DEVELOPING AN AUTOMATED ANALYSIS OF FISH MIGRATION VIDEO USING COMPUTER VISION ALGORITHMS

By

JULIA ENGDAHL

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Oceanography

Written under the direction of

Nicholas Beaird

And approved by

____

____

____

____

New Brunswick, New Jersey

October 2020

ABSTRACT OF THE THESIS

DEVELOPING AN AUTOMATED ANALYSIS OF FISH MIGRATION VIDEO USING COMPUTER VISION ALGORITHMS

by JULIA ENGDAHL

Thesis Director:

Nicholas Beaird

Dams, fishways, and other hydropower structures have been utilized for centuries and have largely been unevaluated regarding their effects on the surrounding fish populations. Conventional methods (seining and angling) used for such evaluation are often costly and time consuming. The aim of this project is to decrease analysis time by automating fish detection with the use of video monitoring and evaluating two computer vision algorithms: background subtraction and machine learning algorithm You Only Look Once (YOLO). We evaluated these algorithms on video data collected from the Island Farm Weir on the Raritan River in New Jersey. Our results indicate that background subtraction models need to be adaptive with respect to time due to the dynamic aquatic environment, and YOLO needs to be trained and tested on the user's specific case study dataset for optimal results.

Table of Contents

List of Tables

List of Figures

Introduction

Commercial, recreational, and subsistence fishing have long supported communities along the U.S. east coast. Many important coastal and inland fish species in this region are diadromous (i.e., different stages of their life cycle are spent at sea or in freshwater). Management of these fisheries is paramount to a successful blue economy, but many have seen declines due to industrialization, riverine pollution, and the development of hydropower structures (Keen et al., 2018). Dams are a major barrier to the spawning migrations of anadromous fish species (i.e., a type of diadromous fish that spawn in freshwater but spend their adult lives in the ocean) (Xu & Matzner, 2018). Anadromous species are important for coastal and inland fisheries and studying the effects of such impediments on these populations is a critical component of management and conservation. It is imperative to develop a method that evaluates these effects in a timely manner, especially if the species in question is endangered, so that reports, protections, and other management decisions can be enacted with minimal delay (Salman et al., 2019; Xu & Matzner, 2018). To restore depleted anadromous fish populations, many dams have been removed or equipped with fish passage devices (fishways), that allow upstream migrating fish to bypass dams (Franklin et al., 2012).

Although dams and fishways have been around for centuries, there have been few studies evaluating the effectiveness of fishways on fish passage (Matzner et al., 2017; Salman et al., 2019). New Jersey is no exception to the proliferation of dams and lack of efficacy studies on constructed fishways. The largest river basin (Raritan) in New Jersey has been subject to the construction of five dams, of which three have been removed, including the most downstream structure – the Calco Dam – in 2011 (NOAA & USFWS, 2016). After the removal of the Calco dam, the Island Farm Weir (IFW), located between Manville and Bound Brook, is the most downstream obstruction that migrating species encounter (NOAA, 2016). This low-head dam was equipped with a vertical-slot fish ladder in 1995 and became functional in 1996 (Boriek, 2013). See Figure 1. It is important to study the effects of the fishway at the IFW because American shad (Alosa sapidissima) and river herring (Alosa aestivalis and Alosa pseudoharengus) are species of concern in this region (NOAA & USFWS, 2016), and the results would be integral for further management and conservation.

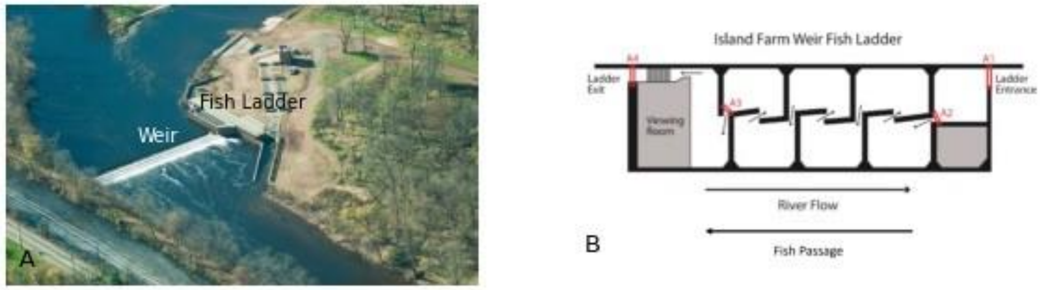

Figure 1. A) Aerial view of the Island Farm Weir and fish ladder. B) Island Farm Weir schematic showing the entrance and exit of the ladder, positions of antenna for the PIT tag detection, viewing room camera location, and the direction of river flow and fish passage. Images modified from Vastano et al. (2018).

