CS153 Shark Tracking Final Report

Francine Wright
Harvey Mudd College
301 Platt Blvd, Claremont, CA 91711
fwright@hmc.edu

code: https://github.com/f-wright/shark-and-human-tracking
analysis of results: https://tinyurl.com/shark-human-data-analysis

1. Introduction

This project aims to automatically label drone video data to track the motion of sharks and humans in the ocean. Prof. Soto from Harvey Mudd College and collaborators from the SharkLab at CSU Long Beach will use this data to better understand how sharks behave in the presence of humans in the water, specifically surfers and bodyboarders.

The video sequences used in this project are from white shark surveys along the California coast. They are of varying quality.

This project will allow researchers to take advantage of drone footage, which is a useful way of gathering animal related data without disturbing them. However, even though this technology has been deployed for shark monitoring, the video footage is not being fully utilized since there is so much of it. This project will also advance the field of computer vision, since it deals with video footage over water, which isn't commonly done. It comes with unique challenges, such as occlusions from waves and sunlight reflections on the water (Example data shown in Figure 1).

Once completed, this project will be able to take an input drone video and initial bounding boxes around objects to track, and output bounding boxes with labels which track each of the sharks and humans in the video over time and space.

Figure 1. Screenshots from 4 different videos in the California coast dataset provided.

2. Detection-Based Multiple Object Tracking Exploration and Challenges

Initially, this project aimed to automatically label drone video data without bounding box input. Videos were given to students in the CS153 course to label using VIA, the VGG Image Annotator [4]. The plan was to convert those annotations into MS COCO format JSON files [6], in order to use them as ground truth data. That ground truth data could have enabled more complex machine learning methods, which have a heavy reliance on a large amount of data.

To perform detection-based multiple object tracking, an object detection model would have been run in conjunction with an object tracking model [8]. In detection-based tracking, object detection is applied in each frame, then these detections are linked to track the object's motion. There is no human input of bounding boxes outside of training.

The initial plan was to use detection-based multiple object tracking (MOT) to compare a couple of pre-made object detection and object tracking models retrained using transfer learning on the California coast dataset given. Three different object tracking models (Occlusion Geodesics [11], Minimum Clique Graphs [14], and Detection- and Trajectory-Level Exclusion [9]) and two different object detection models (Faster R-CNN [13] and YOLOv3 [12]) would have been tested and compared using mean Average Precision (mAP) and Intersection over Union (IoU), since those methods are commonly used to evaluate object tracking models [8]. Due to challenges in accessing the code of the object tracking models, using SORT [2] was also attempted later in the project.

However, the annotations could not be converted into MS COCO format, so the scope of the project was reduced to go around this problem. Instead of performing detection-

based MOT, the final method uses detection-free MOT, so initial bounding boxes are given as input to the code along with the video sequences [8]. The model follows the initial detections in subsequent frames without using ground truth data beyond the input box.

There are some downsides to the new method. Detection-free tracking puts a bit more burden on the user, since they have to input the bounding box when running on a video. Not being able to use ground truth data also prevents usage of machine learning models, which often perform better than more traditional computer vision techniques.

However, there are also benefits to using detection-free tracking. A detection-based method would have to be trained on ground truth data, and the California coast dataset is rather limited. The dataset contains 16 videos labeled as good quality and 6 videos labeled as medium quality, most of which are short. This is not that much data, so a model trained on it may not be very robust. Additionally, while there is variation in the conditions between videos, there still variation not captured in the sample. Most of the videos were taken on clear and sunny days, and there are limited non shark or human objects in the water. With a machine learning model, it would be possible to pass in a video different from what the model expects, resulting in false detections or failed detections. Detection-free tracking could result in a more robust model.

There were also challenges in this project originating from the data itself. These include sunlight reflections off of the water, sharks being occluded by waves, reflections, or other things, humans being ambiguously located on the beach or in the water, and unexpected motion from the drone adjusting the camera. Ideally, the object tracking method would be able to perform well despite this added difficulty.

3. Detection-Free Multiple Object Tracking Method

Due to the aforementioned annotation challenges, detection free tracking was used in the final iteration of this project. A couple of different detection free models were considered.

