MBARI Monterey Bay Aquarium Research Institute

Workflows for Automated Detection and Classification of Unlabeled Deep Sea Imagery

Nathaniel Yee[1,2], **Danelle Cline**[1*], **and Duane Edgington**[1*]

[1]Monterey Bay Aquarium Research Institute, Moss Landing, 95039, USA

[2]Franklin W. Olin College of Engineering, Needham, 02492, USA

ABSTRACT

Over the past 28 years, professional video annotators at the Monterey Bay Aquarium Research Institute (MBARI) have recorded over 5.5 million observations throughout a collection of over 23,000 hours of video footage. MBARI researchers and scientists query these observations through the Video Annotation and Reference System (VARS) to conduct oceanographic research. However, recording these observations requires a lot of time, energy, and knowledge from MBARI's professional video annotators. In addition, due to the ever increasing rate of incoming imagery, an efficient automated detection and classification system would be of great assistance to the upkeep of the VARS database. Because MBARI's 5.5 million observations are currently unable to be used to train deep learning object class detectors, we explore various workflows to create these systems from unlabeled data. We find that combining deep learning algorithms and various annotation methods in a bootstrapping approach can produce automated detection and classification systems capable of accurately detecting and classifying key species with minimal training data.

Introduction

The Video Annotation and Reference System (VARS) connects scientists at the Monterey Bay Aquarium Research Institute (MBARI) to vast amounts of quantitative and qualitative oceanographic data. VARS has been proven itself to be an invaluable tool for oceanographic research with over 375 publications citing the database. To collect data for VARS, MBARI deploys remotely operated vehicles (ROVs) in the Monterey Bay around 300 times per year. Over the past 28 years, MBARI has accumulated over 23,000 hours of

high quality video footage. Professional video annotators at MBARI have manually processed this footage - recording over 5.5 million observations into VARS. This annotation process is both time intensive and requires a significant amount of knowledge to identify the thousands of species of animals.

In addition to ROVs, MBARI uses cameras on autonomous underwater vehicles (AUVs), and on stationary underwater observatories such as Station M. This is largely in part to the minimal environmental impact and high quality data that cameras provide. As cameras continue to decrease in price and increase in performance, MBARI is expected to increase their usage of cameras, and subsequently increase the amount of image data which must be processed.

Due the difficult and time intensive process of manually annotating imagery and an increasing usage of cameras, the need for technology capable of either assisting MBARI video annotators record observations or automatically creating observations is of great importance. In recent years, deep learning has become the state of the art method for object detection and classification. *He et al.* was able to win 1st place in several tracks of the ILSVRC and COCO 2015 competitions using a combination of residual networks and Faster R-CNN[12]. However, these methods require a large amount of annotated data. For example, the imagenet ILSVRC challenge uses 150,000 images, annotated across 1000 object categories[3]. MBARI's 5.5 million observations are unable to be used as the annotations for the current deep learning algorithms. As a result, I will explore workflows that allow humans and computers to work together to increase the speed at which annotated datasets can be built, and explore the effectiveness of deep learning systems trained on small datasets for automated detection and classification of deep sea imagery.

Workflow 1 - Saliency-based neuromorphic selective attention detectors and convolutional neural network classifiers

The first workflow we explored leveraged past work by my mentors Duane Edgington, Danelle Cline, and last year's intern Dallas Hollis. They developed a two part system: detection with a saliency-based neuromorphic selective attention algorithm, and classification with a fine tuned convolutional neural network, Inception V3[45]. We theorized that using the system on 5-7 hours of video would produce a large and clean enough dataset to start experimenting with deep learning algorithms. However, the system was ultimately unable to produce a clean enough dataset as the predictions contained too many false

positive detections or misclassified objects, but we did gain insight into what factors are important in future workflows. To improve the saliency detector, a user iteratively hand tunes saliency parameters to detect animals that were not previously detected, while keeping background detections to a minimum. This workflow is problematic because the user is required to have a certain level of knowledge for how saliency-based detectors work. Neither I, nor our primary users, MBARI video annotators, would be able to tune the detection system to detect every desired object without including a significant amount of unwanted background data. To improve the CNN classifier, a user manually looks through directories of classifier predictions and moves any misclassified predictions into the proper directory. This workflow ended up being very effective because a single user can correct and verify thousands of images per hour.

Workflow 2 - Faster R-CNN and LabelImg

Workflow

The second workflow I explored involves Faster R-CNN and the annotation tool LabelImg[6]. LabelImg is a graphical annotation tool designed for use in deep learning algorithms. Faster R-CNN is a state of the art deep learning architecture for object detection and classification. The workflow begins by building an initial dataset of images and annotations using LabelImg. Then, the sequence of steps is repeated until the dataset reaches a desired size, or the system reaches a desired performance:

1. Train Faster R-CNN on the dataset.
2. Use Faster R-CNN to predict annotations for new images.
3. Use LabelImg to manually correct the predicted annotations.
4. Add the new images and corrected annotations into the dataset.

