MBARI Monterey Bay Aquarium Research Institute

Real Time *In Situ* Plankton Detection Onboard Planktivore

Steven Patrick

Mentors: Dr. Paul Roberts, Dr. Monique Messié, Thom Maughan

Summer 2024

Keywords: machine learning, plankton, phytoplankton, ResNet18, planktivore

ABSTRACT

Phytoplankton produce approximately 80% of Earth's oxygen yet can form harmful algal blooms that poison food webs, making real-time monitoring critical for ocean health assessment. Traditional plankton monitoring using nets and microscopy is time-intensive and provides only discrete snapshots, limiting observational capabilities. This project developed and deployed a machine learning system for autonomous, real-time plankton classification aboard Monterey Bay Aquarium Research Institute’s (MBARI) Planktivore instrument on a long-range autonomous underwater vehicle (LRAUV). A pre-trained ResNet18 convolutional neural network was adapted to classify images into 15 morphology-based categories, and a bootstrapping approach iteratively built a training dataset of over 6,000 verified images over five weeks by training preliminary models, classifying new images, and manually auditing results. The system was successfully deployed in Monterey Bay during summer 2024, detecting two distinct phytoplankton blooms, diatoms and dinoflagellates, validated by independent chlorophyll sensor measurements, with estimated concentrations transmitted to shore in real-time. This work demonstrates that machine learning-based plankton classification can provide ecologically meaningful data on autonomous vehicles, transforming discrete sampling into continuous spatial monitoring and significantly expanding ocean observational capabilities.

INTRODUCTION

Phytoplankton, as the base of the ocean ecosystem, are pivotal to the ocean's health. Between being eaten by lower trophic level animals like Krill to the biggest animal on Earth, the blue whale, none would exist without plankton. Inversely, phytoplankton can also sicken or kill in mass blooming events called harmful algal blooms. These events can result in toxins being produced that can kill various creatures that prey upon them, or deoxygenate the water column upon phytoplankton's death. Studying phytoplankton concentrations as well as the toxin levels in the water is a constant effort done by various organizations and volunteers around the world. The Monterey Bay Aquarium Research Institute (MBARI) is one such organization that studies plankton.

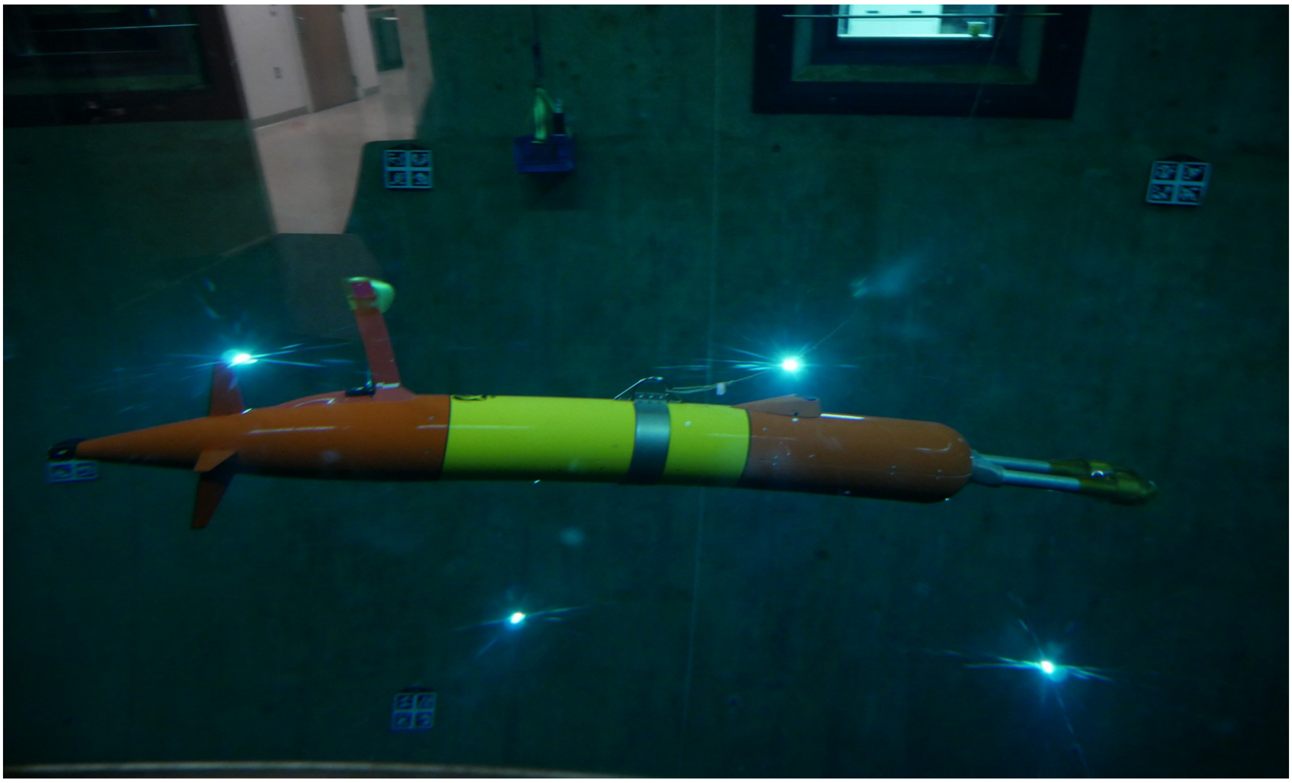

Figure 1: MBARI's Planktivore attached to the LRAUV inside MBARI's test tank.

The equipment under the right third orange section is Planktivore. From the middle yellow section to the end of the left third orange section is the LRAUV.

Photo taken by Steven Patrick.

Traditional methods to count plankton concentrations utilized fine mesh nets called plankton tow to collect concentrated plankton water samples. One would then count the different plankton species under a microscope (Santa 2022). This process is rather tedious, but MBARI has created more advanced equipment that is suitable for the 21st century. One such tool MBARI has created to study plankton *in situ* is called the Planktivore (Figure 1). This instrument photographs magnified, back-lit, 5 MP images of the water column. Software identification of regions of interest (ROI) and cropping of said regions result in a unique image of various plankton (An example of an ROI shown in Figure 2). At the time of writing this paper, Planktivore has been out to the Monterey Bay attached to a modular long range autonomous underwater vehicle (LRAUV) that has also been developed by MBARI. From these deployments, the Planktivore has cumulatively collected and reported millions of ROIs from the Bay that have been slowly categorized by hand.

This project therefore sought to develop a machine learning model to enable Planktivore to classify the ROIs into groupings like “Diatoms” and “Dinoflagellates”. Additionally, adding functionality for Planktivore to communicate to shore estimated plankton counts for scientists at MBARI to make real time decisions *in situ*.

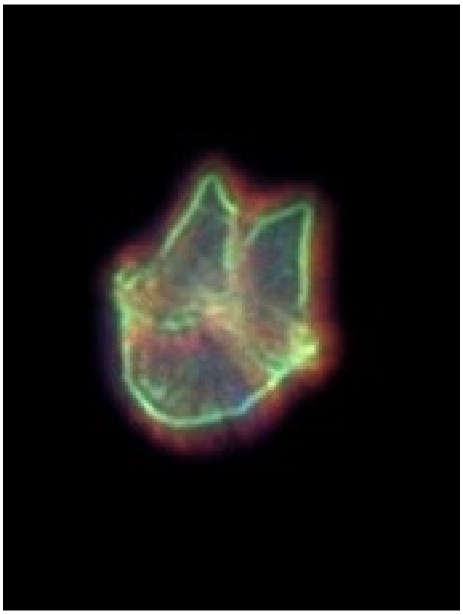

Figure 2: An Akashiwo taken by the Planktivore.

MATERIALS AND METHODS

DATASET CREATION

The foundation of any machine learning project is clearly specifying the goals and objectives the model must achieve. Having well-defined goals enables the development of effective training strategies that align with the desired outcome. For this project, a classification model was deemed appropriate given the task of assigning images to labeled categories. The training process relies on labeled datasets that enable the model to distinguish between different plankton types.

As this project began, a significant hurdle emerged: no existing dataset of images was available that closely matched the characteristics of the images produced by Planktivore. This presented a challenge in terms of finding suitable training data for the machine learning model. Fortunately, as mentioned earlier in the introduction, a valuable resource had been accumulated from previous deployments of Planktivore, a vast collection of images that could be leveraged to build a custom dataset tailored to this project's specific needs. To become familiar with the data, exploration of the images was necessary, along with supplemental reference materials to aid in manual classification (Figures 3 & 4).

