Determining Requirements and Analyzing Costs for Automated Classification of Deep Sea Species.

Connor Hale, **California State University Monterey Bay**
Mentors: Danelle Cline and Duane Edgington

ABSTRACT

The Monterey Bay Aquarium Research Institute (MBARI) has a large arsenal of data collection tools, utilized by various underwater video and image collection methods. The resulting data collection consists of over 24,000 hours of video footage, much of which requires further annotation. However, the human cost of manual annotation to classify species in still-frames and video is high, and the rate of data collection continues to increase. As such, MBARI has tested several methods of automated detection and classification throughout the years. However, there are many unanswered questions in this field, such as what the ideal machine learning algorithm to use is, how much data is required initially, and what is the cost to produce a model capable of inference.

Our work focused on the results of building a machine learning model through transfer learning and a preexisting deep convolutional neural network architecture(CNN) [1]. We found that to classify species from a specific collection of five benthic species, between 200-300 training instances per species was sufficient for our machine learning model's accuracy bottleneck to no longer be training data. Furthermore, we collected information on the human power required to curate said training instances, as well as the finances required to use Google cloud resources to train and test the models we worked with.

INTRODUCTION

At MBARI, past work has been done by researchers [2] and interns [3] to test out various detection and classification workflows. As such, there is already some knowledge about the implementation of various workflows, including using local and cloud resources to train and test models. However, the identification of certain species which are of scientific interest at MBARI, have performed with varying accuracy in the past.

One such species is Echinocrepis rostrata, a deep-sea urchin, which has a very distinctive triangular shape. From a human perspective, we would expect such a distinct shape to be easily classified by a machine learning model, however this has not been the case. One of our primary hypotheses for this phenomenon is that sufficient training images were lacking to properly train a classifier to detect Echinocrepis rostrate. As such, our primary goal built on work of past interns Dallas Hollis and Nathan Ye, to increase the number of images for certain scarcely annotated species and determine a point where a machine learning model will not improve its validation accuracy with the addition of more training instances.

To accomplish this task, the project workflow began with data collection and curation from the physical tapes stored at MBARI. The resulting video clips were then stripped down to single frames and fed into a morphology driven object detection machine learning model. This step allowed individual occurrences of species to isolated from frames where multiple species occurred. The results were separated and then used as training sets for an existing deep CNN architecture, inceptionV3 [4]. Two additional layers were added to the end of the CNN, to allow it to function on a custom training set, and then learning curves were calculated.

A learning curve gives information about how the addition of more training data affects the overall performance of a machine learning model [5]. By gradually increasing the size of our training set, we can see these changes in performance. However, this is a computationally heavy job, as it requires training the same model for many epochs, and thus requires strong computing resources. Since GPU resources at MBARI were being utilized, we trained our models using the Google Compute Platform (GCP), and utilized free data visualization tools from a startup, Weights and Biases, to visualize our results.

WORKFLOW – DATA COLLECTION

The first step in the project workflow was augmenting our existing data collection for the five scarce species which scientists at MBARI were interested in. These species were Oneirophanta mutabilis complex, Cystechinus loveni, Echinocrepis rostrate, Peniagone Sp. 1, and Fungiacyathus (Bathyactis) marenzelleri. At MBARI, custom software has been created to search and annotate video and images, called the Video Annotation and Reference System (VARS). VARS was used to query the video collection to find annotations which indicated a tape ID and timestamp where a species of interest could be seen. These tapes were then located

in the MBARI Video Lab vault, and with the help of the video lab researcher technicians, four second clips of the tape were digitized, with the timestamp for the species of interest being in the center of the video clip.

The first three weeks of work on the project was dedicated to this task, and between two interns, 1100 four second clips were generated, giving on average 220 clips per species. These clips were then converted into still frames, where every third frame was saved, to avoid identical frames when the camera capturing the source video was moving slowly. Each clip was shot at 30 frames per second, and the results after the frame extraction yielded 39,000 frames. Next, these frames were deinterlaced using a simple NumPy array slicing function. This resulted in frames which reduced the resolution of the frame by a factor of four.