First, MDNet [10] was attempted, but was discarded due to a lack of a pretrained model combined with the annotation issues.

Then, OpenCV Multiple Object Tracking (MOT) was considered, which was attainable with the lack of ground truth data. OpenCV MOT provides access to seven different models [3], which allowed for easy comparison of different methods. Since it provided an easy interface to compare many of the best models that don't require training data, OpenCV MOT was selected for this project.

Three different OpenCV MOT models were tested. Dardagan et al. evaluated all the different OpenCV MOT models on IoU and Center Distance (CD) [3]. They found that CSRT was the best for accuracy and precision, but did not perform as quickly as other models. They found that MIL and Boosting did not perform as well as CSRT, but were the closest options while also performing somewhat faster. Thus, these three models were applied and evaluated on the dataset.

These models were all published independently prior to their OpenCV implementations. CSRT, the Channel and Spatial Reliability Tracker, is the newest of the three. It was originially published in 2017 by Lukezic et al. [7] and achieved impressive results on VOT 2016, VOT 2015 and OTB100. Most authors agree that CSRT is the best model available through OpenCV MOT, so it seems likely that this performs the best on the shark and human data as well. MIL, Multiple Instance Learning, was proposed in 2009 by Babenko et al. [1]. It was designed to overcome the drift that occurred in many supervised learning models at the time. This could provide improvement over machine learning models, but might not work as well in comparison to other traditinoal methods. Boosting was the oldest method included in OpenCV MOT models, published in 2006 by Grabner et al. [5]. It was designed to handle appearance changes of objects such as rotation or illumination changes and select the most discriminating features between the object and the background to track. This could prove useful given the frequent rotation of sharks and humans and the similar color between the sharks and the background.

Each of these three models were applied to each of 16 of the 22 videos provided for this project. Due to a lack of accessible ground truth data, the options for analysis were limited. For each video and tracker, the amount of time the shark or human was partially contained in the bounding box was tracked. Additionally, the amount of time each object spent in frame and the amount of time each object spent in frame without a high degree of occlusion were collected. From that data, multiple different comparisons were done to evaluate which tracker performed the best. Additionally, qualitative analysis was performed to supplement the minimal quantitative data available.

4. Results and Analysis

The first quantitative comparison done was quite simple. For each shark or human in each video, the tracker that kept track of the object for the longest was recorded. If two trackers performed equally well, they were both recorded. If all three trackers performed equally well, the object was excluded from this test. The results of this comparison are shown in Figure 2.

By this metric, CSRT was the best tracker, followed by Boosting, since they had the best performance on the most

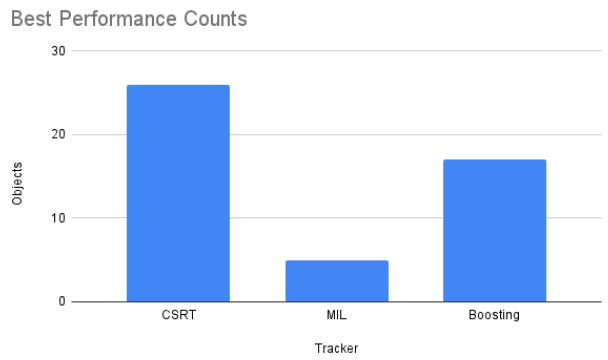

Figure 2. Number of objects each tracker performed the best on, excluding those where all three trackers performed the same.

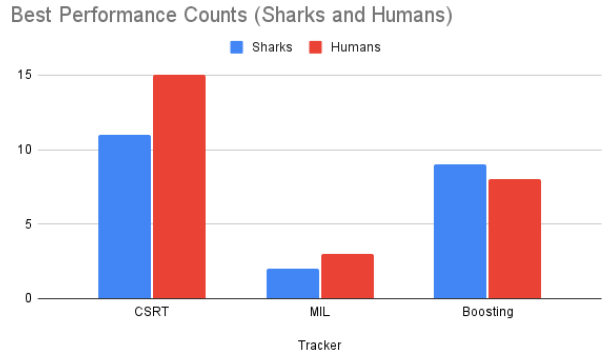

Figure 3. Number of sharks and humans each tracker performed the best on, excluding those where all three trackers performed the same.

objects. MIL lagged far behind the two of them, performing the best on only 5 objects where the three trackers had different performance.

