Unsupervised Underwater Fish Detection Fusing Flow and Objectiveness

David Zhang Giorgos Kopanas* Chaitanya Desai Sek Chai Michael Piacentino
SRI International, Princeton, NJ 08540 *University of Thessaly, Greece
{david.zhang, Chaitanya desai, sek.chai, Michael.piacentino}@sri.com *kopanas@uth.gr

Abstract

Scientists today face an onerous task to manually annotate vast amount of underwater video data for fish stock assessment. In this paper, we propose a robust and unsupervised deep learning algorithm to automatically detect fish and thereby easing the burden of manual annotation. The algorithm automates fish sampling in the training stage by fusion of optical flow segments and objective proposals. We auto-generate large amounts of fish samples from the detection of flow motion and based on the flow-objectiveness overlap probability we annotate the true-false samples. We also adapt a biased training weight towards negative samples to reduce noise. In detection, in addition to fused regions, we used a Modified Non-Maximum Suppression (MNMS) algorithm to reduce false classifications on part of the fishes from the aggressive NMS approach. We exhaustively tested our algorithms using NOAA provided, luminance-only underwater fish videos. Our tests have shown that Average Precision (AP) of detection improved by about 10% compared to non-fusion approach and about another 10% by using MNMS.

1. Introduction

Fish stock assessments using underwater camera systems are becoming more popular in recent years because cameras are environmentally friendly, and they need less use of extractive and obtrusive midwater and bottom trawling. Such underwater camera systems often carry multiple cameras that capture real-time videos. Data are collected at hundreds of locations throughout the stock's range annually, and the vast amount of videos pose a significant challenge to manual processing.[1] Image processing algorithms have been used to detect, segment and track fishes.[2-4]. However, fish stock assessment is still processed manually, with an operator selecting an image region by pointing and clicking frame to frame to measure fish size, which are correlated with fish age and growth rate. This workflow amounts to hundreds of hours for fish stock assessment, which can be automated with robust detection and segmentation algorithms for underwater fishes.

Basic methods for fish detection in pixel-based processing include change detection and background subtraction methods. The change detection method depends on the difference of the alignment of current frame and reference frame; the background subtraction method needs to build a background model e.g., Gaussian Mixture Model (GMM), and the foreground is the difference of the current frame to the background model. These methods work relatively well when cameras are close to stationary and the scene is not crowded. In moving platforms such as underwater vehicles, however, 3D parallax from the scene, slow changing foreground, illumination change, and low light noise pose challenges to these image processing methods. False alarms and missed detections limit the recall and precision rate. Detections based on optical flow discontinuities, though the segmentation is simple and straightforward, poses the same problems as statistical and change detection based methods if they are used independently.[5]

Recent employment of deep learning neural networks in object detection improves the performance in all the aforementioned challenge areas. ImageNet using ConvNet can classify about 10 sea fish species out of 1000 general objects from static images [6] Anderson et al. discussed fish classification using CNN and RBM for underwater videos.[7] However, much more work is needed to improve the current tedious stock assessment: (1) Need an automated segmentation method to output baseline annotations of fish regions per frame. This is an immediate need considering hundreds of terabytes of unprocessed videos and gigabytes of new data generated daily. Given the baseline annotations, with the help of fish tracking, users can annotate fish species from key frames with great ease. (2) A public large-scale fish dataset with species annotations is needed for deep learning training and classification. (LifeCLEF[8] is a relatively small marine dataset on coral fish species identification). (3) A framework needs developed to allow online video processing on fish detection, tracking, type classification, activity analysis and size measurement, and other statistical assessments.