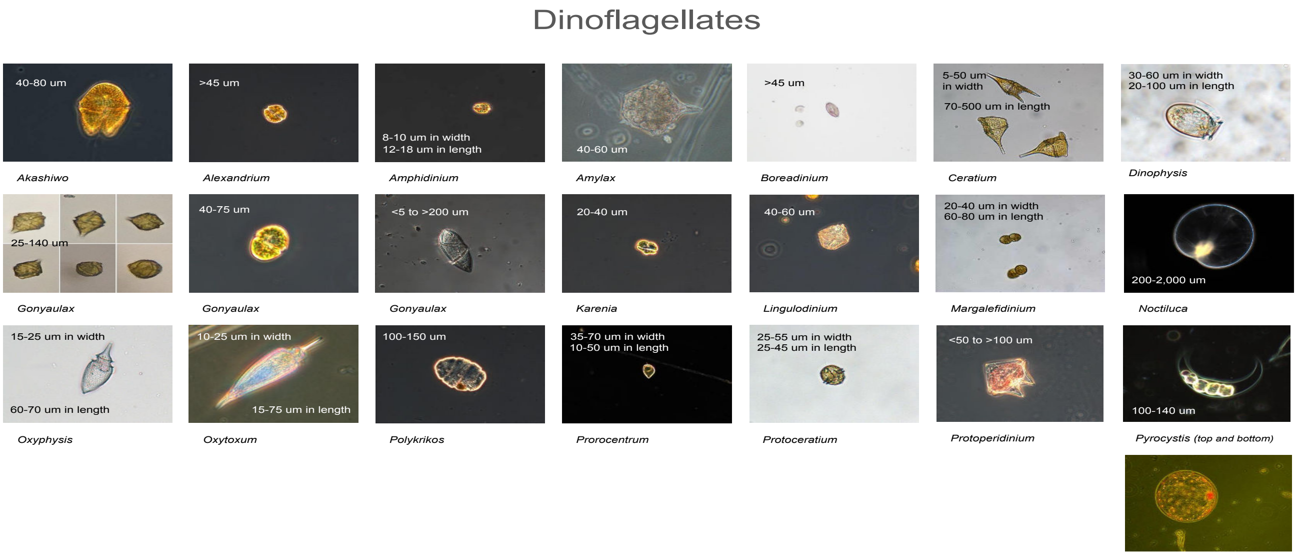

Figure 3: Dinoflagellate Poster used for manually identifying dinoflagellates.

Created by Steven Patrick.

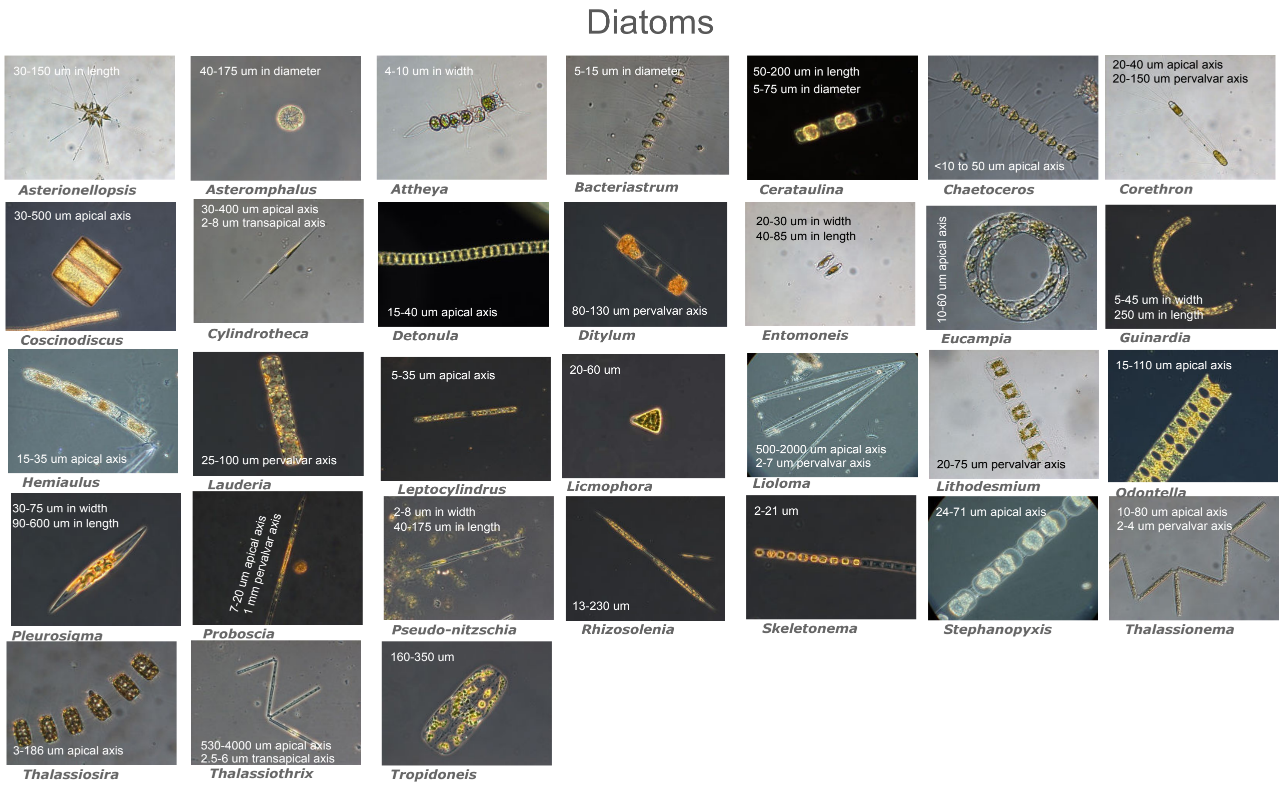

Figure 4: Diatom Poster used for manually identifying diatoms.

Created by Steven Patrick.

Once familiar with the data, classification categories needed to be established. Ideally, these classes would be based on the taxonomic rankings of genus and species. While this would enable the model to provide more precise identifications it was deemed infeasible within the limited internship timeframe. Additionally, concerns arose on the model's flexibility. Given the nature of where this model would be deployed, unpredictable images and undiscovered species are certainties. If a model were trained on a species-specific dataset, it would be ill-equipped for these situations.

Instead, the classification scheme utilized higher-level taxonomic groups combined with generic morphological descriptors. For example, rather than creating a class specifically for the dinoflagellate genus Akashiwo (Figure 2), the class "Dinoflagellate_Horns" was created to encompass all dinoflagellates with horn-like projections. This naming scheme allows the model flexibility; any similar-looking organisms can be categorized into the same class. A complete list of the 15 created classes can be seen in Figure 5.

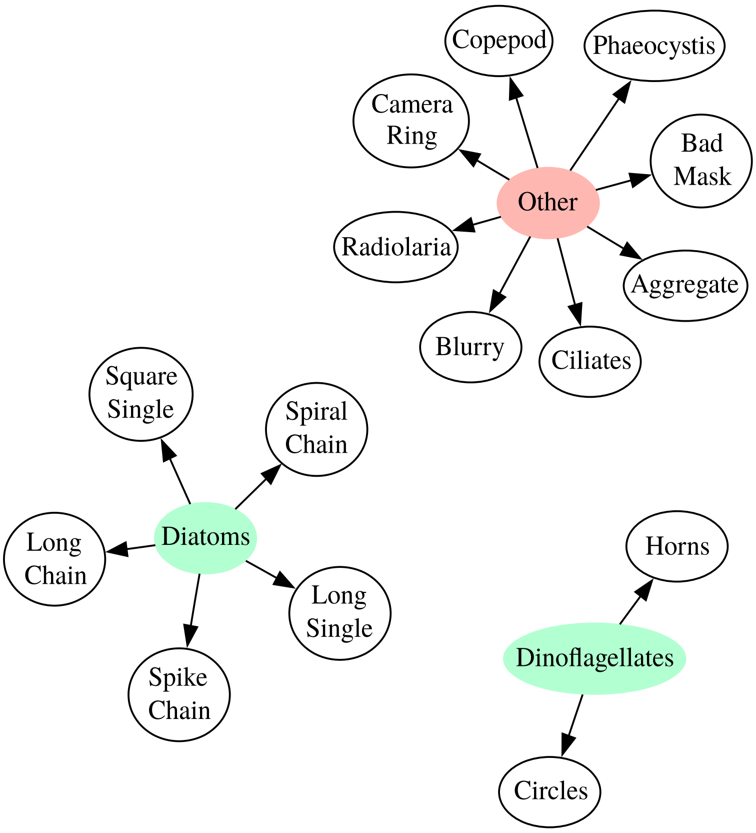

Figure 5: A list of all classification group. Super groups colored with green were the target for reporting to scientists in situ.