American Shad and River Herring Restoration

Historically, large numbers of American shad would migrate up the Raritan River to their spawning grounds (NOAA & USFWS, 2016; Darrow & Neilan, 2013). This migration was integral to the local economy during early human settlement in the area (Darrow & Neilan, 2013). After the 1900s, when the area became heavily populated, shad numbers decreased due to the construction of dams and overfishing (Darrow & Neilan, 2013). Attempts to rebuild the shad population include stocking efforts, fishway construction, recreational and commercial fishing limits, and dam removal (NOAA & USFWS, 2016; Darrow & Neilan). There have been few reports on the efficiency of the IFW fish ladder conducted by the New Jersey Department of Environmental Protection (NJDEP) up until 2012, and continued evaluation has been conducted by the Jensen Lab at Rutgers University from 2012 to present. Field observations have been the best way to evaluate fish passage (Vastano et al., 2018). Conventional methods used to evaluate the success of fish passage include capturing fish using a 300-foot-long seine net and angling (hook and line), in addition to video monitoring (Vastano et al., 2018). American shad and river herring that are sampled via physical capture methods undergo surgical insertion of a passive integrated transponder (PIT) tag. The PIT tag gives the individual a unique identification number and when they near the fish ladder the tag is detected by a series of antennae throughout the fish ladder (Vastano et al., 2018). These sampling methods are often carried out three to four times a week during the sampling season (Vastano et al., 2018). Over a six-year period from 2012 to 2017, a study performed by Vastano et al. (2018), tagged a total of 50 American shad and 116 river herring. These small sample sizes make analysis of fish behavior difficult. In that same six-year period, IFW video data was manually analyzed and there were over 27,000 recorded observations of fish passing through the fish ladder. Because the video data captures nearly all fish traversing the fish ladder, it can be used to calculate annual run size, arrival time, determine large-scale patterns in fish movements, and this long record allows for multi-year analyses and comparisons (Vastano et al., 2018). These methods vastly increase the data coverage but are time consuming and challenging to analyze.

Video equipment with infrared lighting at the fish ladder allows for 24/7 monitoring during the spawning season, accumulating well over 2,160 hours of video footage per year (Vastano et al., 2018). Manual analysis (i.e. species identification, fish count, and passage timestamp) is highly effective, but difficult, and analysis of video from one spawning season requires about 720 hours of technician effort (it takes approximately 8 hours to view a 24-hour section of video).

The goal of this project is to decrease manual labor time by automating the video analysis process using computer vision, a technique used to enable computers to gain high-level information from input images analogous to how the human brain analyzes images, to perform object detection. To accomplish this, we implemented and evaluated two computer vision algorithms: background subtraction using OpenCV-Python (Bradski, 2000) and a machine learning algorithm, You Only Look Once (YOLO) version 3 (Redmon et al., 2016).