In this paper, we will address the first need, to develop an unsupervised underwater fish detection method. The idea is to generate automated training samples based on the fusion input from flow segmentation and object proposals. After training, the detection of fishes per frame based on the training models is automatic. Since the training is unsupervised, the algorithm can be implemented in the underwater capturing systems as both training and classification will be realized in real-time while capturing. Our contributions are twofold: (1) we propose a way to train deep NN to classify the fish class without the need of manual annotation. We generate a large set of training samples using strictly constrained flow detection. Although about 30 percent of false annotations of fish samples include fish parts, overlapped fishes, occlusions and backgrounds in the training set, we found that the trained network gives satisfying results in defiance of false annotations. The key to generating a good classifier is to provide enough (~100,000) weighted training samples to counteract influences from the noise and consistently use object region proposals during the training and detection procedure. (2) A variety of recent papers offer methods for generating category-independent region proposals.[9,13-16] We show that by fusing the proposed regions with the regions from motion flow, we can improve both recall and precision rate by a large margin compared to the respective region approaches in classification. Furthermore, we modified Non-Maximum Suppression (NMS) based on the observation that multiple regions are around a single object and by boosting the scores of the enclosed small regions we improve the detections. This fusion approach improves Average Precision by a large margin compared to the respective image processing method or learning method. We tested the proposed unsupervised object detection method with luminance-only fish data sets from NOAA.

We will briefly discuss other work on learning related object detection and NMS in Section 2. The overview of our proposed work is summarized in Section 3. The descriptions of our contributions in the training and classification of fishes, including fusion of proposals from Selective Search[9] and flow segmentation regions , and modified Non-Maximum Suppression (MNMS), are described in details in Section 4.

2. Related work

Celik et al.[10] and Nair et al.[11] have had good reviews on the unsupervised online learning methods for object detection. Most of them use motion as the primary cue to collect information. In their work, a background model was used in collecting motion based samples. Nair et al. designed an automatic labeler that thresholds the difference between the current frame and the previous averaged frames that do not contain significant motions. The labeling of positive samples is poor. Their pedestrian detector is a pre-defined window detector based on the human size ratio. Celik et al. used two stage detectors: a coarse detector to find samples, and a fine detector to remove bad positives. In their fine detector, spectral clustering based on the similarity of the collected samples is adopted to remove the bad samples. Wu et al.[12] presented an online re-training of a detector based on the outputs of a coarse detector using boosting. They used a weighting mechanism to penalize less confident samples. An interesting work from Ren et al.[20] in their R-CNN paper, suggested that if the negative samples dominate, it is possible to optimize the lost function towards negative samples. Although their paper still samples positive and negatives with a 1-to-1 ratio, we adopted the idea in our work.

For object detection using deep learning, region proposals are a popular pre-processing step. The idea is to generate multiple overlapping segmentation windows. This shifted from an earlier effort that focused on diversity of segmentations as opposed to a single and perfect segmentation. In our work, we use Selective Search as the region proposal.

Another segmentation work of interest is the video segmentation as regional candidates. Grundmann et al. [17] developed a hierarchical graph-based spatial-temporal segmentation algorithm. It begins by oversampling a volumetric graph into space-time regions by appearance. A region graph is iteratively generated over multiple levels to create a tree of spatio-temporal segmentations. The algorithm takes tens of seconds per frame, and thus is not practical in online detection.

Most deep learning algorithms generate confidence score of each proposed region. Simple thresholding does not generate good detection results. Malisiewicz et al. adapted an aggressive NMS approach from to Pedro Felzenszwalb's version reduce region proposals.[18] They sort all the regions first and start with the region of highest confidence. To reduce redundant regions, any other regions that overlap with this region with the overlap percentage larger than a threshold are removed. This region is then regarded as the first valid object. The algorithm then goes to the next highest confidence region remained in the list, and apply the same operations to generate the second object and so on. The problem with this algorithm for our approach is that it does not always generate the best detection.

3. Overview of our approach

The online training and detection of underwater fish in our proposed solution are summarized by the scheme in Figure 1 and 2 respectively.

3.1. Fish training

In training (Figure 1), we first generate moving object segmentation. Each moving blob is labeled as an object candidate and together they composite the ground truth. We define these ROI as motion region, mr_{reg}^{fr}, where *fr* is the frame id, and *reg* is the region id. For every frame, we run Selective Search to get class-independent *proposed regions*, $\left\{pr_{reg}^{fr}\right\}$. Then for every region pr_i^N belongs to frame *N*, we compute the maximum overlap with every mr_j^N, if the maximum overlap is below a low threshold this *region* is annotated as a negative sample, else if the overlap is higher than a high threshold this *region* is annotated as a positive sample. The assumption here is the object proposals are of more objectiveness than the flow regions. Small overlaps are with high probability the parts of the objects, and large overlaps are regarded as the entire objects. We drop a big number of proposed regions that their maximum overlap with the moving blobs is between the low and high threshold since we don't have sufficient information about their class. Furthermore, the sample generation in training and classification from Selective Search ensures sampling consistency which leads to performance improvement. We modified the top layer (fc8) of CaffeNet from Convolutional Neural Networks (CNN) that was first used in Krizhevsky et al. to output a binary classification.[19-21] More details for the training process will be discussed in 3.3.