Created by Steven Patrick.

To support the model development workflow, a file folder hierarchy was established early in the project. This organizational structure created a known framework for directly loading images into the plankton classification model and would prove essential during the iterative dataset-building process described in the Bootstrapping section. Combined with RIMS, an in-house MBARI search tool for querying the Planktivore image library, this structure enabled an efficient workflow for manually tagging images during the initial dataset creation phase.

MODEL CODE CREATION

Due to the scope and timeline constraints of this project, developing a custom neural network architecture from scratch was not feasible. Instead, a pre-trained ResNet18 model was selected as the foundation for the classification model. Resnet18, a residual network of only 18 layers, was introduced by He et al. in their 2015 seminal paper "Deep Residual Learning for Image Recognition". This architecture has been extensively trained on ImageNet, a massive dataset containing over 14 million images across 1,000 categories (He 2015). ResNets address the degradation problem that occurs when training very deep neural networks by introducing skip connections (also called shortcut connections or residual connections) that allow gradients to flow more easily through the network during backpropagation. This innovation enables the training of substantially deeper networks than was previously possible as well as converging faster than plain/residual nets (He 2015). The faster convergence being key for this time sensitive project.

By leveraging this pre-trained model through transfer learning, the project could benefit from the low-level feature detection capabilities (edges, textures, shapes) that ResNet18 had already learned from ImageNet, significantly reducing the training time and data requirements needed to achieve good performance (Chilamkurthy 2017, Singhal 2020). Transfer learning is a technique where knowledge gained while solving one task is transferred to a different but related task, allowing models to achieve strong performance even with limited training data like with this project (Dang 2023). An additional practical consideration influenced this choice: ResNet18's relatively small size (approximately 11 million parameters) and computational efficiency made it suitable for deployment on the Nvidia Jetson embedded GPU aboard Planktivore's LRAUV platform, where computational resources and power consumption are limited compared to desktop or server environments.

To adapt ResNet18 for plankton classification, the model's architecture required modification. The original ResNet18 model outputs predictions across 1,000 ImageNet classes, but this project required classification into 15 distinct plankton groupings. This was accomplished by replacing the final fully connected layer of the network. The original layer, which had 512 input features and 1,000 output classes, was replaced with a new fully connected layer maintaining the 512 input features but reducing the output to 15 classes corresponding to the custom plankton categories defined in this project’s dataset.

The training strategy employed a two-phase approach to optimize model performance while managing the limited dataset size. In the first phase, all pre-trained layers of ResNet18 were frozen, resulting in their weights remaining fixed and not being updated during training; only the newly added 15-class output layer was trained. This allowed the new output layer to map the low-level feature detectors in the earlier layers to the new classifications. The second phase, by contrast, had all layers of the network unfrozen, allowing the entire model to now be fine-tuned (Chilamkurthy 2017, Singhal 2020). This step enabled the model to adjust its earlier-layer feature detectors beyond their ImageNet training to better suit the specific visual characteristics of the images generated by Planktivore. While the earlier layers had learned generally useful features, this fine-tuning phase allowed the model to refine those features for the back-lit, magnified plankton imagery that it would receive.

Given the persistent concern about limited training data throughout the project, a k-fold cross-validation strategy was implemented to maximize the utilization of available labeled images. The dataset was split into five folds using scikit-learn's RepeatedKFold function. During each fold iteration, four-fifths of the data served as the training set while the remaining fifth served as the validation set for evaluating performance. This process repeated five times with different validation folds, ensuring that every image in the dataset was used for both training and validation. This approach was accompanied by a custom early stopping mechanism that monitored the validation loss during training and would halt the process if the loss failed to improve by a minimum threshold (delta = 0.0) for a specified number of consecutive epochs (patience = 10 epochs). When triggered, early stopping would restore the model weights from the epoch with the best validation performance, ensuring that the final model represented the optimal point before overfitting began. These two mechanisms ensured that every image was available to be trained on while prevented overfitting from occurring.

BOOTSTRAPPING THE MODEL AND DATASET

It quickly became apparent that the number of manually tagged images in the initial dataset was woefully inadequate for the task of training a model. While manual tagging continued to increase the dataset size, it lagged far behind the pace needed for timely model development. Two approaches were conceived to accelerate dataset growth. The first involved using an unsupervised clustering model from a previous project; giving it unlabeled images, and specifying that it should organize them into a certain number of categories. While this model would not know the specific classes developed during the dataset creation phase, it would at least group similar images together. The second approach involved bootstrapping: training a classification model on the small hand-annotated dataset currently available, using that model to classify new images, and then manually auditing the results to ensure accuracy. Both approaches ultimately required manual verification of the automated classifications.

The unsupervised model was attempted first, but several downsides arose from this method. Given its unsupervised nature, images were rarely organized into groupings that aligned with the desired classification scheme. Additionally, the Python script crashed frequently when loading images into the graphics card's memory. While this could have been debugged and fixed given sufficient time, RIMS was already able to provide similar images based on search criteria comparable to what the unsupervised model offered, but without the crashes and misaligned categories. Since the unsupervised model still required manual auditing of its classifications, little was seen to be gained from continuing to debug it.

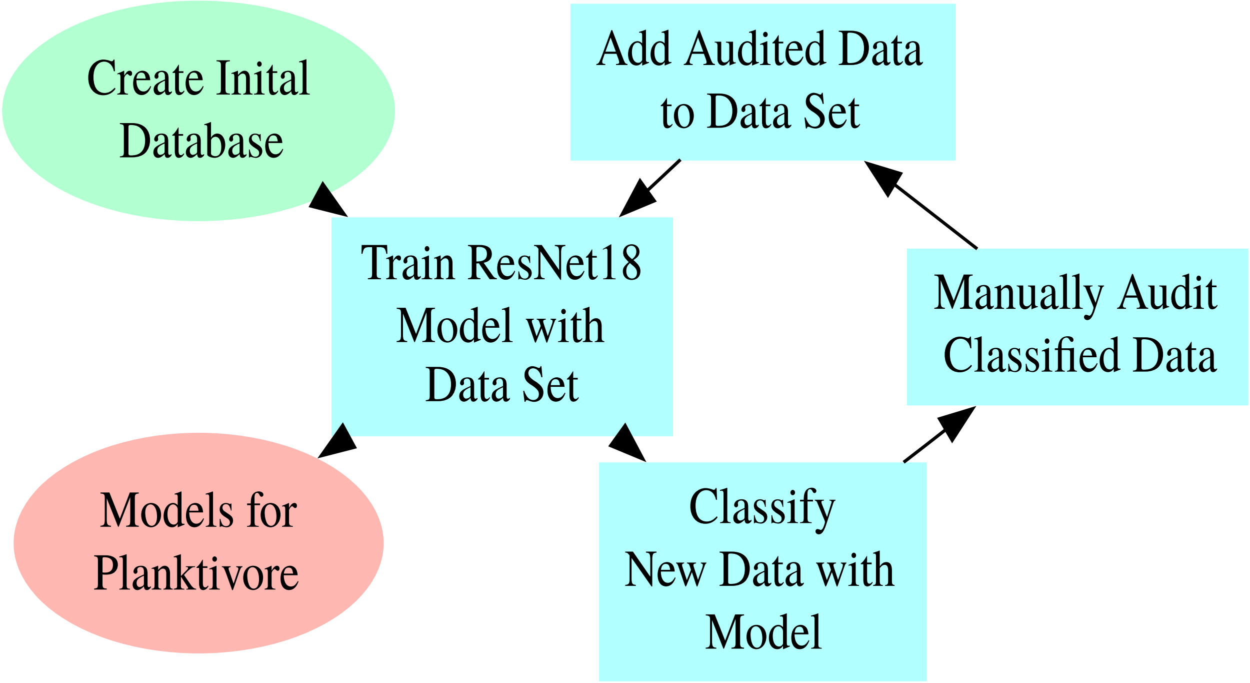
Create Inital Database
Add Audited Data to Data Set
Train ResNet18 Model with Data Set
Manually Audit Classified Data
Models for Planktivore
Classify New Data with Model

Figure 6: The Bootstrapping Cycle:

Starting with the initial database (in green), a loop began. After training a modified ResNet18 model with the available dataset, the model was used to classify new data. A manual audit was then performed on the new classified data before being added into the dataset. Next, the loop would begin again with training the model. Every instance of model training outputted a snapshot of the model that could be used in the Planktivore.

