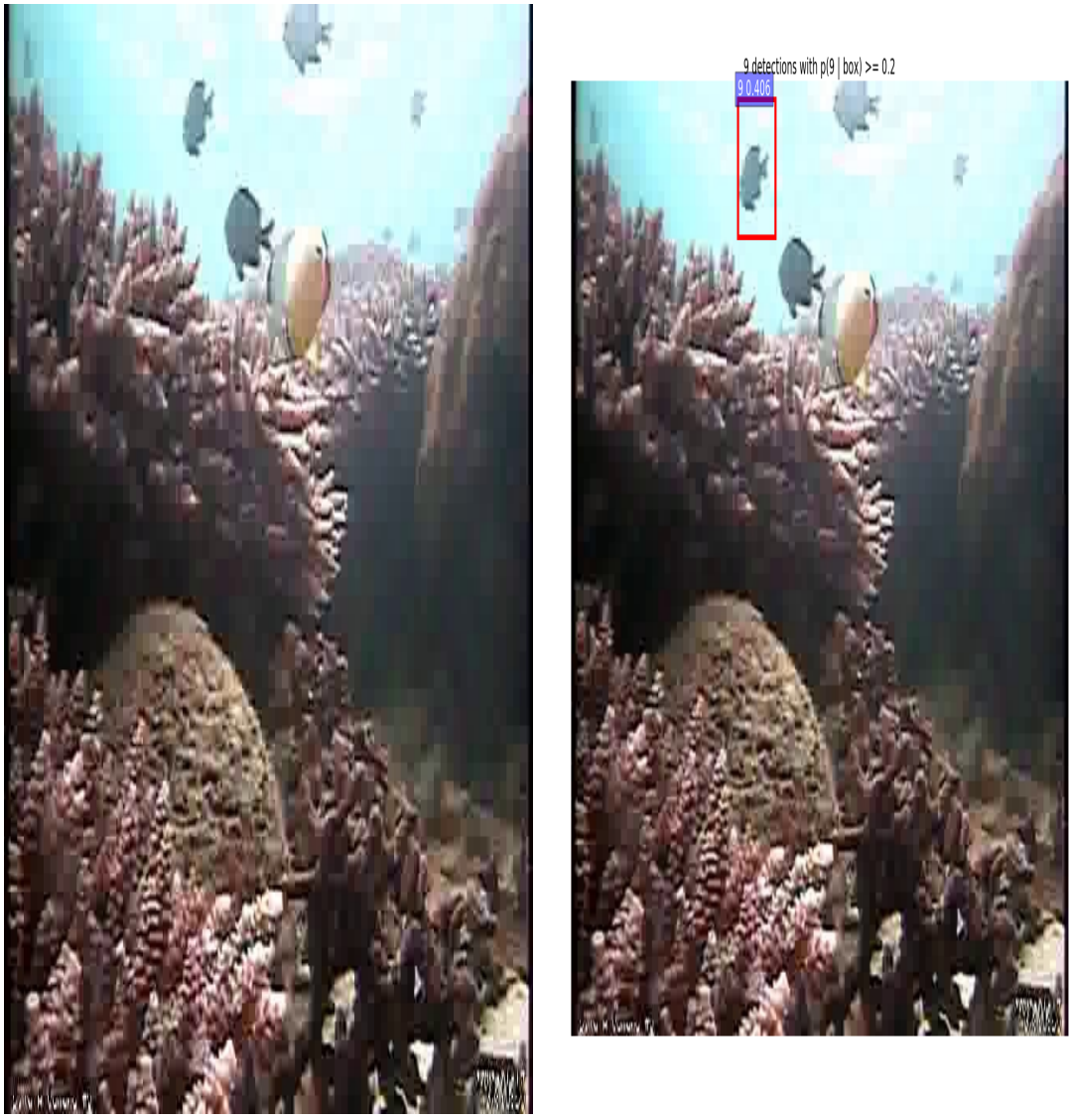

(a) "Input Image" (b) "Network Output"

Figure 5.3: correctly predicting one out of two fishes present in the image




(a) "Input Image" (b) "Network Output"

Figure 5.4: correctly predicting the class but high region proposal error




(a) "Input Image" (b) "Network Output"

Figure 5.5: Incorrect Class prediction

# Bibliography

A.Khosla, Yao, B. and Fei-Fei, L. (2014). Integrating randomization and discrimination for classifying human-object interaction activities, in human-centered social media analytics.