This workflow is effective because it compounds on itself in a bootstrapping manner. Correcting Faster R-CNN's predicted annotations allows for a faster annotation process. Increasing the size of the dataset trains a better performing Faster R-CNN. This workflow is also effective because it does not require a deep understanding of the underlying algorithms. Faster R-CNN's hyperparameters must only be configured once and generally perform well within a large range of values. The user must only be capable of correcting

misaligned bounding boxes, identifying bounding boxes that don't contain and object, and identifying misclassified species of animals.

Faster R-CNN

The first time Faster R-CNN is trained on the dataset, it is initialized with Imagenet weights to take advantage of transfer learning. The second time and every time henceforth, the user can choose to train a fresh model initializing with Imagenet weights, or continue training on top of the previous model. Creating a fresh model guarantees that each image is used an equal number of times throughout the training process. Continuing to train on top of a previous model requires less training time to achieve maximum performance but will have used each image an unequal number of times. While in the annotation correction process, I would recommend continuing to train on top of the previous model for fastest training time. I would recommend initializing with Imagenet weights when you want to evaluate the performance of the current dataset.

Annotation Correction Process

Annotation corrections are done in four steps.

1. Delete any bounding boxes that don't contain an object.
2. Correct the classes for all predicted bounding boxes.
3. Correct the position of all predicted bounding boxes.
4. Create bounding boxes and corresponding classes for all objects that are missing a bounding box.

Note that when creating or correcting bounding boxes, if the bounding box contains an animal, it should contain the animal's body and shorter limbs but not any long or relatively skinny extremities (such as the tentacles on the Scotoplanes Globosa). This helps the neural network learn to identify the objects and not the neighboring background area.

Results and Conclusions

By utilizing the Faster R-CNN and LabelImg workflow, I created a dataset of annotated benthic imagery and tested Faster R-CNN's performance for this size of dataset.

Dataset

Each image in the dataset begins as a frame grab every five seconds from transect benthic dive footage. During the annotation process, the user decides if the image should be included in the dataset. Annotations have been recorded across 17 categories among the hundreds potential object categories that appear throughout the images. In any selected image, every occurrence of the 17 categories is annotated so Faster R-CNN's object detector will not get penalized for correctly detecting an object. Every image is also de-interlaced by taking every other row and every other column of pixels such that a 1920x1080 image is reduced to 960x540. The resulting dataset is split between a training set and a testing set. The training set is built from seven, one hour dives and consists of:

- 1,659 images
- 17 species of animals
- 3,634 bounding boxes

The test set is built from two, one hour dives, that were recorded sequentially after dives in the training set. Sequential dives were chosen so the image backgrounds and detritus would look similar to that of training data, but the individual animals would be unique. Performance metrics from this test set are useful because annotating sequential dives is a valid use case for MBARI. The test set consists of:

- 618 images
- 16 species of animals
- 1,077 bounding boxes

Faster R-CNN

Adjustments to the original Faster R-CNN paper are made to better suit the benthic transect imagery. Anchor sizes have been reduced to 64, 128, and 256. Images are kept at original 960x540 pixels. An image mean of 102.83, 126.35, 104.67 (RBG) is subtracted to normalize images. Resnet50 initialized with Imagenet weights is chosen as the base model. A version of Faster R-CNN called, Keras frcnn, is used

for its very straightforward instructions and customizable codebase[7]. After training Faster R-CNN for approximately 1,500 epochs, average precisions per class are evaluated from the test set.