Created by Steven Patrick.

This left bootstrapping as the preferred approach for accelerating dataset growth. The bootstrapping process, illustrated in Figure 6, involved training a classification model on the initial hand-created dataset, then using that model to classify new unlabeled images. This required modifications to the model training code. Command-line arguments were implemented to allow the

Python script to operate in either training mode or classification mode. The classification mode needed additional functionality to organize newly classified images appropriately. Since the hand-created dataset already existed as images organized in category folders, continuing with this file-based approach made sense rather than developing a database system under tight time constraints. Classified images were organized into nested folders: the outermost folder used the date of classification, within which a timestamped subfolder indicated when the classification run occurred, and within that, the 15 category subfolders held the classified images. Additionally, a CSV file was generated with each classification run, recording the filename and assigned category for each image.

The bootstrapping cycle proceeded over several weeks. Each iteration followed the same pattern: classify a batch of new images using the current model, manually audit the results by reviewing each classified image and moving any misclassified images to their correct category folders, then add all verified images to the training dataset and retrain the model. Through this iterative process, the dataset ballooned to over 6,000 images by the end of this stage. While this is modest compared to massive datasets like ImageNet (14 million images), it would prove sufficient for the project's objectives.

The early iterations provided only marginal improvements over pure manual tagging, as the preliminary models frequently misclassified images. However, as the training dataset grew and model accuracy improved, the process became increasingly efficient. Despite this increasing accuracy, intensive auditing of the data added to the dataset continued throughout the process to ensure dataset quality remained high. This human-in-the-loop approach was essential—even as the model improved, manual verification caught edge cases, ambiguous organisms, and the occasional misclassification that could have corrupted the training data.

A downside of the bootstrapping process was the ever-increasing training time. As the dataset grew, each training cycle required more time to complete, with later iterations taking significantly longer than early ones. Despite this limitation, the bootstrapping approach successfully built a dataset of appropriate size within the project timeline while simultaneously developing and refining a classification model.

TESTING THE MODELS

Overfitting the models to the small sample size was a concern throughout this project's development. As previously mentioned in the Model Code Creation section, early stopping was implemented to reduce overfitting. However, this alone did not guarantee that the models would generalize well to new data.

The best way to test whether a machine learning classification model is overfitting is to evaluate it on data it has never seen. The dataset created for model training could not be reused for this testing phase since the model had already been exposed to all images in that dataset. As a result, a new test dataset had to be created specifically for evaluation purposes. Given the time constraint, this test dataset (Figure 8) was not hand-created like the training dataset (Figure 7), but was instead generated using the unsupervised clustering model previously mentioned. This was deemed acceptable since the test dataset would be used solely for model evaluation, not for training.

The key requirement was ensuring the test set had a substantially different distribution than the training set. While both datasets drew from the same pool of Planktivore images, the sheer number of images captured before this project began meant that overlap between the two sets was minimal. This difference in distribution would reveal any overfitting, as an overfit model performs well on its training distribution but poorly on different data distributions.

Comparing the training set (Figure 7) and the test set (Figure 8) reveals clear distributional differences. Due to the large number of diatom subcategories, the training set is heavily skewed toward diatoms compared to other categories, with dinoflagellates being the only comparably numerous category. The test data presents a stark contrast: while it contains many diatoms, the "Dinoflagellates" and "Other" categories are more numerous. This shift in data distribution would expose any overfitting. Figure 9 shows how various models performed when classifying the test dataset. All three models tested were able to classify a dataset with a different distribution than what they were trained on, though with varying degrees of success. The three models had been trained for different durations.

The “Bootstrapped-model” was the initial model trained and used throughout the bootstrapping process mentioned earlier. This model had the highest potential for overfitting given the small dataset it was repeatedly trained on. Some signs of this can be observed in Figure 9, though these could also be explained by issues in the test data itself, given its creation using an unsupervised clustering program rather than manual labeling.

Concerns about the “Bootstrapped-model”'s potential overfitting prompted creation of the "11-7-2024-model", which was trained for over a week on the final training dataset. Unfortunately, this model classified far too many images as dinoflagellates and diatoms while missing nearly every image from other categories. This model exhibited classic signs of overfitting: its classification distribution closely matched its training data distribution but diverged significantly from the test data distribution.

Finally, "Paul's model" was created by Dr. Paul Roberts using the same training script as the first two models. While this model's performance closely followed the “Bootstrapped-model”, some key differences were observed. Most notably, the “Bootstrapped-model” classified more images as diatoms and dinoflagellates.

Based on these tests, the decision was made to proceed with the “Bootstrapped-model” given its overall performance and better generalization to the test dataset. Despite being trained iteratively on a growing but consistently small dataset, it appeared to avoid the severe overfitting exhibited by the “11-7-2024-model” while maintaining reasonable classification accuracy across all categories.

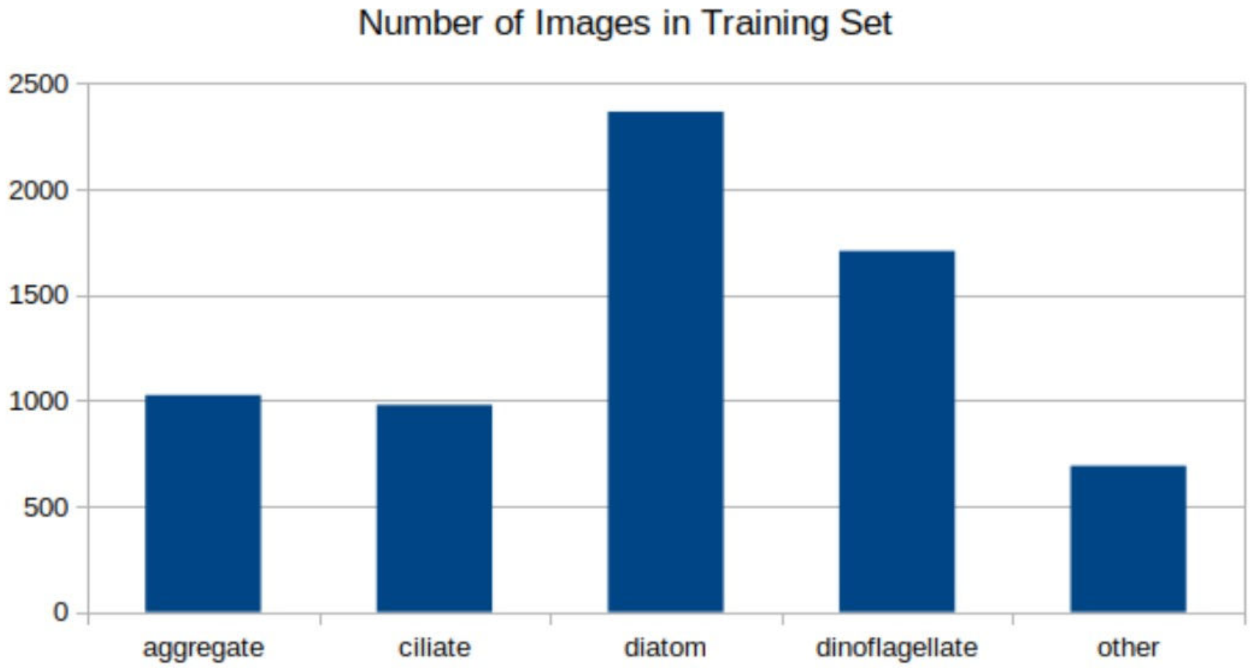

Figure 7: Distribution of Images in the Training Dataset.

Created by Paul Roberts using the training dataset made by Steven Patrick.