It is also worth comparing the performance of the trackers for sharks and humans separately, since they are rather different tracking problems. The humans had much more color contrast with their backgrounds. They also were less prone to changing shape, since they were often on fixed size surfboards and above water. To see if the different trackers performed differently on sharks and humans, the same data was separated into sharks and humans, shown in Figure 3.

This figure shows similar results. On both sharks and humans, CSRT performed the best, followed by Boosting, followed by MIL. Both CSRT and MIL perfomed better on humans than on sharks by this metric. Boosting, however, performed better on sharks than on humans. This does not mean that Boosting was better at tracking sharks than it was at tracking humans, but it does show that it had a relative advantage over CSRT and MIL in shark tracking.

Additionally, it is worth considering how well the trackers performed generally, not just which were comparatively better. To evaluate this, the average percent of time a bounding box stayed on an object was calculated. The percentage of time the box remained on the object was calculated (using the amount of time the object was in frame, not the length of the video). This is shown in Figure 4.

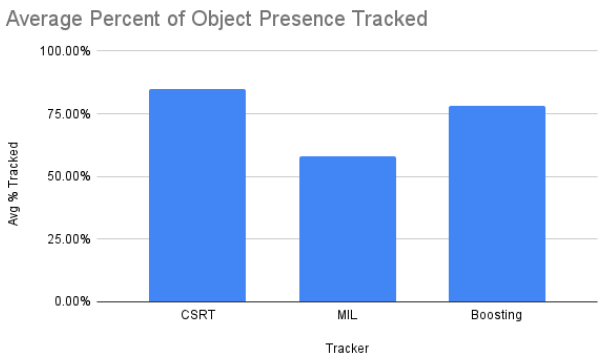

Figure 4. Average percent of time an object in frame was successfully tracked.

By this metric, CSRT is again the best tracker, followed by Boosting, then MIL. CSRT tracked each object for 85 percent of the time it was in the frame on average. Considering the difficulty of some of these videos, with objects frequently being partially or completely occluded and sharks having very similar coloration to their backgrounds, this performance is relatively good. Boosting tracked each object 78 percent of the time on average, and MIL lagged behind at 58 percent.

Interestingly, CSRT and Boosting show similar performance under this metric, even though their outputs appear quite different. CSRT is also a much newer algorithm than Boosting. CSRT was published in 2017, while Boosting was published in 2006.

Similarly to the best tracker counts metric, it is worth considering how this statistic varies across the tracking of sharks and humans. This is shown in Figure 5.

Again, CSRT shows the best performance on both sharks and humans, followed by Boosting, then MIL. All three algorithms perform better on humans than on sharks. This was expected, since the sharks have less color contrast with their background and change in shape more. MIL performed notably poorly on sharks, tracking them for only 24 percent of the time they were in frame on average. The highest percentage time tracked (CSRT's) was 89 percent for humans and 76 percent for sharks.

Qualitatively, it is worth noting that MIL and Boosting do not resize their bounding boxes to account for rotations and other shape changes of objects. This does not appear in the quantitative analysis since the only value calculated was whether or not the bounding box stayed on the object,

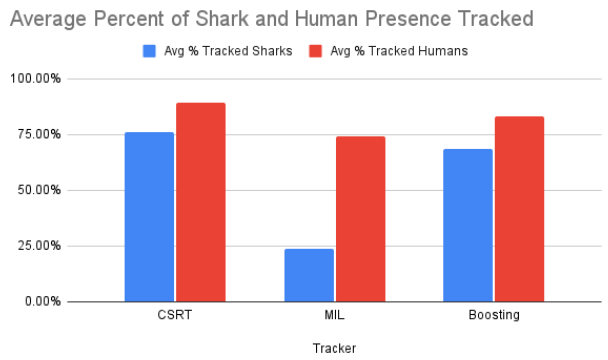

Figure 5. Average percent of time an object in frame was successfully tracked, divided up by sharks and humans.

but it would certainly appear in a metric like IoU which accounts for the overlap of the predicted bounding box and the ground truth value.