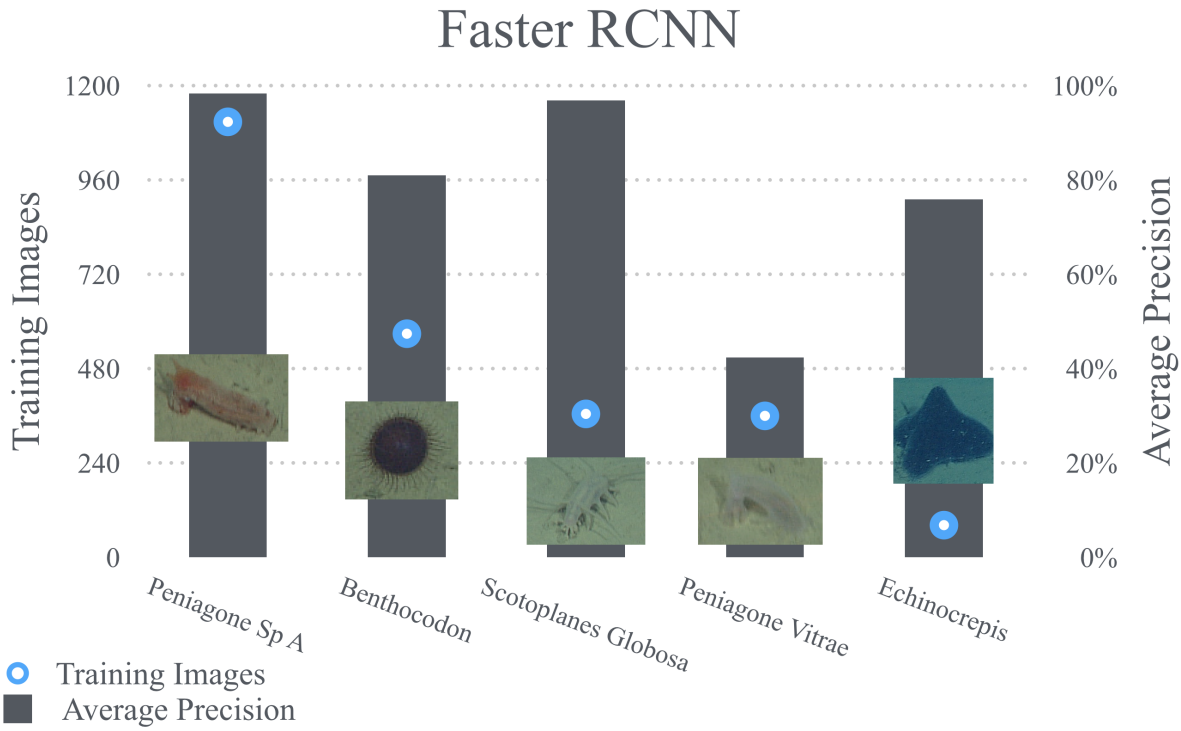

Figure 1. Faster R-CNN evaluated on the test set. 5 of 17 animals were included for statistical significance

- The Peniagone Sp A showcase the ideal and upper limit of Faster R-CNN's performance on the benthic transect imagery. The Peniagone Sp A is one of the most abundant animals with 1,100 occurrences in the training set. The Peniagone Sp A also has very unique and strong visual features. As a result, Faster R-CNN is able to achieve a 98.3% average precision for the Peniagone Sp A.

- The Benthocodon help us understand the flexibility of Faster R-CNN's hyperparameters. In this case, we have set the smallest anchor size to be 64x64 pixels while the smallest Benthocodon are around 16x16 pixels, or 6.25% of the original anchor size's area. However, Faster R-CNN is still able to achieve an 81.0% average precision for the Benthocodon. Ultimately, this means that using deep learning architectures like Faster R-CNN give us the flexibility to focus on creating better workflows rather than tuning the algorithm's hyperparameters.

- Like the Peniagone Sp A, the Scotoplanes Globosa are relatively abundant and have strong visual features. However, in the beginning stages of workflow 2, they were constantly being confused with the background. However, after updating the annotations such that only the body and legs of the Scotoplanes Globosa were included in the bounding box, Faster R-CNN no longer confuses background for the Scotoplanes Globosa and is able to achieve an average precision of 96.8%.

- Faster R-CNN has a 42.3% average precision for Peniagone Vitrae. Even though the Peniagone Vitrae has a similar number of occurrences as the Scotoplanes Globosa in the training set, it is a much more difficult animal to detect and classify. It is a small, translucent, gelatinous animal that lives in a habitat that is filled with many other small, translucent, gelatinous objects.

- The Echinocrepis represents one of the key animals that MBARI wants to detect, but is relatively rare. Often times, MBARI researchers want to collect images of just a single animal, so understanding the minimum required training data is important. In this case, Faster R-CNN is able to achieve a 75.9% average precision for the Echinocrepis with just 82 occurrences throughout the training set. This result shows that Faster R-CNN could be used as a method to suggest images that might contain rare animals with little training data.

Suggestions for future work

Deep learning and annotation corrections

Workflow 2 has shown promising results. However, there exist many tests or changes that could make the workflow more efficient.

1. **Test different detection and classification networks for training speed, frame rate, precision, and recall -** Because we now have a dataset compatible with the common implementations of deep learning models, it makes sense to test which models perform best for MBARI. Moving in this direction would allow MBARI to determine which networks have the lowest training time, highest image processing speeds, and best performance for the Benthic Transect Imagery. Some alternative models that are worth trying are, YOLOv2, Single Shot Multibox Detector, and R-FCN. It is also

worth reimplementing Faster R-CNN with Google's new built in object class recognizer TensorFlow models that were released during my internship for the same reasons.