Figure 1: Block diagram of the proposed online training method.

3.2. Fish classification

The real-time classification algorithm is summarized as follows (illustrated in Figure 2):

1. Run Flow Segmentation and Selective Search to generate fused region proposals.
2. Classify all proposed samples in CNN to compute confidence.
3. Apply Modified Non-Maximum Suppression (MNMS) to find unique regions per object.
4. Output detected regions.

3.3. Training process

Our training process is based on the procedure that was done for the R-CNN by Girshick et al.[19] In our work we show that it is possible to train a NN with an adequately noisy training set if the training set is big enough and contains good annotation.

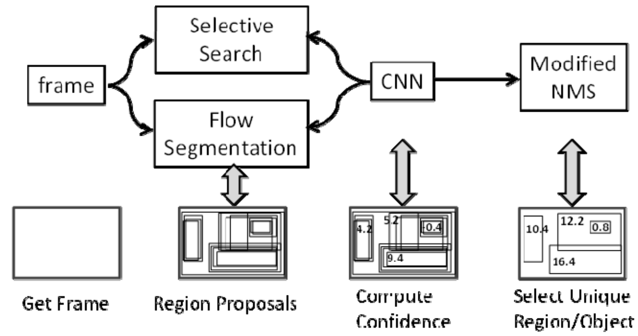


Figure 2: Block diagram of the proposed online object classification method.

In initialization, we used a pre-computed CNN that matches Krizhevsky et al. performance and architecture.[21] The CNN was trained by using the ILSVRC 2012 dataset. We then fine-tuned the CNN to match the new domain. We replaced the 1000 class top-layer (fc8) with a randomly initialized 2 class layer. We continued the Stochastic Gradient Descent (SGD) from that point while tuning the last layer's learning rate to be 10 times faster than the pre-trained layers. For every SGD iteration, we create a 128-image batch that consists of positive ROIs and negative ROIs in a ratio of 1:3 respectively. The annotation of the training set is automatically generated with the details explained in Section 3.1. The number of both positive and negative training samples is in the range of 100,000. We trained the CNN for 80,000 iterations until the AP of the validation set converged. The feature of the training negates the detection of parts of objects, however, it does not negate multiple and overlapped object regions. Because of that, in order to get a best region for an object, we modified the NMS algorithm to select single objects.

To validate the auto-annotation training, we manually picked 20,000 samples from a full set of auto-generated samples and ran the training in CNN. The precision-recall rate is compared by running an identical test video set from two separate trained models shown in Figure 3(a). The manual annotation only outperforms less than 5% at 80% recall. To prove that sampling consistency is important, we trained CNN with samples from flow segmentation only versus using additional Selective Search to filter samples (see Section 3.1). In Figure 3(b), This training was done (blue curve) by using the flow segmentation blobs as positive class training samples and random ROIs that don't overlap with the segmentation blobs as negative class training samples. The red curve is the modified training process by generating training samples from Selective Search using the overlap criteria.

The sampling consistency performs much higher precision at all recall rates.