IFW Video Data

The IFW fish ladder is a vertical slot fishway, with a viewing window located by the exit of the ladder on the upstream side. One side of the window is in contact with the river flow, while the other side is a dry chamber (Figure 1B). The dry chamber houses an infrared-capable security camera (Speco model# CVC-627B), paired with a mini DVR (ATV DYK14G 1-Channel Mini DVR) that records to a 32GB SD card. Additionally, an infrared weather resistant LED spotlight (11", 18 Watt, 850 nm wavelength, superbrightLEDs Model# LBIR-850-35) was installed above the viewing window to provide further illumination of the water column seen through the viewing window. This allows recording to take place continuously during the sampling season (mid to late March until July 1^{st}) when river conditions allow for safe deployment of the video monitoring equipment. Data gaps in the video data may occur throughout the sampling season due to rain events that cause the river discharge to increase such that river conditions no longer allow for safe deployment of the video monitoring equipment. Therefore, the equipment may be removed multiple times per season, for several days at a time. A range of river condition examples captured in the video dataset are presented in Figure 2. In Figure 2A and 2B, the viewing area is uniformly lit, and the water is clear; these good day and good night conditions allow for effective fish detection. However, the aquatic environment is dynamic, and its changes can generate poor fish detection conditions like those seen in figure 2C and 2D. In these images the water is murky, the light shimmering in the water column, and erratic reflections of light are present. These dynamic changes may generate false positive detection if these changes surpass the detection limits defined in the algorithm. This system outputs a sequence of video files with a frame rate of 12 frames per second (FPS) and a duration of about 30 minutes.

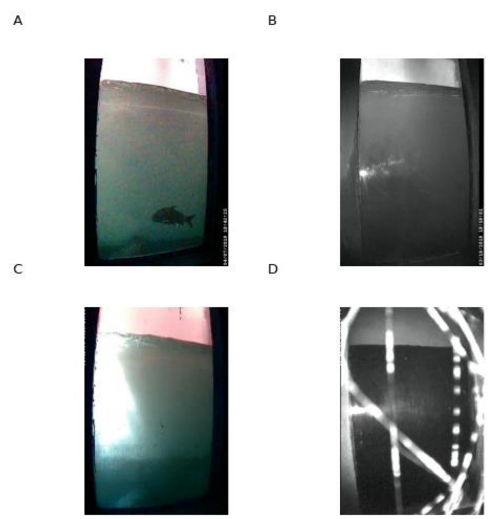

Figure 2. Example images from the IFW dataset. The algorithm performs well during good day (A) and good night (B) conditions when the water is clear, the light is uniform throughout the viewing area, and there is low turbulence. During bad day (C) and bad night conditions (D), when there are large anomalies, such as the shimmering light rays due to high turbulence and the erratic reflections that cause changes larger than the set parameterizations, there will be a drop in the algorithm performance due to the increase in false positive detection.

Background Subtraction Methods

Background subtraction, specifically the frame differencing method, is a relatively simple algorithm for detecting moving objects in each image or video (Amandeep & Goyal, 2015). This is accomplished by initializing a static background frame (i.e. an image where objects of interest are not present) by selecting the first frame (Figure 3A) and converting the color channels to grayscale (i.e. intensities 0 to 255) (Figure 3 color bar). The first frame becomes the baseline that is used to calculate the frame difference between the first frame and subsequent frames (e.g. frame0 – frame 28 in Figure 3C) (Amandeep & Goyal, 2015). This algorithm calculates the absolute difference between the pixels based on grayscale intensities (e.g. detected foreground objects will have a brighter intensity [white] than the background [black]) (Figure 3D) (Amandeep & Goyal, 2015) and it is filtered with a threshold (e.g. 25) on an intensity scale from 0 to 255. Intensities above the threshold are assigned as foreground objects while the intensities below the threshold will be assigned as background (Figure 3E). Thus, based on light intensity differences between two images and a threshold set by the user, pixels are assigned a binary map of 0 or 1, in which background (unchanged) pixels are 0 and foreground (changed or motion) pixels are 1. In addition, a size constraint is added where foreground objects smaller than a certain size (described by a radius of a circle) are excluded from the analysis. The result (Figure 3F), is a recorded .AVI video clip containing the frames where the algorithm detected and contoured those specified changes. This software will continue to record until there are no changes that surpass the specifications (i.e. intensity threshold and radius) for a specified buffer time (e.g. no movement for 12 frames). When the algorithm works correctly, these clips should contain instances when fish are present and exclude times where no fish 'objects' are present, and thus reduce the length of video that needs to be manually evaluated by researchers for fish identification. The background subtraction algorithm is coded using community developed packages in Python, including numpy, pandas, OpenCV, pytesseract and matplotlib.