WORKFLOW – OBJECT DETECTION

A pre-existing workflow for morphology-based object detection was implemented by one of the project mentors, Danelle Cline, to reduce human labor needed and test how well object detection works with when separating by morphotype. The results of this step yielded a series of folders which had cropped images stored in them, where the folder name indicated what the general shape of the cropped images within looked like. This classification worked quite well for some species, such as the distinct Echinocrepis rostrata, however for other species whose shape and position in images was more varied, it was not as accurate at sorting into the appropriate morphotype folder. However, it did effectively crop out 29,000 regions of interest, which were then sorted through, and approximately 500 images from the five scarce species were curated from the collection. Care was taken to avoid multiple frames from the same video, unless later frames depicted the species in a new unique pose. This uniqueness was determined based on subjective judgement by the researcher sorting the images.

WORKFLOW – COMPUTE ENVIRONMENT

After data acquisition, an environment had to be set up to train the inceptionV3 model with the collected data. After consideration, our team decided to test the Google Compute Platform (GCP). GCP offered several features which we initially assumed would benefit our project, including online data storage through buckets, as well as processing units which are specialized for use with TensorFlow, an open source library which we used to build the models

for project. However, the main benefit of using GCP was the scalability of our graphical processing unit options. Machine learning is a complex series of simple arithmetic operations, and as such, using a GPU, which has densely packed arithmetic logic units, enables training to be done much faster than with a central processing unit.

The first step in setting up our environment was to create a virtual machine on a cloud server and allocate it sufficient resources. This allocation was determined through trial and error, running training and using Weights and Biases to check the average GPU usage during runs. Since the GPU is the most expensive resource on the virtual machine, determining the most cost-efficient GPU to use was an important step in understanding what kind of resources would be needed to increase the scale of the project in the future. Another benefit of using virtual resources comes from the ease of creating new virtual machines when a team member wants to test something new, without risking changes in an environment other team member are working in. The details to set up the environment, upload data to the cloud, and run training can be found in the README of the GitHub repository https://github.com/AtlasHale/ml_classify.

RESULTS

The results for the project can be separated into two sections: the training images required results, and the human power and monetary cost to obtain these images and train a model. At the start of the project, there existed on average 100 cropped and sorted images per species of the primary five scarce species. Both interns assigned to the task had no knowledge of deep sea animals, and as such were not experts at classifying organisms by species based on visuals. As such, the time required to collect additional training instances was likely higher compared to the time a professional from the MBARI video lab would need to curate the same set of data. However, the greatest bottleneck we faced in data collection was digitizing.

All tape at MBARI prior to 2018 was collected on physical magnetic tape cassettes. As such, the process to collect the training data went as follows:

- VARS query to find tape of interest
- Search for tape in video lab vault
- Using Blackmagic Media Express, enter timestamp of species
- Digitize the clip

From this four step process, by far the greatest amount of time was spent entering time stamps and digitizing the clip. The nature of digitizing from tape means that it must be a 1:1 time cost, meaning to digitize 1100 four second clips, the actual digitization 4400 seconds were needed to digitize all the clips. However, when digitizing tapes there is tremendous overhead in the form of fast forwarding to the position the clip starts and rewinding the tape after finished. The tapes are one hour in length, and if a species sighting occurred near the end of the tape, it would take up to two minutes for a tape to fast forward to the position of the clip, 4 seconds to digitize the clip, and then two minutes to rewind the tape. The rewinding step is necessary to preserve the condition of the tapes. As such, a single 4 second clip could take over 5 minutes in some cases, when considering the physical act of retrieving and putting the tape back. By utilizing strategies such as picking tapes with frequent species occurrences, this overhead was minimized to some extent, however some species were simply to rare to be picky with the tapes. Thus, to increase the collection of 500 images to 1000 images, two interns spent 4 weeks becoming familiar with the process of digitizing, and then parsing and sorting the resulting clips into still frames categorized by species.