4. Combing regional proposal networks with flow segmentations

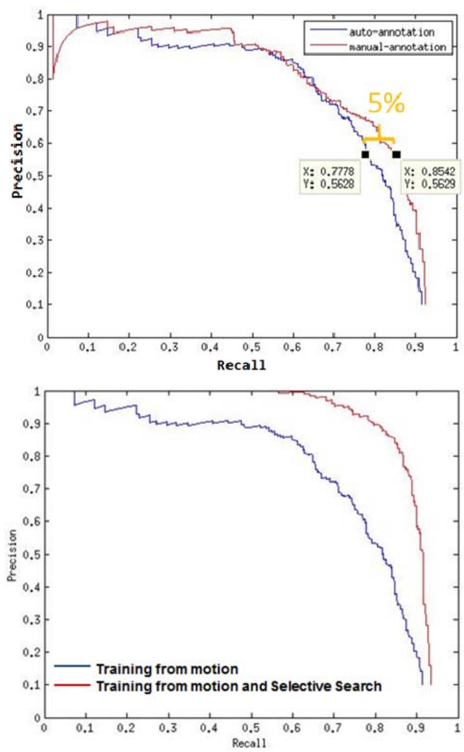


Figure 3: Top. The precision-recall graph compares manual-annotation training with auto-annotation training from the same videos. The small discrepancy of performance demonstrates the value of using auto-annotation in real-time training. Bottom: The blue curve is the performance of the classifier using samples from flow segmentations. The red curve shows at least 20% better precision at 80% recall from the classifier that its flow samples are tuned by Selective Search during training.

In this section we introduce key functions used in training and detection of dominant moving objects.

4.1. Flow segmentation

We have used optical flow to detect fish in the hazy underwater environment in which objects about 10 meters away are obscured. We generate a binary mask from the optical flow computation. Fish motion is deformable thus flow is not oriented and uniformly distributed from an object. Thus we generated a mask from the amplitude of motion and uncertain pixels enclosed by the moving pixels are filled up. Based on the connectivity of the valid pixels, blobs are extracted. Each blob is an object candidate. Besides checks of blob size, aspect ratio, we also compute center displacement of each blob. If the flow projected blob center in the previous frames is inconsistent with the blob centers in these frames, then the blob is not a candidate. The block diagram of the flow based segmentation is summarized in Figure 4.

In the training stage, the binary threshold is set such that the precision rate is relatively high. Some moving targets are missed, but the blob regions have less probability to be the valid moving pixels are filled up. To generate positive samples, the motion candidates are examined against Selective Search proposals as described in Section 3.1. The proposed samples from Selective Search are

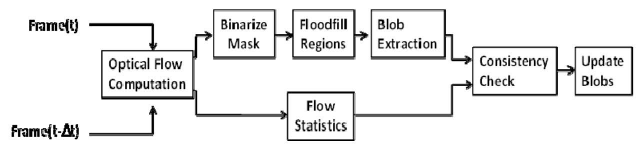
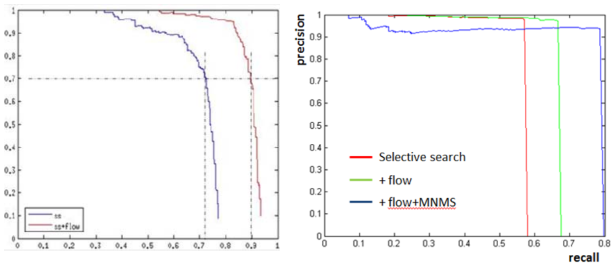


Figure 4: Block diagram of the proposed online training method.

generated by the criteria of overlap percentage. In classification stage, this constraint is relaxed. We have made the recall rate relatively high so that no moving object is missed. In consistency, Selective Search is still used to select input samples. Nevertheless, we rely on the deep learning NN to classify the proposed regions.

4.2. Fusion of region proposals

During the detection time we run Selective Search to obtain objective candidate ROIs. In addition to the motion segmentation from which moving targets are segmented, Selective Search provides potential candidates of near-static objects, which flow segmentation has missed; Selective Search also pick up overlapped objects that flow segmentations sometimes fail to recognize. Additionally, it provides more complete window candidates around an object. In cases when flow segmentation samples are part of the deformable objects or it miss-samples overlapped objects, Selective Search can provide supplementary ROIs. It is worth mentioning that in previous literature the use of the detector was to parameterize this algorithm to suggest all possible objects without taking into consideration the number of proposed candidates. However, it is unrealistic to propose ~2,000 regions per frame that provide 100% recall rate from Selective Search for real-time detection, and secondly, the fact that our CNN was trained with a noisy training set makes it vulnerable to extremely high numbers of proposals. Fortunately, we observed that we just need to keep the crowd of the regions close to 100 while keeping the recall rate close to 90% in the fused proposal. Figure 5(a) shows precision-recall rate on one of the test video sequences. The recall of Selective Search only regions at precision=70% is 72%, but the fused region proposals gives 90% recall rate from most of the sequences.