Branson, S., Wah, C., Schro, F., Babenko, B., Welinder, P., Perona, P. and Belongie, S. (2010). Visual recognition with humans in the loop, *Springer* **6314**: 438–451.

Farrell, R., Oza, O., Zhang, N., Morariu, V., Darrell, T. and Davis, L. (2011). Portmanteau vocabularies for multi-cue image representation., *IEEE International Conference on Computer Vision (ICCV)* pp. 161–168.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. and Ramanan, D. (2010). Object detection with discriminatively trained partbased models, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *CVPR* pp. 580–587.

He, K., Zhang, X., Ren, S. and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition, *European Conference on Computer Vision(ECCV)* .

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (n.d.). Caffe: Convolutional architecture for fast feature embeddingâĂİ, year=.

Khan, F., van de Weijer, J., Bagdanov, A. and Vanrell, M. (2011). Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance., *NIPS* pp. 1323–1331.

Kuan, D., Parikh, D., Crandall, D. and Grauman, K. (2012). Discovering localized attributes for fine-grained recognition., *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 3474–3481.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner., P. (1998). Gradient-based learning applied to document recognition., *IEEE* .

Matai, J., Kastner1, R., Jr., G. R. C. and Demer, D. A. (2012). Automated techniques for detection and recognition of fishes using computer vision algorithms, *NOAA Technical Memorandum NMFS-F/SPO-121* .

Matsugu, M., Mori, K., Mitari, Y. and Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network, *IEEE* **16**: 555–559.

Pornpanomchai, C., Lurstwut, B., Leerasakultham, P. and Kitiyanan, W. (2013). Shape- and texture-based fish image recognition system, *Kasetsart Journal-Natural Science* **47.4**: 624–634.

R.Girshick (2013). Fast r-cnn, *IEEE International Conference on Computer Vision(ICCV)* .

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. (n.d.). Imagenet large scale visual recognition challengeâĂİ, journal =.

Saitoh, T., Shibata, T. and Miyazono, T. (2016). Feature points based fish image recognition, *CVPR* .

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks, *International Conference on Learning Representations(ICLR)* .

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, *International Conference on Learning Representations(ICLR)* .

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D. and Rabinovich, A. (n.d.). Going deeper with convolutionsâĂİ, journal =.

Szegedy, C., Reed, S., Erhan, D. and Anguelov, D. (n.d.). Scalable object detection using deep neural networksâĂİ, journal =.

Uijlings, J. R., van de Sande, K. E., Gevers, T. and Smeulders, A. W. (2013). Selective search for object detection, *International Journal of Computer Vision(IJCV)* .

Wah, C., Branson, S., Perona, P. and Belongie, S. (2011). Interactive localization and recognition of fine-grained visual categories, *IEEE International Conference on Computer Vision (ICCV)* .

Yao, B., G.R.Bradski and Li, F. (2012). A codebook-free and annotation-free approach for fine-grained image categorization, *CVPR* pp. 3466–3473.

Yao, B. and Li, F. (2010). Grouplet: A structured image representation for recognizing human and object interactions., *CVPR* pp. 9–16.

Zitnick, K. L. and Dollar, P. (2014). Edge boxes: Locating object proposals from edges, *European Conference on Computer Vision(ECCV)* .

# Curriculum vitae

Roshan Pati
C-319,Patel Hall,IIT Kharagpur

## Research Area

- **Image processing and computer vision:** Deep Neural Networks for Simultaneously Localizing Fishes and Recognizing their Species in Underwater Videos
- **Machine learning:** Vehicular Environment Detection for Autonomous Driving.
- **Data Analytics:** Building correlation between social media sentiments with real time stock trends.

## Education

**BTech** Department of Electrical Engineering — Jul. 2013 - May. 2017
*Indian Institute of Technology Kharagpur, India*
**Major:**Electrical Engineering