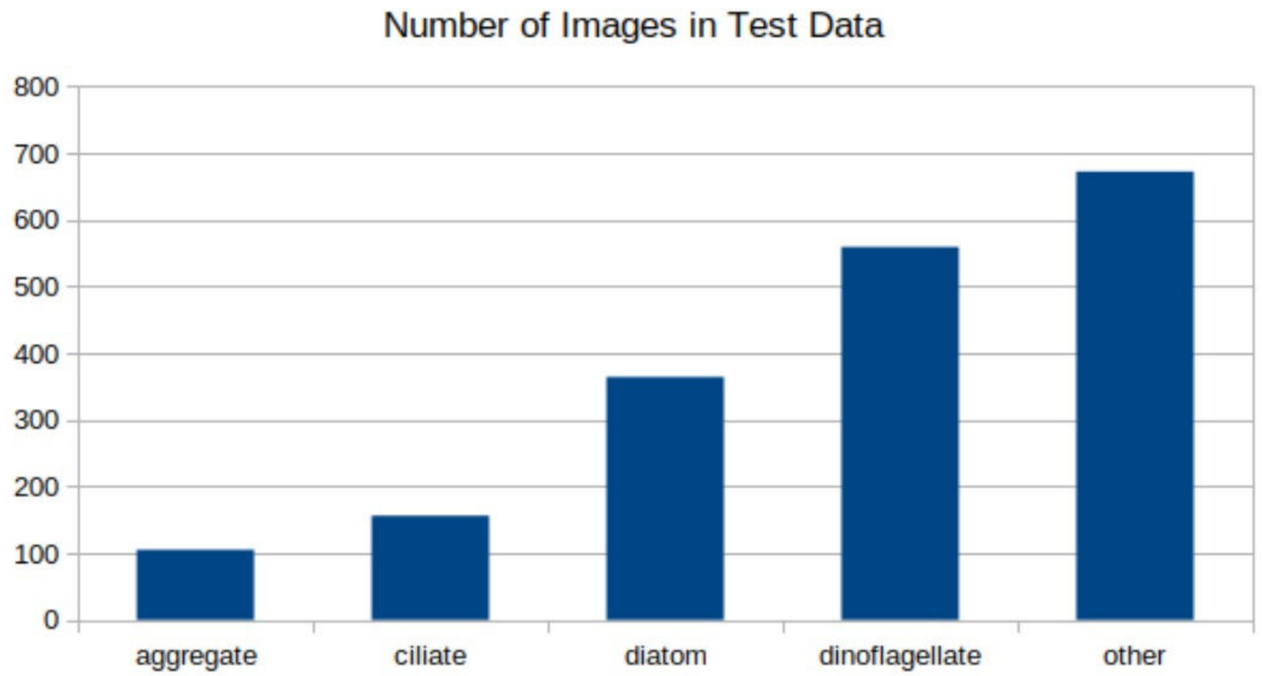

Figure 8: Distribution of Images in the Test Dataset.

Created by Paul Roberts using the test dataset created by Paul Roberts.

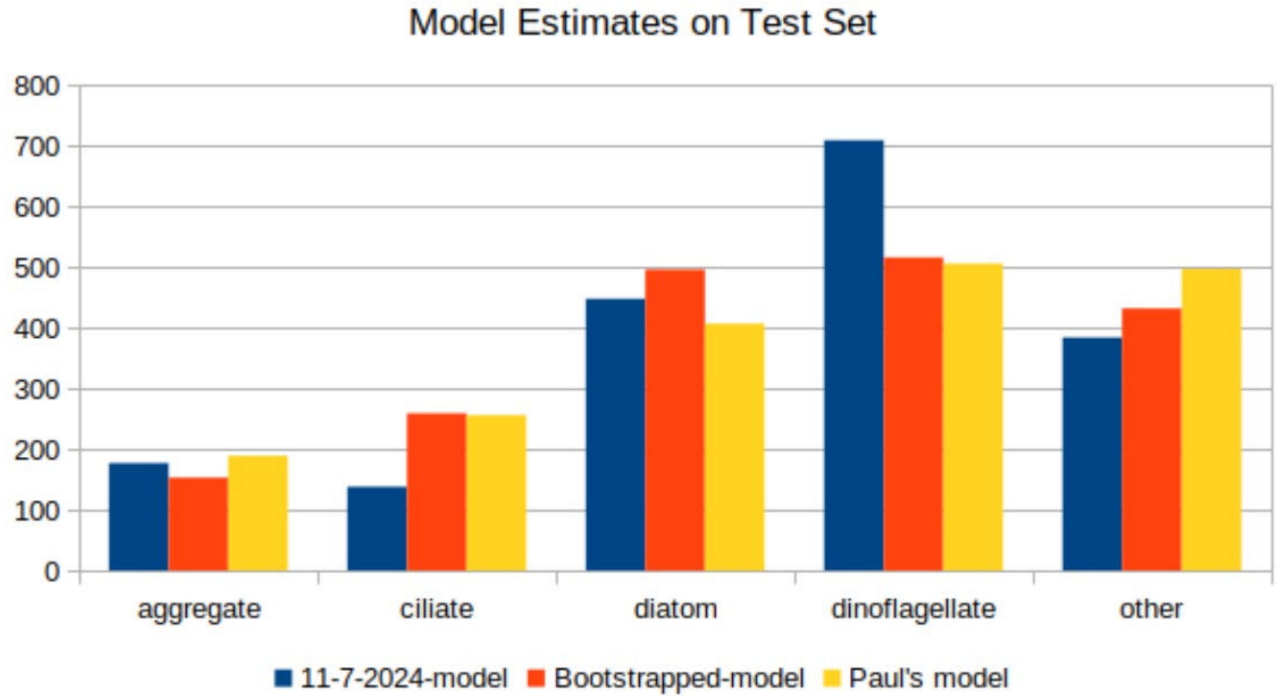

Figure 9: Categorize image counts of various model using test dataset.

Figure created by Paul Roberts.

INTEGRATION INTO PLANKTIVORE

The integration process began before model testing and selection were complete. This required designing the integration software to easily accommodate different models as they were developed, an essential feature with the ongoing model refinement during the project timeline and allowing for future projects to incorporate improved models into the Planktivore. Fortunately, the image classification code had already been developed during the bootstrapping process, making integration largely a matter of adapting existing code to work within Planktivore's operational environment.

Dr. Paul Roberts provided critical assistance in ensuring Planktivore's images were properly routed to the classification software. This involved rerouting appropriately sized images into designated folders where the model could access them. Dr. Roberts also provided essential code from the previous imaging software that detected ROIs in the water column and transmitted data back to shore. Of particular importance was the code that calculated estimated concentrations of ROIs given a processing interval. This concentration calculation code was modified to compute separate estimated concentrations for diatoms and dinoflagellates based on the classification model's output.

Once these concentrations were calculated, they needed to be transmitted to shore for real-time monitoring. Pre-existing functions integrated into Planktivore's communication system were leveraged for this purpose. These functions appended the concentration estimates to the data stream that the LRAUV routinely transmitted back to MBARI, allowing scientists on shore to monitor bloom dynamics as the mission progressed.

With all code finalized and installed on Planktivore, testing in MBARI's test tank commenced. This testing phase verified that all systems functioned correctly and that the new classification code would not adversely affect the equipment's operation. After successful completion of these tests, Dr. Roberts cleared Planktivore for deployment in the Monterey Bay.

RESULTS

During the deployment, a software bug affected data quality in the first half of the mission. The issue stemmed from an excessively long waiting interval between classification calls within the code. With the initial 30-second interval, the data collection process took too long, resulting in images from different depths accumulating within the classification folder Dr. Roberts created. When classification finally began on the accumulated images, organisms from multiple depths were lumped together, causing depth "smearing" visible in Figures 10 and 11. Dr. Roberts pushed a mid-mission hotfix that reduced the processing interval from 30 seconds to 10 seconds, allowing images to be classified more frequently and maintaining proper depth association. After this fix, the deployment proceeded smoothly with no further issues.

During the deployment, two phytoplankton blooms occurred: one dominated by diatoms and another by dinoflagellates (Figure 10 and 11). These detected blooms were independently validated by the onboard chlorophyll sensor (Figure 12).