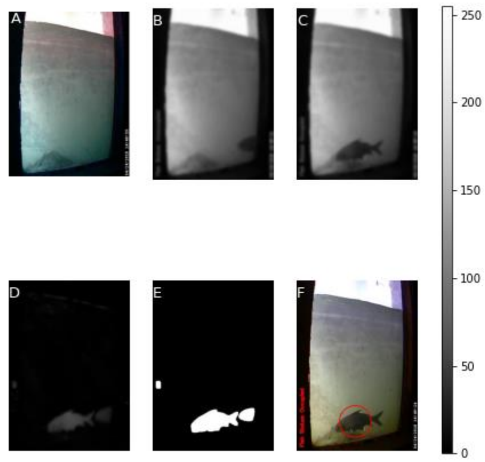

Figure 3. Background subtraction example: A) Raw video file. B) Grayed out first frame. C) Grayed out 28th frame. D) Frame difference from B minus C. E) Threshold of D. F) Contoured video file generated from threshold and radius parameterizations. The range of grayscale intensities, 0 [black] to 255 [white], is presented by the color bar and associated with B, C, and D.

Background Subtraction Results & Discussion

We executed this method using a combination of intensity thresholds [25, 50, 75, 100] and radii [10, 35, 60, 85 pixels] on a video from March 19, 2019 (Table 1). This sample video was manually evaluated frame by frame to test the skill of the background subtraction method. This video contains 1,819 frames where a fish is present and 18,337 frames that do not contain fish, totaling 20,156 manually identified frames. These manually identified frames define the condition positive (P) (i.e., the total number of frames with fish present) and condition negative (N) (i.e., the total number frames without fish present), which are the total number of positive and negative events, respectively. The environmental conditions in this video were clear towards the top of the viewing area and murky at the bottom, with large rays of light shimmering in the water column, as in Figure 2B. Changing the two parameters (threshold, radius) elicit changes in the sensitivity or True Positive Rate (TPR) (e.g. true positives (TP) divided condition positive or TP/P) and specificity or 1 - False Positive Rate (FPR), where FPR is false positives (FP) divided by condition negatives or FP/N) (Fawcett. 2006). When the sensitivity or TPR increases, the specificity decreases (i.e., the FPR increases), and vice versa (Fawcett, 2006).

Threshold	Radius	Total frames	Total frames with fish	Total frames without fish	Algorithm detections	True positives	False positives	TPR	FPR	% Time Reduction
25	10	20156	1819	18337	19392	1782	17610	98	96	3.79
50	10	20156	1819	18337	13499	1589	11910	87.4	65	33.03
75	**10**	**20156**	**1819**	**18337**	**9043**	**1353**	**7690**	**74.4**	**41.9**	**55.13**
100	10	20156	1819	18337	5365	728	4637	40	25.3	73.38
25	35	20156	1819	18337	16384	1639	14745	90.1	80.4	18.71
50	35	20156	1819	18337	11560	1546	10014	85	54.6	42.65
75	35	20156	1819	18337	8164	1234	6930	67.8	37.8	59.50
100	35	20156	1819	18337	4054	694	3360	38.2	18.3	79.89
25	60	20156	1819	18337	15694	1636	14058	89.9	76.7	22.14
50	60	20156	1819	18337	10698	1483	9215	81.5	50.3	46.92
75	60	20156	1819	18337	6260	846	5414	46.5	29.5	68.94
100	60	20156	1819	18337	3093	664	2429	36.5	13.2	84.65
25	85	20156	1819	18337	15280	1636	13644	89.9	74.4	24.19
50	85	20156	1819	18337	10197	1452	8745	79.8	47.7	49.41
75	85	20156	1819	18337	5710	766	4944	42.1	23	71.67
100	85	20156	1819	18337	2680	629	2051	34.6	11.2	86.70

Table 1. Background subtraction results for a range of parameter values. Calculated true positive ratio, false positive ratio, and percent time reduced for thresholds [25, 50, 75, 100] and radii [10, 35, 60, 85] combinations. The row in **bold** is the best parameter combination.