When the new training set was used to train the inceptionV3 model, and the resulting learning curve graph is shown below as figure 1. From the graph, there appears to be a plateau occurring as the percentage of training data nears 100%. However, this test was ran on the collection when each class had on average 160 images, since a certain percentage of data had to be held out to use for validation. Performing a similar test with a more abundant collection of species gives the graph shown in Figure 2, which shows that at 300 training images per species, from a similar deep sea benthic image collection, there is total convergence. Thus, we can say with some certainty from these results that between 150 to 300 images per species will likely be enough training data for other, similar collections.

Figure 1

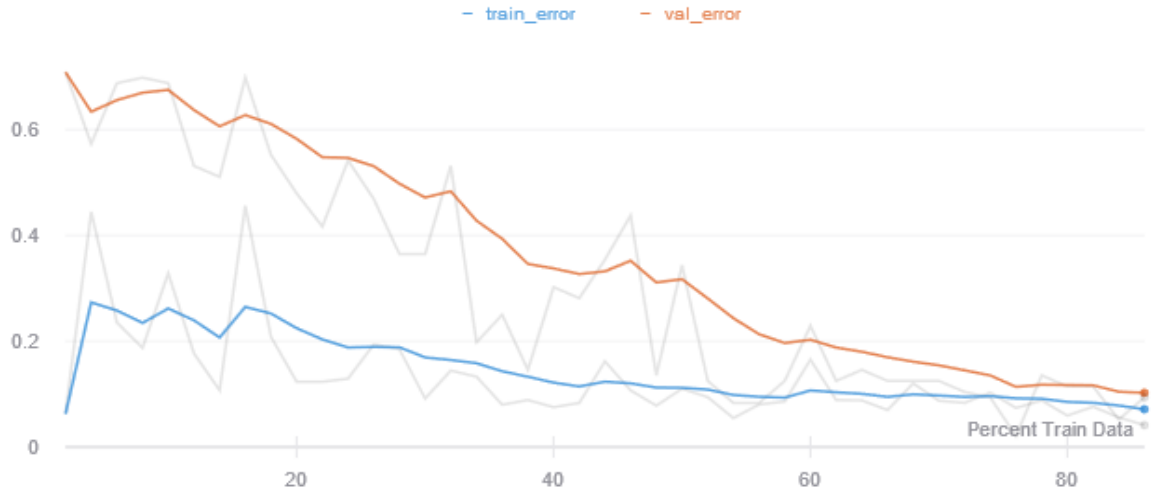
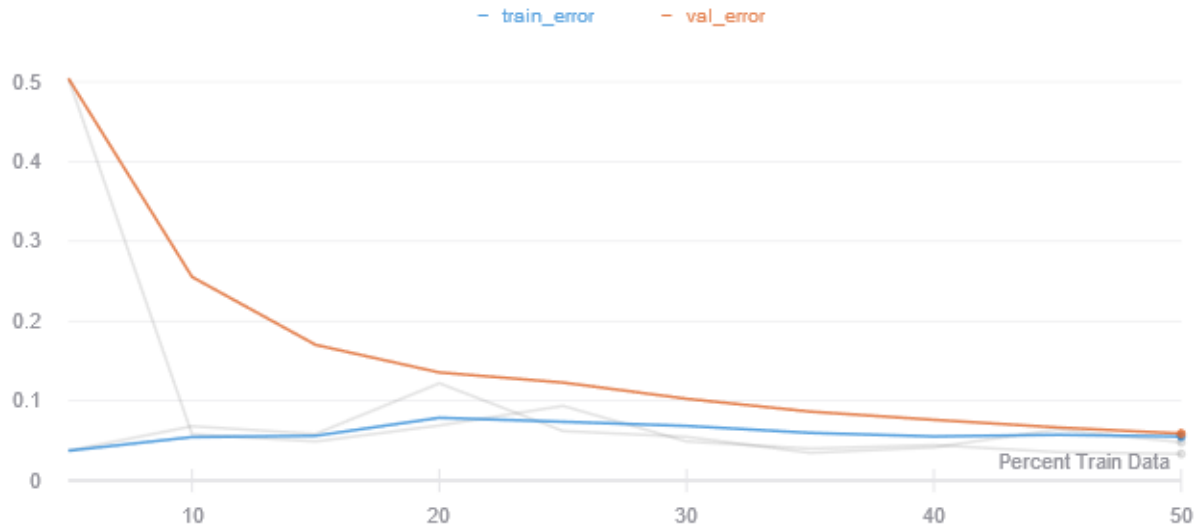