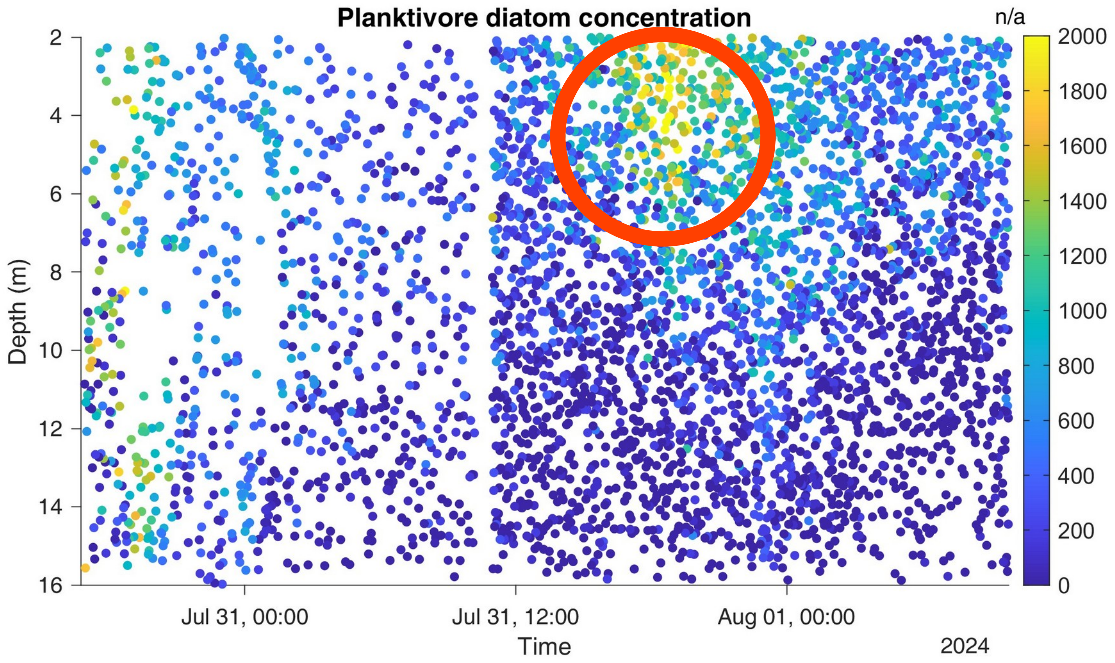

Figure 10: Estimated diatom concentrations. The red circle made on the graph encompasses the diatom bloom that was detected.

Graph created by Dr. Monique Messié using estimated diatom concentration data from the Plankton Classification model created by Steven Patrick.

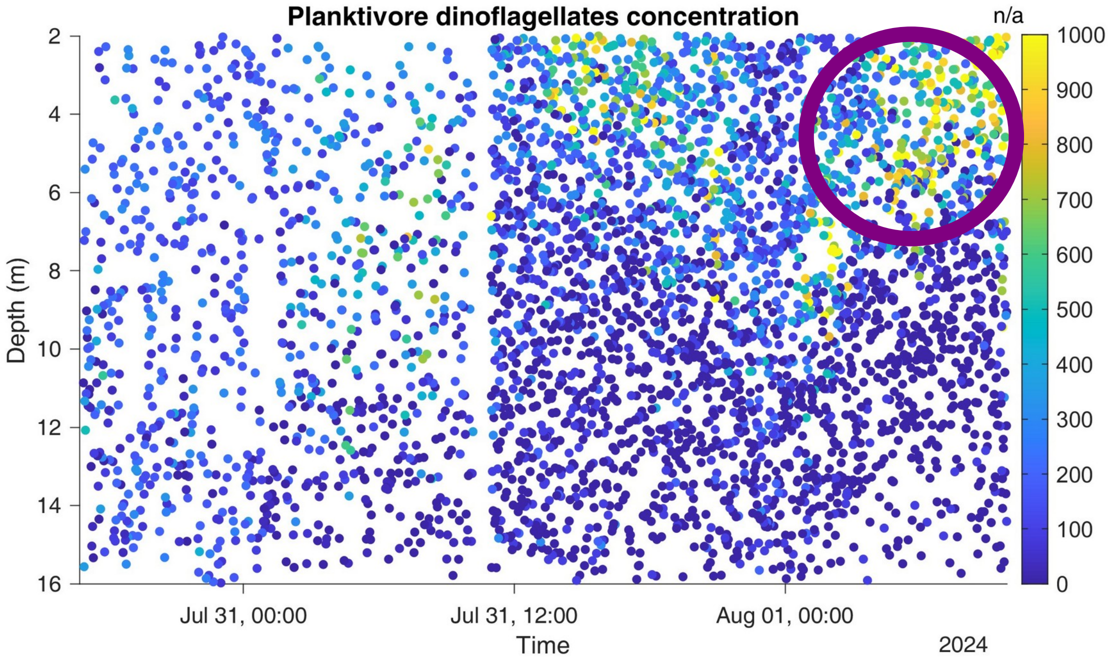

Figure 11: Estimated dinoflagellate concentrations. The purple circle made on the graph encompasses the dinoflagellate bloom that was detected.

Graph created by Dr. Monique Messié using estimated dinoflagellate concentration data from the Plankton Classification model created by Steven Patrick.

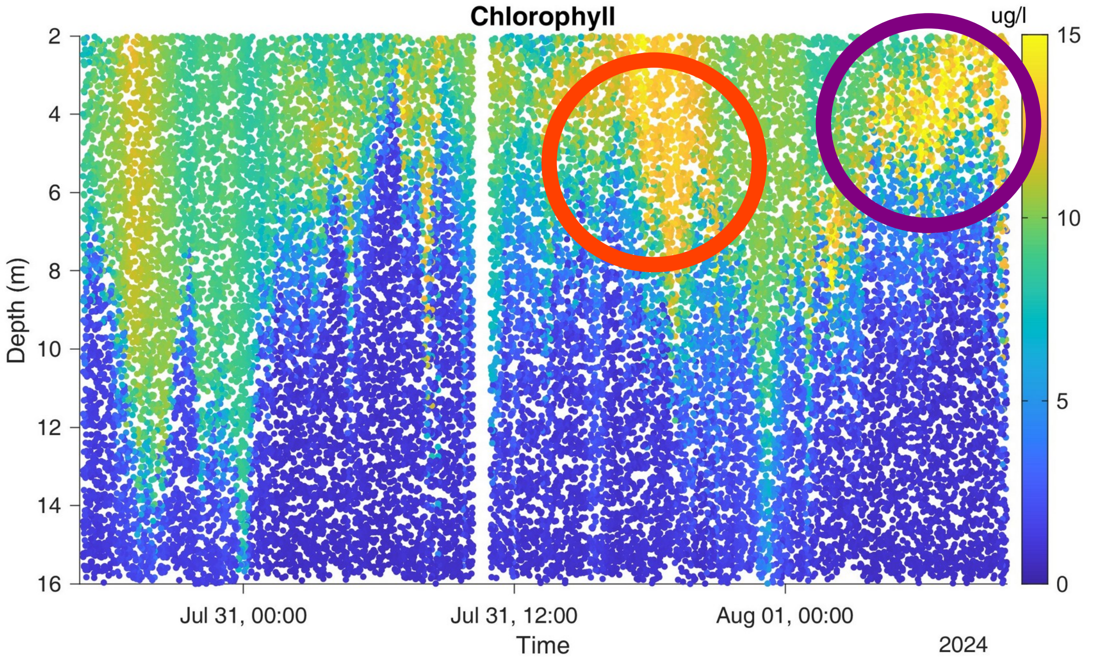

Figure 12: Chlorophyll concentrations graph showing the depth of detected chlorophyll levels. The corresponding diatom (red) and dinoflagellate (purple) bloom are circle with their corresponding color from previous figures.

Graph created by Dr. Monique Messié.

The diatom bloom, circled in red in the figures, was very apparent in both the estimated concentrations (Figure 10) and chlorophyll sensor readings (Figure 12), appearing from the surface down to approximately 10 meters depth in the water column. The dinoflagellate bloom was also clearly visible in the chlorophyll sensor data. However, the estimated concentrations from the classification model showed a weaker signal for dinoflagellates than for diatoms. Dr. Monique Messié had to increase the sensitivity of her data analysis program to make the dinoflagellate bloom easier to visualize in the concentration estimates (Figure 11). This discrepancy could result from either model performance limitations or from the bloom being dominated by phytoplankton that were smaller than what the model was trained to detect.

In addition to the depth concentrations, Planktivore also mapped the GPS locations of its route and integrated this spatial information with the concentration estimates transmitted back to shore (Figure 14, and 13). Figure 13 clearly shows the higher sensitivity required to visualize the dinoflagellate bloom that appeared in Figure 11. These spatial maps demonstrate the system's capability to not only detect blooms but also to characterize their geographic extent and location, providing scientists with actionable information about where blooms are occurring in the Monterey Bay.