(a) (b)

Figure 5. (a) In classification, Region proposals with fusion (*ss+flow*, *red* line) against pure selective search (*ss*, *blue* line). (b) *Red*: Selection Search only. *Green*: fusion of Selective search and flow segmentations. Both of them use NMS. Blue: Modified NMS on fusion proposals.

Although the reduced region proposals do not provide 100% recall rate as most of the region proposal based trainings do, we think it is a good compromise in the real-time detection. As we discuss in Section 5, the training and NMS are the other important functions that determine the precision and recall rates of the detection.

4.3. Modified Non-Maximum Suppression

The modified NMS boosts the enclosed regions before aggressive suppressions. This algorithm is based on the observation that parts of objects have much less confidence than the entire object in the training result and multiple regions surrounding the object may have similar scores. Therefore, after the regions of parts of an object are rejected by a single threshold, a region closest fit to a full object shall be selected even its score may not be the highest among all the region candidates around the object. In this section, we use an example to explain the modified NMS algorithm.

In Figure 6, region candidates are marked with black, green, red, blue and yellow. Let's assume the confidence score from the CNN detection is also in this order. The overlap percentage of the green with respect to (black $\cup$ green) is p1, and that of the blue to (black $\cup$ blue) is p2, and p2 < p1. From NMS[18], if the overlap threshold is set to be thr>p1, then black, green and blue regions are regarded as objects; if p2<thr<p1, then black and blue regions are computed as objects (Fig. 6b); if thr<p2, only the black region is selected (Fig. 6c). In all these cases, the optimized selections, the red and yellow regions are not selected.

We modified the original MNS by boosting the enclosed region with scores from the regions that surround this region. The boost equation is expressed as follows:

$$P_j = P_j + \sum_i O(i,j) \cdot P_i \cdot \delta\big(A(j) < A(i)\big) \qquad (1)$$

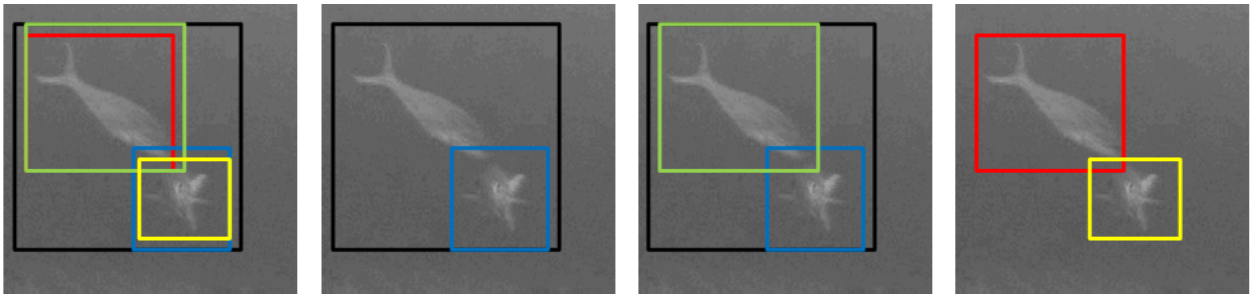
where P_i and P_j are confidence scores of region i and j respectively. $O(i,j)$ is the overlap percentage of the two regions: $O(i,j) = \frac{overlap(i,j)}{\min(A_i, A_j)}$. The δ() function is 1 when the area of region j is smaller than that of region i, and 0 otherwise. For example, the blue region boosts it score from overlapped regions of black, green and red, and the yellow boosts from black, green, red and blue. The weight of contribution is based on the overlap percentage. The boost equation changes the order of confidence scores of the 5 regions to: red, green, yellow, blue and black. The confidence order of red and yellow or green and yellow is not important.

Because the modified NMS guarantees the smallest enclosed regions have the highest scores, after sorting, these regions are among the highest score of regions that NMS first pick them as top seeds. They can then suppress other overlapped regions. The result of the new approach is in Figure 6(d).