To evaluate the performance of the algorithm classifier (parameter combinations: threshold and radius), we use a receiver operating characteristic (ROC) curve, which is a two-dimensional graphical depiction of the tradeoffs between the TPR plotted on the y-axis and the FPR plotted on the x-axis. Like Matzner et al. (2017), we defined our algorithm performance objectives to be 90% TPR and 30% FPR. These performance objectives are subjective, depending on what tradeoffs between TPR and FPR the user is willing to accept. In our case, we aimed to detect as many fish as possible at the cost of some false positive detections. Ideally, the algorithm would detect 100% true positives, 0% false positives, and reduce the viewing time to only fish events. In ROC space, this perfect classification would yield a point at (0,1) (Fawcett, 2006). This means that the model would have a good measure of separability or distinguishing between positive and negative classes (i.e. the detection of fish). If the model did not have the ability to distinguish between classes (i.e. random classification), the resulting points would land on the diagonal line y = x or the line of no discrimination (Fawcett, 2006), as represented by the black solid line in Figure 4. The parameter combination that results in random classification is threshold 25 and radius 10, which yields 98% TPR and 96% FPR with minimal video time reduction.

Although the small threshold and radius values result in random classification, our sample analysis suggests that the algorithm has some measure of separability given that the majority of the points that make up the ROC curve are above the line of no discrimination when thresholds are greater than 25 and pixel radius greater than 10. The parameter combination that optimizes this method is threshold 75, radius 10 resulting in a 74.4% TPR and 41.9% FPR and reduced the video time by approximately 53.13% (Figure 4). Upon manual analysis of the false positive detections, the conditions that triggered the algorithm were changes in the environment due to the dynamic characteristics associated with aquatic environments (e.g., changing light conditions, unequal spectral propagation, particle scattering, debris floating by such as vegetation, light attenuation, and variations in turbidity). These dynamic changes are often detected as foreground objects (i.e., fish) because they are large and in motion, exceeding the threshold and radius parameterizations. For the scope of this project, we gather that an increase in threshold and radius (i.e., decreasing sensitivity) will decrease algorithm detections, whereas, a decrease in threshold and radius (i.e., increasing sensitivity) will increase algorithm detections (Figure 4). As the true positive rate and false positive rate increase, the percent of video time reduced decreases (Figure 4). The objective is to maximize fish detection; therefore, a good measure of percent time reduction should be based on the FPR (i.e., the resulting video to be analyzed will be majority fish events with minimal false positives). The caveat to this result is that it is from one video and it is only representative of a narrow range of environmental conditions.

To optimize or generalize the results even further, a manual analysis would need to be applied to the broader dataset.

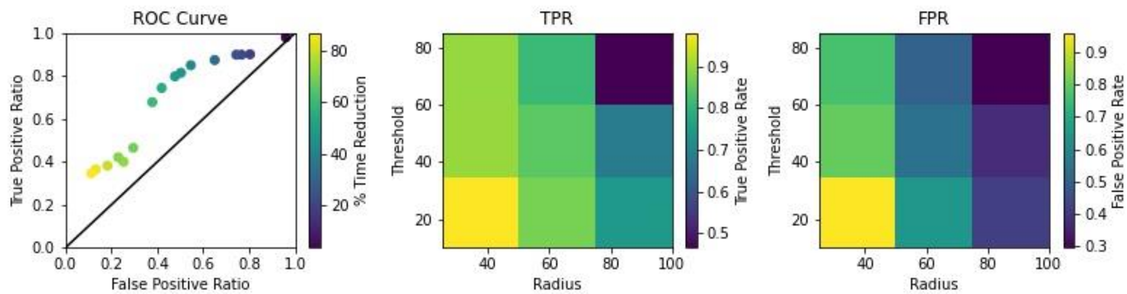

Figure 4. The first panel is the Receiver Operating Characteristic curve for the algorithm at the different thresholds and radii. The black line indicates the 0.5 performance measure, or the line of no discrimination, in which the algorithm would have the inability to distinguish between classes (positives and negatives). The color indicates the percent of the video time reduced. To test the parameter space of the background subtraction algorithm the second and third panels are plots of threshold as a function of radius where color represents the true positive rate and false positive rate for each combination.