Despite the weaker dinoflagellate signal, Dr. Messié confirmed that both detected blooms matched her expectations for diatom and dinoflagellate bloom characteristics in Monterey Bay. This validation marked the project as a success; Planktivore was now capable of identifying both diatom and dinoflagellate blooms in real time during autonomous deployments.

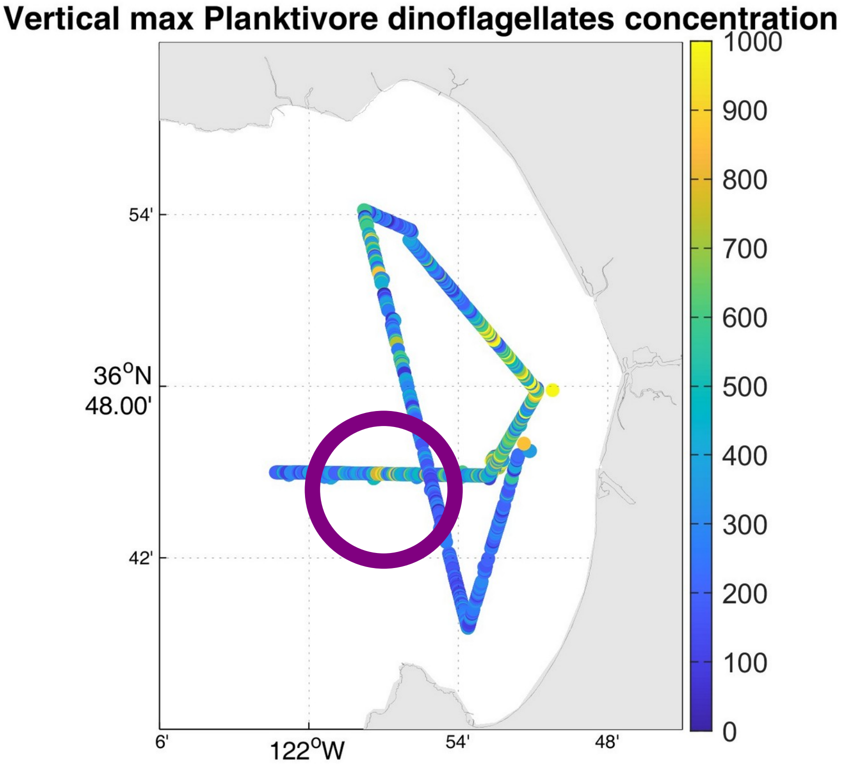

Figure 13: Dinoflagellate concentrations mapped to GPS locations.

Graph created by Dr. Monique Messié.

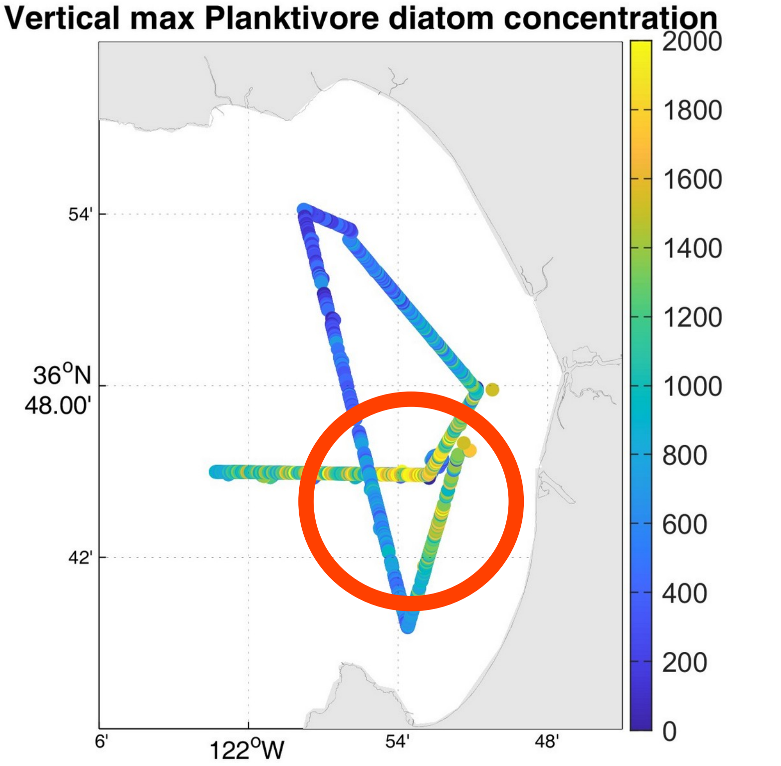

Figure 14: Diatom concentrations mapped to GPS locations.

Graph created by Dr. Monique Messié.

DISCUSSION

This project successfully demonstrated the feasibility of real-time, in situ plankton classification aboard an autonomous underwater vehicle. The deployment of the classification system on Planktivore resulted in the detection of two distinct phytoplankton blooms, one dominated by diatoms and another by dinoflagellates, both of which were independently validated by the LRAUV's onboard chlorophyll sensor. This validation confirms that the machine learning model developed through this project can accurately identify ecologically significant plankton concentrations in real-world ocean conditions, marking a significant advancement in MBARI's capacity for autonomous ocean monitoring.

The diatom bloom detection (Figure 10) was particularly clear, with estimated concentrations showing a distinct signal at approximately 10 meters depth that closely corresponded to elevated chlorophyll readings (Figure 12). The strong performance in detecting this bloom demonstrates that the model successfully learned to recognize diatom morphologies despite being trained on a relatively small dataset of approximately 6,000 images. The spatial distribution of the bloom (Figure 14) further illustrates the value of autonomous, real-time classification; the LRAUV was able to map the geographic extent and vertical distribution of the bloom while still deployed, providing data that would have been impossible to obtain through traditional ship-based sampling methods.

The dinoflagellate bloom detection (Figure 11), while successful, presented a more subtle signal that required increased sensitivity in post-deployment analysis to visualize clearly. Dr. Monique Messié confirmed that the bloom characteristics matched expectations for dinoflagellate distributions, validating the model's classification accuracy. However, the weaker signal relative to the diatom bloom raises important questions about model performance across different plankton types and size classes. The most likely explanation for this weaker signal relates to organism size distribution. Prior to this project, Planktivore had captured numerous small region of interest (ROI) images that were too small to permit reliable visual identification of the organisms. Following guidance from project mentors, these small images were excluded from the training dataset. If the dinoflagellate bloom was dominated by smaller species that fell below the size threshold used for dataset creation, the model would have been unable to classify these organisms, resulting in an underestimation of dinoflagellate concentrations. This hypothesis is supported by the fact that the chlorophyll sensor detected a strong signal (Figure 12), indicating substantial phytoplankton biomass that the classification model only partially captured. This limitation highlights an important area for future improvement: expanding the training dataset to include smaller organisms, even if their classification is less precise, may improve the model's ability to detect blooms dominated by small species.

A significant challenge encountered during deployment was a processing interval bug that affected data quality in the first half of the mission. The issue stemmed from the time required for the model to classify each batch of images. With the initial 30-second processing interval, the classification process took longer between imaging samples, causing new images from different depths to accumulate before the image classification began running. When the model finally started classification and began the current set of data, images from multiple depths were lumped together, resulting in depth "smearing" visible in Figures 10 and 11. Dr. Paul Roberts implemented a mid-mission hotfix that reduced the processing interval from 30 seconds to 10 seconds, allowing images to be classified more frequently and maintaining proper depth association. This fix immediately improved data quality, as evidenced by the clearer patterns in the second half of the deployment. This experience underscores the importance of thorough testing under realistic operational conditions and demonstrates the value of MBARI's capability to push software updates to deployed vehicles.

The bootstrapping approach proved highly effective for building a training dataset from scratch within the compressed timeline of a summer internship. Starting with zero labeled images, the iterative cycle of training, classification, manual auditing, and data expansion produced a dataset of over 6,000 verified images across 15 categories in approximately five weeks. While this dataset size is modest compared to large-scale machine learning datasets like ImageNet (14 million images), it proved sufficient to train a model capable of detecting ecologically relevant patterns in real deployment conditions. The success of this approach suggests that domain-specific applications of transfer learning can achieve useful performance with relatively limited training data, provided the base model (in this case, ResNet18 pre-trained on ImageNet) has learned sufficiently general visual features.