5. Experiments

In our experiment we run flow segmentation of 30Hz 720x480 frame-size videos on a dual core CPU at 2.5GHz. Image training and classification is processed in parallel on an *Nvidia* Titan Black GPU. Both flow computation and CNN classification are under 50ms. Selective search on ~200 regions averages about 0.5s. In Moving Object Training (Section 3.1), low threshold of overlap percentage is set to be 20%, and high threshold to be 60%. Videos were provided by NOAA. In one of the sequences, 100,000 samples were generated from the first 20 minutes of video. Fish detection was tested from a different camera monitoring the same area. Figure 7 shows steps of fish detection. In 7(b), fish regions are first proposed by Selective Search. Flow segmentations are then added as additional sample proposals in 7(c). Detection scores are displayed in 7(d). A simple threshold removes negative scores. We compared original Non-maximum suppression (NMS) in 7(e) and modified NMS result in 7(f). The yellow regions that were incorrect from the original NMS is fixed in our approach. MNMS with fusion proposals gives best results. In less crowded sequences, the average precision (AP) of the proposed solution is larger than 90%.

Figure 5(b) shows the performance of the fused proposal compared with Selective Search only and that with the addition of the modified NMS. AP of the Selective Search only = 56%, and AP of the fused proposal (green curve) = 66%. Both curves use the original Non-maximum Suppression to reduce the proposed regions. The blue curve is the fusion proposal using Modified NMS, AP = 75%, which is about another 10% improvement. The modified NMS has a slighter less precision at small recall rate (< 8%), but increases recall rate by about 20% while keep high precision.



(a) (b) (c) (d)

Figure 6. (a) 5 proposed region after negative regions are removed. (b) Result from original NMS when p2<thr<p1. Black region encloses both fishes. (c) Result from original NMS when thr<p2. (d) The result from our approach (MNMS).

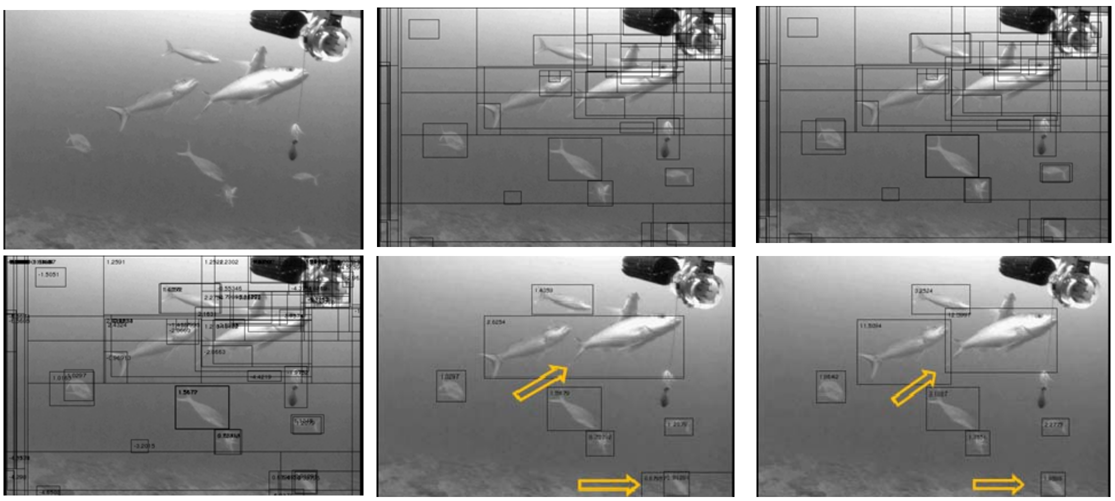


Figure 7. (a) top left: raw input (b) top middle: regions from Selective Search (c) top right: fusion of Selective Search and motion flow segments (d) bottom left: confidence scores on the fused regions (e) bottom middle: thresholding and original NMS (f) thresholding and modified NMS.