Figure 2



The second result which we wanted to understand was the cost of collection and implementation. The overall financial cost for the 10 weeks of work using GCP was $261.18, and a breakdown of the costs can be seen in figure 3. These overall costs are significantly lower than the price of a single unit of the cheapest GPUs used in the various tests. In our tests, GPU quotas had a cost of approximately $165 out of the $261 total spent. The remainder of the cost came from virtualized CPU cores, memory and RAM allocation, and network transfer costs.

Figure 3

SKU	Product	SKU ID	Usage	Cost	One time credits	Discounts	Subtotal
Nvidia Tesla T4 GPU running in Americas	Compute Engine	88B8-C3ED-03F0	92.93 hour	$88.28	–	–	$88.28
N1 Predefined Instance Core running in Americas	Compute Engine	2E27-4F75-95CD	852.48 hour	$26.95	–	–	$26.95
Storage PD Capacity	Compute Engine	D973-5D65-BAB2	414.16 gibibyte month	$15.37	–	–	$15.37
N1 Predefined Instance Ram running in Americas	Compute Engine	6C71-E844-38BC	3,355.44 gibibyte hour	$14.22	–	–	$14.22
Nvidia Tesla K80 GPU running in Americas	Compute Engine	ADAE-1096-790D	30.63 hour	$13.78	–	–	$13.78
Nvidia Tesla P100 GPU running in Americas	Compute Engine	7929-334B-6348	6.8 hour	$9.93	–	–	$9.93
SSD backed PD Capacity	Compute Engine	B188-61DD-52E4	43.17 gibibyte month	$7.34	–	–	$7.34

FUTURE WORK

Future work is required to tune the model more, in order to improve accuracy and other metrics. Hyperparameters such as learning rate and batch size need to be adjusted using hyperparameter tuning passes, which requires further computational power. Furthermore, allowing training to be preemptible would allow for more cost efficiency in the training process. Another possible change to test would be to reduce the number of frozen layers when transferring over a pretrained model. Doing this would increase the time required to train the model, since the amount of parameters would increase, however it could allow the model to be more specialized to the data we are interested in, rather than the generic ImageNet image collection the original model was trained on.

AKNOWLEDGEMENTS

I would like to thank my mentors, Danelle Cline and Duane Edgington, for the guidance they provided me with when I would become unsure of what came next in our project. I would also like to thank the MBARI video lab and all of the scientists who work there, for all of their help and kindness in letting us use their computers and resources to curate our image collection. I would also like to thank George Matsumoto and Madison Heard, who created a great space for all the interns to learn and grow together.

REFERENCES

[1] K. Simonyan and A. Zisserman, " Very Deep Convolutional Networks for Large-Scale Image Recognition ," *arXiv*, vol. arXiv:1409.1556, Sep. 2014.

[2] D. E. Cline, D. R. Edgington, and J. Mariette, "An Automated Visual Event Detection System for Cabled Observatory Video," in OCEANS 2007, 2007.

[3] Ye. Nathan, "Workflows for Automated Detection and Classification of Unlabeled Deep Sea Imagery," unpublished.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[5] Andrew, Ng, Machine Learning Yearning, Draft ed. Palo Alto: deeplearning.ai, 2018.