The project also revealed limitations in the current approach that should be addressed in future work. The decision not to implement data augmentation techniques: random rotations, flips, brightness adjustments, and other transformations that artificially expand dataset diversity, represents a missed opportunity to increase training data size without requiring more images. Data augmentation is particularly valuable when working with small datasets, as it exposes the model to more varied examples of each category and typically improves generalization performance. Given the ongoing nature of Planktivore deployments, future iterations of this work should incorporate data augmentation to improve model robustness.

The choice to use morphology-based classification categories (e.g., "Diatom_Long_Chain," "Dinoflagellate_Horns") rather than species-level identification proved appropriate for the project's scope and timeline. This approach provided flexibility to accommodate morphological variation and unknown species while still delivering ecologically meaningful information. The detection of diatom and dinoflagellate blooms demonstrates that even these broader taxonomic groupings provide valuable data for understanding phytoplankton community dynamics. However, the inclusion of quality control categories (Aggregate, Blurry, Bad_Mask, Camera_Ring) in the classification scheme was essential, as it allowed the model to filter out poor-quality images and artifacts rather than forcing them into biological categories. This design decision contributed to the reliability of concentration estimates by reducing false positives.

Comparison to traditional plankton monitoring methods highlights the advantages of the autonomous, in situ approach enabled by this project. Traditional methods using plankton tows and microscopy are labor-intensive, provide only discrete samples at specific times and locations, and introduce delays between sample collection and analysis that preclude real-time decision-making. In contrast, the integrated Planktivore system can continuously monitor plankton distributions over extended missions (days to weeks), adapt sampling strategies in response to detected blooms, and provide near-real-time data to scientists on shore. The spatial coverage demonstrated in Figures 13 and 14, mapping bloom distributions across multiple kilometers and depth ranges, would require dozens of ship-based sampling stations and represent weeks of laboratory work to achieve through traditional methods.

The successful transmission of estimated diatom and dinoflagellate concentrations back to shore during deployment represents a critical capability for adaptive sampling strategies. If scientists on shore observe elevated concentrations of a particular phytoplankton group in real-time, they could potentially redirect the LRAUV to sample that area more intensively, collect water samples for toxin analysis, or coordinate with ship-based teams for follow-up studies. This capability transforms autonomous underwater vehicles from purely data collection platforms into active participants in adaptive ocean observing systems.

Model selection based on the testing phase proved sound, as the "Bootstrapped-model" demonstrated remarkable performance during actual deployment. The concern about potential overfitting due to repeated training on a small, growing dataset did not materialize into significant performance degradation. This suggests that the combination of k-fold cross validation and early stopping effectively prevented overfitting despite the challenging training conditions. However, the test set evaluation (Figure 9) revealed that the model's performance varied across categories, with some categories, particularly the quality control categories, being more challenging to classify consistently. This variation in per-category performance likely contributed to differences in bloom detection clarity between diatoms and dinoflagellates, beyond the size-related issues discussed earlier.

Looking forward, the model and dataset created through this project provide a foundation for continued improvement. Every Planktivore deployment generates thousands of new images, and these can be periodically integrated into the training dataset to improve model accuracy and expand coverage of seasonal and spatial variation in plankton communities. The modular design of the classification code, with its ability to swap different trained models, facilitates this iterative improvement process. Additionally, the concentration estimation algorithm developed in collaboration with Dr. Roberts can be adapted to calculate concentrations for additional plankton categories beyond diatoms and dinoflagellates, enabling more detailed community composition monitoring.

In conclusion, this project successfully demonstrated that machine learning based plankton classification can be deployed on autonomous underwater vehicles to provide real-time, ecologically meaningful data about phytoplankton bloom dynamics. Despite challenges related to processing speed, organism size limitations, and limited training data, the system successfully detected and mapped two distinct phytoplankton blooms during its first deployment. The validation of these detections by independent chlorophyll measurements confirms the scientific utility of this approach. This work establishes a foundation for continued development of autonomous plankton monitoring capabilities that can enhance MBARI's ocean observing systems and contribute to improved understanding of phytoplankton ecology in Monterey Bay and beyond.

RECOMMENDATIONS

Several opportunities for improvement emerged from this project that future work should address to enhance the system's capabilities.

Database Architecture

Transitioning from the current folder-based organization to a structured SQL database for image classification and storage would provide significant benefits. The folder based structure allowed for quick turnaround during model training and bootstrapping, which was essential given the project's tight timeline. However, this approach led to folder clutter after deployment, an issue that was not apparent during the bootstrapping process when the focus was on rapid iteration. An SQL database would provide better organization, more efficient querying capabilities, and easier management of metadata such as timestamps, GPS coordinates, depth information, and classification confidence scores. This would be particularly valuable as the system generates thousands of classified images across multiple deployments.

Data Augmentation

Incorporating data augmentation techniques should be a priority for future model training efforts. As discussed in the Discussion section, the decision not to implement augmentation represents a missed opportunity to improve model flexibility. Data augmentation artificially expands dataset diversity without requiring additional manual labeling, which is particularly valuable when working with limited training data. Given that plankton images can appear at various orientations in the water column, augmentation would likely improve the model's ability to recognize organisms under varying imaging conditions and reduce overfitting.

Small Organism Detection

Expanding the training dataset to include smaller organisms is critical for improving bloom detection across all size classes. As evidenced by the weak dinoflagellate bloom signal during deployment, the exclusion of small ROI images from the training dataset created a significant blind spot in the model's detection capabilities. While small images are more challenging to classify accurately due to limited visual detail, including them in the training dataset would be preferable to missing entire blooms dominated by small species. This could be implemented by creating size-specific categories (e.g., "Dinoflagellate_Small").

These recommendations provide a roadmap for transforming the current proof-of-concept system into a production-ready tool for autonomous ocean monitoring. Each improvement builds on the foundation established by this project while addressing limitations that became apparent during development and deployment.

ACKNOWLEDGMENTS

I would first like to acknowledge my mentors: Dr. Paul Roberts for helping me out through this journey and teaching me the various pieces of equipment that MBARI uses and is creating. I greatly appreciate Dr. Roberts entertaining all of my questions, especially about the camera optics project he was working on. Dr. Monique Messié for our meetings on the project requirements and also explaining to me the success of the deployment. Thom Maughan for explaining the significance of my work and how impactful my project will be at MBARI. Additionally, I want to thank Brent Jones, who took me out with Dr. Roberts to both deploy and recover the Planktivore for its deployment. Also, thank you Denis Klimov for guiding me through my presentation and giving me recommendations throughout my slides. Your influence has helped not just the presentation, but also this paper and other endeavors in my life. Dr. Colleen Durkin for including me in meetings that not only exposed me to more projects at MBARI but made me feel like I was a member of the team.

Thank you Dr. George I. Matsumoto and Megan Bassett for organizing the internship program at MBARI and creating events for your interns to learn and network every week.

I want to thank my professors, Dr. Sam Ogden and Ryan Eller for writing excellent letters of recommendation for me to get this position. Their teaching and the opportunities they gave me allowed me to have this wonderful project.

Special thanks to Dr. Guillaume Liniger, Dr. Mauro Candeloro, Dr. William Oestreich and others from the postdoctoral room for making me a part of their community.

REFERENCES

Chilamkurthy, S. (2017, March 24). *Transfer Learning for Computer Vision Tutorial — PyTorch Tutorials 1.7.0 documentation*. Pytorch.org. https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

Dang, K. (2023, February 2). *Deep learning: Computer vision using transfer learning (ResNet-18) in Pytorch — Skin cancer…*. Medium; Medium. https://medium.com/@kirudang/deep-learning-computer-vision-using-transfer-learning-resnet-18-in-pytorch-skin-cancer-8d5b158893c5

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. In *arxiv*. https://arxiv.org/pdf/1512.03385

ImageNet. (n.d.). Image-Net.org. https://image-net.org/about.php

Santa Barbara Channelkeeper. (2022, June 28). *Monitoring Plankton to Protect Wildlife, People, and Ocean Health*. Www.sbck.org; Santa Barbara Channelkeeper. https://www.sbck.org/monitoring-plankton/

Singhal, G. (2020, May 5). *Transfer Learning with ResNet in PyTorch*. Www.pluralsight.com; Pluralsight. https://www.pluralsight.com/resources/blog/guides/introduction-to-resnet