6. Conclusion

In this paper we have proposed an unsupervised underwater fish detection approach based on deep learning method in response to the immediate need of an automated segmentation method to output baseline annotations of fish regions for abundance estimate. We focused on an approach that needs no manual annotation and can lead to real-time online training after speed optimization. We used motion flow segmentations to configure and generate candidates, and then apply Selective Search to reduce false labeling in training. We also biased sampling in favor of more negatives in order to provide a robust classification model. In detection using CNN, we fuse region proposals from Selective Search and motion segmentation, which yields better performance than the respective individual proposals. Modified NMS optimizes the suppression towards a full object while reducing large amount of overlapped windows.

The framework is particularly useful for video surveillance in oceanic environment, airport, security in school, and government facilities. The online training can be skipped if a pre-trained model exists for the same environment. Nevertheless, updated online training can use the trained model from the last trip as initialization for training.

The detection performance can improve on objects, which are occluded or only part of the object is captured in the images. Further improvement includes refined training with motion cues on overlapped objects and training with temporal activities of moving objects. Future work can include object tracking which can improve the overall performance.

7. Acknowledgements

We want to acknowledge both NOAA (National Oceanic and Atmospheric Administration) and FAU (Florida Atlantic University) as the results presented in this paper were partly developed under a program with NOAA and FAU. Portions of this research are also sponsored by the AFRL and DARPA (contract FA9453-15-C- 0056).

References

[1] http://www.nmfs.noaa.gov/stories/2012/05/05_23_12stock_assessment_101_part1.html

[2] M. Chuang, J. Huang, K. Williams and R. Towler, "Automatic fish segmentations via double local thresholding for trawl-based underwater camera systems", in Proc. *IEEE ICIP*, 2011.

[3] H. Yao, Q. Duan, D. Li, J. Wang, "An improved K-means clustering algorithm for fish image segmentation", *Math and Com modelling*, V58, I3-4, pp790-798, August, 2013.

[4] S. E. Grasty, "Use of a Towed Camera System for Estimating Reef Fish Populations Densities on the West Florida Shelf", thesis, Univ. of South Florida, 2014.

[5] D. Zhang and G. Lu, "Segmentation of Moving Objects in Image Sequence: A Review ", Circuits Systems Signal Processing, Vol. 20, No.2, 2001, Pp. 143-183.

[6] A Krizhevsky, I Sutskever and GE Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, 1097-1105.

[7] J. Anderson, B. Lewis and C. O'Byrne , "Intelligent Fish Classification in Underwater Video", http://web.cse.ohio-state.edu/~andejose.

[8] http://www.imageclef.org/2016

[9] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, "Selective search for object recognition", ICCV, 2013.

[10] H. Celik, A. Hanjalic, E.A. Hendriks, S. Boughorbel, "Online Training of Object Detectors from Unlabeled Surveillance Video", in *Proc. IEEE CVPR*, 2008.

[11] V. Nair, J.J. Clark, "An Unsupervised, Online Learning Framework for Moving Object Detection", in *Proc. IEEE CVPR*, 2004.

[12] B. Wu and R. Nevatia, "Improving part based object detection by unsupervised, online boosting," in *Proc. IEEE CVPR*, 2007.

[13] B. Alexe, T. Deselaers and V. Ferrari, "What is an object", in *Proc. IEEE CVPR*, 2010.

[14] J. Carreira and C. Sminchisescu, "Constrained Parametric Min-Cuts for Automatic Object Segmentation", in *Proc. IEEE CVPR*, 2010.

[15] S. Manen, M. Guillaumin, L. Van Gool, "Prime object proposals with randomized Prim's algorithm", ICCV, 2013.

[16] F. Li, J. Garreira and C. Sminchisescu, "Object Recognition as Ranking Holistic Figure-Ground Hypotheses", CVPR, 2010.

[17] M. Grundmann, V. Kwatra, M. Han and I. Essa, "Efficient Hierarchical Graph Based Video Segmentation",in *Proc. IEEE CVPR*,2010.

[18] T. Malisiewicz, A. Gupta, A. A. Efros, "Ensemble of Examplar-SVMs for Object Detection and Beyond", ICCV, 2011.

[19] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proc. IEEE CVPR*, arXiv:1311.2524, 2014.

[20] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", in *Proc. IEEE CVPR*, arXiv:1506.01497, 2015.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *NIPS* 2012.