CSRT, in contrast, does adjust the bounding box for the size of the object, but often tends to overestimate the object's size. It includes a lot of the ocean instead of staying close to the outline it was supposed to track. This would also be penalized in a metric like IoU. However, this could create a bias towards CSRT in the quantitative results in this project. Since the only metric tracked was how long the bounding box contained some portion of the object, too large bounding boxes were not penalized and had a higher probability of containing some part of the object, even if it wasn't being effectively tracked. While in many cases CSRT did maintain small boxes localized to the desired object, there definitely were also cases where the bounding box extended far beyond the desired object. Accurate ground truth data would be necessary to effectively penalize CSRT boxes getting to be "too big" without being too arbitrary, so that is a shortcoming of this project's analysis. However, generally, CSRT's bounding boxes still fit the objects better than Boosting's or MIL's, since they regularly changed shape to accommodate rotations and zooming of the video.

Another thing not covered in the quantitative analysis is the consistency of the bounding boxes predicted by different models. CSRT tends to provide relatively smooth bounding boxes, so the transition between one frame to the next seems less sudden. MIL and Boosting both output more jittery bounding boxes. Boosting seems to have the most shaking of the three.

All three tracking methods suffered from occasional video skipping of their output videos. This could not be resolved within the time constraints of this project, since other users of OpenCV's multiple object trackers did not seem to encounter the same issue. The output videos remained the same length as the input videos, and were consistent across trackers.

5. Conclusion and Next Steps

From all of the tests performed, in addition to qualitative analysis, CSRT performed the best of the three trackers. This is consistent with the expectation gathered from the literature on these tracking methods [3]. Boosting followed in performance by all metrics, then MIL.

Boosting was a surprisingly close second, since it was published 11 years before CSRT and 3 years before MIL. It performed far better on the sharks then MIL (45 percent). This is likely because Boosting was designed to handle appearance changes in objects, like the rotations and lighting changes that the sharks underwent as they moved through the water.

The final iteration of this project required far more user involvement than was initially intended. It is tedious to label the sharks and humans in many videos, especially when there are many and some are difficult to see. However, the interface is simple to use and allowing user involvement means that the tracker works on a broader range of videos. The resulting code from this project could theoretically be used to track any objects, not necessarily just sharks and humans.

Additionally, there are many cases that the tracker does not function in. Since all objects must be labeled in the first frame, objects that enter the video frame later cannot be tracked, nor can objects that temporarily leave and re-enter. While the results are promising for the sharks and humans tracked, there could be many left out due to this. This could be circumvented by cutting the video into snippets and passing them in individually. This would allow objects that enter later in the video to be tracked, as well as objects that exit to be tracked again when they re-enter. However, this puts more burden on the user in terms of labeling objects, which is not desirable.

For a future iteration of this project, it would be useful to have accurate ground truth data to do more robust analysis and switch to a detection-based MOT model. A detection-based model could be compared against CSRT to see if there is an improvement in performance, but it would undoubtedly reduce the work on the user. More analysis, such as IoU and CD would be useful to determine the accuracy and precision of the OpenCV models tested. Since CSRT bounding boxes occasionally became very large and Boosting and MIL bounding boxes never change size, those metrics would help in determining the practicality of these algorithms for real-world applications.

However, this project has shown that CSRT provides a good option for tracking sharks and humans. Despite its shortcomings, it could potentially be used in a practical application, like Prof. Soto's research.

References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pages 983–990. IEEE, 2009. 2

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 1

[3] Nadja Dardagan, Adnan Brdjanin, Dzemil Dzigal, and Amila Akagic. Multiple object trackers in opencv: A benchmark. *CoRR*, abs/2110.05102, 2021. 2, 4

[4] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA, 2019. ACM. 1

[5] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Bmvc*, volume 1, page 6. Citeseer, 2006. 2

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 1

[7] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017. 2

[8] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021. 1, 2

[9] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3682–3689, 2013. 1

[10] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[11] Horst Possegger, Thomas Mauthner, Peter M Roth, and Horst Bischof. Occlusion geodesics for online multi-object tracking. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1306–1313, 2014. 1

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[14] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European conference on computer vision*, pages 343–356. Springer, 2012. 1