One shortcoming of these results is that our algorithm uses the first frame to define the background and does not update over the duration of the video. This proved to be problematic when the environment captured in the viewing window changed over time or when there was an object present in the first frame, thus producing changes in subsequent frames that resulted in false positive detections (e.g. the fish in the first frame of Figure 3B). To improve the background subtraction algorithm, statistical background modeling should be employed. Background modelling allows for the algorithm to become adaptive as the time goes on by utilizing several methods (e.g., moving average, adaptive Gaussian Mixture Model (GMM), color and texture models) (Matzner et al., 2017; Prasad et al., 2018). A study conducted by Matzner et al. (2017), evaluated the performance of three different background subtraction models: Robust Principal Components Analysis (RPCA), Gaussian Mixture Model (GMM), and Video Background Extraction (ViBE). All three of their tested algorithms performed better compared to our simple background subtraction algorithm with results as followed: RPCA 71.64% TPR 42.52% FPR; GMM 80.60% TPR 47.92%; ViBE 87.76% TPR 45.65% FPR (Matzner et al., 2017). Our optimal performance of 74.4% TPR 41.9% FPR only contends with the performance of the RPCA model and when compared to the other models tested by

Matzner et al. (2017), RPCA performed the worst. Also, Prasad et al. (2018) tested 22 background subtraction algorithms ranging in complexity (e.g., mean background, GMM, machine learning-based methods, among many others) and their results indicated that adaptive background modeling enhances algorithm performance compared to simplistic background subtraction methods. While Matzner et al. (2017) and Prasad et al. (2018) show that adaptive background modeling can enhance performance, background subtraction alone still does not meet performance objectives of 90% TPR and 30 % FPR. Additionally, the adaptive background model's performance was susceptible to the same environmental conditions that hindered our simple, non-adaptive background subtraction algorithm's performance (e.g., light rays in the field of view, low contrast between the objects and the background, vegetation the same size as objects of interest, color variations) (Matzner et al., 2017; Prasad et al., 2018). While these results may indicate the background subtraction algorithm's potential to aid in data reduction in the context of fish detection, there is still a significant challenge in meeting performance goals. In addition, background subtraction does not aid in a second important stage of analysis for these types of data: object classification (i.e. fish species identification). Significant manual labor time is thus still required for object classification.

YOLO Methods

To improve performance metrics while also tackling the species identification problem, we began to explore and evaluate a more complex computer vision algorithm, object detection, using an open-source machine learning algorithm, YOLO version 3. Object detection is a common computer vision technique in which images are processed based on identifying and locating features that are representative of an assigned class or classes. Labeling video data is accomplished by dividing the video into frames and drawing a bounding box or contour around that object with an associated label (class) manually. Once the labeled dataset is created, the machine learning model is trained, tested, and evaluated. YOLO accomplishes this in one evaluation by 1) resizing the input image 2) running the single convolutional network consisting of 24 initial layers that extract the features and 2 fully connected layers that predict the output probabilities and confidence 3) thresholds the output by model confidence (Redmon et al., 2016). We then threshold the model confidence with an ignore threshold of 0.6 (i.e., want the model to be 60% confident or better that the detection is a true positive), which is at the discretion of the user. This means that if a detection has a model confidence equal to or above 0.6 it will be classified as a positive. If the model confidence is below 0.6, it will be classified as a negative. YOLO is a model that reasons globally (i.e., it views the entire image and all the objects it contains) which gives it the ability to implicitly encode contextual information (i.e., learn generalizable representations of objects) about class type and appearance directly from the full image (Redmon et al., 2016). YOLO can process in near real-time at 45 frames per second (Redmon et al., 2016). Reducing IFW video analysis time is not only cost-effective, but also, more time can be allocated to conducting phenology studies on species of interest, calculating annual run size, and evaluating interannual variability in fish migration.