2. **Use the classification verification step of workflow 1 in workflow 2 -** One of the most time efficient processes in workflow 1 is the use of a file explorer to correct and verify image predictions in the classification step. In comparison, using LabelImg in workflow 2 requires the user click each and every object in an image and make sure that it corresponds to the correct class. It is much faster and more accurate to organize cropped thumbnail images into various folders to verify or correct the predicted classes of each predicted object. This method would also remove any false positive detections.

3. **Create animal specific annotation guidelines -** If MBARI chooses to continue using bounding box methods, it makes sense to document guidelines for how to annotate certain animals. For example, the Scotoplanes was only properly detected and classified after drawing the bounding box just around the animals main body and shorter legs. It can also be confusing when to annotate two animals that overlap each other, or just throw away the image. Consistency is key to determining which annotation styles works best.

4. **Integrate workflow 2 into VARS -** Right now, workflow 2 exists completely independent of VARS. Integrating the two could allow querying of predicted annotations and collaborative annotations across MBARI. However, it is probably worth continuing to explore other workflows before committing to integrating a particular workflow into VARS.

Switch to a different type of annotation

1. **Annotate presence or absence of object instances -** Rather than annotating bounding boxes and corresponding class names, *Gokberk et al* showed that object detection could be done with only binary labels for the presence or absence of object instances in each image[8]. Although the results are less accurate than traditional bounding box methods, current annotations at MBARI are done similar to this annotation style so integration into VARS might be much easier.

2. **Click annotations -** *Papadopoulos et al* showed high quality object detectors could be built with

click based annotations[9]. Because clicking the center of an object is a much faster and easier task than drawing bounding boxes, crowd sourcing click annotations becomes a fast, accurate, and cheap option.

3. **Bounding box verification -** *Papadopoulos et al* showed that object class detectors could be built with just human verification of proposed bounding boxes[10]. This is similar to workflow 2, but it requires that the user answers questions about the proposed bounding boxes rather than correct the bounding boxes.

Acknowledgements

I would like to thank my mentors Duane Edgington and Danelle Cline for their support, guidance, encouragement, and freedom and to pursue what I thought was important throughout the project. I would also like to thank George Matsumoto and Linda Kuhnz for their hard work creating this amazing intern experience. And an additional thanks to Linda for the help in MBARI's Video Lab. Finally, I want to thank all the MBARI interns for their excitement and company throughout the summer.

References

1. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv e-prints* (2015). 1506.01497.

2. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *ArXiv e-prints* (2015). 1512.03385.

3. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis. (IJCV)* **115**, 211–252 (2015). DOI 10.1007/s11263-015-0816-y.

4. Walther, D., Edgington, D. R. & Koch, C. Detection and tracking of objects in underwater video. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, I–I (IEEE, 2004).

5. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *ArXiv e-prints* (2015). 1512.00567.

6. tzutalin. Labelimg. https://github.com/tzutalin/labelImg (2017).

7. yhenon. keras-frcnn. https://github.com/NathanYee/keras-frcnn/tree/nathan-dev (2017).

8. Gokberk Cinbis, R., Verbeek, J. & Schmid, C. Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning. *ArXiv e-prints* (2015). 1503.00949.

9. Papadopoulos, D. P., Uijlings, J. R. R., Keller, F. & Ferrari, V. Training object class detectors with click supervision. *ArXiv e-prints* (2017). 1704.06189.

10. Papadopoulos, D. P., Uijlings, J. R. R., Keller, F. & Ferrari, V. We don't need no bounding-boxes: Training object class detectors using only human verification. *ArXiv e-prints* (2016). 1602.08405.

Appendix

Table 1. Number of occurrences in dataset and average precisions for each class.

Class	Train	Test	Average Precision
Peniagone Sp A	1108	453	98.3%
Benthocodon	569	286	81.0%
Elpidia	493	13	15.2%
Scotoplanes Globosa	365	63	96.8%
Peniagone Vitrae	360	93	42.3%
Tjalfiella	230	66	52.6%
Cystechinus Loveni	174	40	71.7%
Peniagone Sp 1	111	2	14.0%
Echinocrepis	82	26	75.9%
Fungiacyathus Marenzelleri	64	13	51.9%
Synallactidae	30	5	52.1%
Peniagone Papillata	14	10	100.0%
Peniagone Sp 2	12	1	10.0%
Fish	8	0	0.0%
Long White	8	3	13.5%
Oneirophanta Mutabilis Complex	6	3	25.9%