Run pre-trained model on IFW dataset

We used Xu and Matzner's (2018) implementation to evaluate the generalization performance of YOLO on the same IFW video that was manually analyzed and used in the background subtraction analysis. This means that Xu and Matzner's (2018) dataset was used to initialize training, and the IFW dataset was excluded from training and used only for evaluation. Specifically, we utilized their pre-trained model that was trained on three marine hydrokinetic and hydropower datasets: Voith Hydro located in Scotland, UK; Wells Dam located in Washington, USA; and Igiugig located in Alaska, USA, which contains salmonid species of Chinook and Sockeye salmon at various life stages (Xu & Matzner, 2018). This dataset contains 54,516 frames with associated annotations and was trained for 20 epochs to achieve the best model weights (Xu & Matzner, 2018). We executed the pre-trained model using a range of combinations between two parameters defined by Xu and Matzner (2018) as the object threshold (i.e., the threshold to distinguish between object and non-object) and the nms threshold (i.e., the threshold that determines if two detections are duplicates) seen in Table 2. When applied to the IFW video, the pre-trained YOLO model results in few true positive and false positive detections. The best performance was 29% TPR and 23% FPR and reduces the video time by 76.6% (Table 2). In ROC space, these points are in the lower left-hand corner (Figure 5) and indicate that the model is 'conservative' (Fawcett, 2006). Therefore, the model will only make positive classifications with strong evidence and will have low false positive detections, but this often results in low true positive detections (Fawcett, 2006).

obj_thresh	nms_thresh	Total Frames	Total frames with fish	Total frames without fish	YOLO detections	True Positives	False Positives	TPR	FPR	% Time Reduction
0.9	0.85	20156	1819	18337	2537	278	2259	15	12	87.4
0.6	**0.75**	**20156**	**1819**	**18337**	**4725**	**535**	**4190**	**29**	**23**	**76.6**
0.5	0.45	20156	1819	18337	449	57	392	3.13	2.14	97.8
0.2	0.75	20156	1819	18337	2413	258	2155	14.2	11.8	88.0
0.2	0.35	20156	1819	18337	45	11	34	0.6	0.2	99.8

Table 2. Pre-trained YOLO results for a range of parameter values. Calculated true positive ratio, false positive ratio, and percent time reduced for object threshold and nms threshold combinations. The row in **bold** is the best parameter combination.

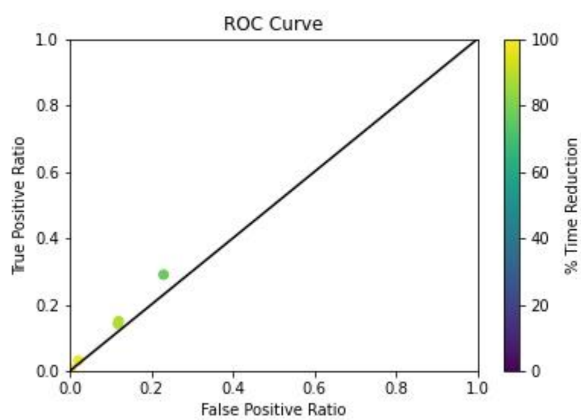

Figure 5. The Receiver Operating Characteristic curve for the pre-trained YOLO results at different object and nms threshold combinations. The black line indicates the 0.5 performance measure, or the line of no discrimination, in which the algorithm would have the inability to distinguish between classes (positives and negatives). The color indicates the percent of the video time reduced.

Our results indicate that this model is not robust enough to detect features that were not contained in the training dataset (i.e., unlearned features will be excluded). For example, our dataset contains different camera specifications, lighting conditions, turbidity, water flow, direction in which fish traverse, and fish species (e.g., American shad, river herring, bass, white sucker, etc.). All these new features need to be incorporated into the training dataset to elicit accurate detection for our specific case study. This falls in line with Xu and Matzner's (2018) generalization test where they reserved the Igiugig dataset from training (i.e., only used Voith Hydro and Wells Dam during training) and tested the model on all three datasets. This experiment resulted in good detections for the two datasets (Voith Hydro and Wells Dam) that were used during training, and very poor detection results for the dataset (Igiugig) excluded during training (Xu & Matzner, 2018). Both their and our results show that it is imperative to train the model on the user's specific case study for the model to correctly detect fish (Xu & Matzner, 2018).

Conclusion

Commercial, recreational, and subsistence fisheries are important components to maintaining a successful blue economy, and evaluating the effects of industrialization on these fisheries (e.g., dams, fishways, and other hydropower structures) are paramount to the conservation of the species that these fisheries encompass. Conventional methods of evaluation (i.e., seining and angling) are costly and time consuming. Underwater video monitoring is often the solution because it can capture nearly all the fish that traverse the area in question, but this method is also labor intensive. Our goal was to reduce video analysis time by automating the fish detection process using two computer vision algorithms: background subtraction and YOLO. Upon executing the background subtraction algorithm on the IFW video, we found that the optimal parameter combination is threshold 75, radius 10 resulting in a 74.4% TPR and 41.9% FPR and reduced the video time by approximately 53.13%. This suggests that background subtraction alone is not sufficient to reach performance goals of 90% TPR and 30% FPR. Furthermore, we evaluated the generalization capabilities of the machine learning algorithm, YOLO, using a pre-trained model on the IFW video. Our results indicate that the model is not robust enough when the dataset in question is not included during training. Additionally, we observed challenges in automating fish detection consistently in a dynamic environment where light, turbidity, and water velocity are changing over time. In future efforts, if the user implements the background subtraction approach, the algorithm needs to be dynamic (i.e., the background model needs to update over time) for optimal results. If the user implements YOLO, then the user needs to develop and train the YOLO model on the specific case study dataset. In both cases, the user needs to define acceptable tradeoffs between the true positive rate and false positive rate.

Ongoing work for this project includes labeling and annotating the IFW video dataset in VOC format for YOLO training and testing. The dataset will include ultimately subclasses for species identification to further reduce video analysis time.

Bibliography

Amandeep, E., Goyal, M. (2015) Review: Moving Object Detection Techniques. *International Journal of Computer Science and Mobile Computing, Vol.4 Issue.9, September- 2015, pg. 345-349.*

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Darrow, J., & Neilan, B. (2013). Shad of the Raritan River. New Jersey Fish & Wildlife Digest, 2013 Freshwater Fishing Issue.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874. doi:10.1016/j.patrec.2005.10.010

Franklin, A., Haro, A., Castro-Santos, T., & Noreika J., (2012): Evaluation of Nature-Like and Technical Fishways for the Passage of Alewives at Two Coastal Streams in New England, Transactions of the American Fisheries Society, 141:3, 624-637

Keen, M. R., Schwarz, A., & Wini-Simeon, L. (2018). Towards defining the Blue Economy: Practical lessons from pacific ocean governance. *Marine Policy, 88*, 333-341. doi:10.1016/j.marpol.2017.03.002

Matzner, S., Hull, R.E., Harker-Klimes, G.E., & Cullinan, V.I. (2017). Studying fish near ocean energy devices using underwater video. *OCEANS 2017 – Anchorage*, 1-7.

NOAA & USFWS (2016). IN-RIVER RESTORATION PLAN/ ENVIRONMENTAL ASSESSMENT (RP/EA) FOR THE AMERICAN CYANAMID CO. SUPERFUND SITE, BRIDGEWATER TOWNSHIP, SOMERSET COUNTY, NEW JERSEY . Retrieved from https://casedocuments.darrp.noaa.gov/northeast/am_cyanamid/pdf/Final_River_Restoration_Plan_Envir omental_Assessment_for_American_Cyanamid_Nov_2016.pdf

Prasad, D. K., Prasath, C. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2019). Object Detection in a Maritime Environment: Performance Evaluation of Background Subtraction Methods. *IEEE Transactions on Intelligent Transportation Systems, 20*(5), 1787-1802. doi:10.1109/tits.2018.2836399

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.91

Salman, A., Maqbool, S., Khan, A. H., Jalal, A., & Shafait, F. (2019). Real-time fish detection in complex backgrounds using probabilistic background modelling. *Ecological Informatics, 51*, 44-51. doi:10.1016/j.ecoinf.2019.02.011

Vastano, A., Weldon, O., Hernandez, M., Melara, S., Lahr, A., Hazlett, S., Jensen, O. (2018). Evaluating Passage Performance of Spawning American Shad and River Herring at the Island Farm Weir on the Raritan River: Results of a Six Year Study. *American Fisheries Society August 23, 2018 – Atlantic City.*

Xu, W., & Matzner, S. (2018). Underwater Fish Detection Using Deep Learning for Water Power Applications. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. doi:10.1109/csci46756.2018.00067