CPSC 490 Report

Author: Nadia Irwanto
Advisors: Dr. Allison Hsiang, Professor Pincelli Hull, Professor Steven Zucker

Created: 26/01/2021. Updated: 13/05/2021.

0 Abstract

Automatic segmentation of planktonic foraminifera from microscopy images is an important task for paleoceanographic research. Planktonic foraminifera species identification is central to reconstructing sea-surface temperatures, but is usually time-consuming and relies on expert input. Documenting more of these organisms will therefore help streamline this challenging process. However, generating the necessary training data for training classifiers is a rate-limiting step in this undertaking, and would benefit enormously from improved algorithms for segmenting foraminifera from images of bulk samples taken on microscope slides. In this vein, we train an instance segmentation model to perform highly effective segmentations of variable images taken from a global planktonic foraminifera census data set of the Pliocene ocean. Our dataset consists of 188 images containing 2,203 instances of planktonic foraminifera, carefully selected to ensure species as well as image characteristic diversity, and then annotated and reviewed by members of our lab. We use the well-known, two-stage Mask-R-CNN as our baseline model, and compare it with the single-stage SOLOv2 in order to test two fundamentally different strategies in instance segmentation. The Mask R-CNN model performs exceedingly well, and is further optimized by experimenting with different backbone architectures (ResNet-50+FPN, ResNet-101+FPN, ResNeXt-101+FPN, ResNet-50+DC5) and learning rates (0.025, 0.0025, 2.50E-04, 2.50E-05). We also investigate how the amount of training data affects model performance, by systematically sampling from the training images, as well as whether the model is able to handle more difficult cases in the dataset, by visualizing and inspecting the model predictions for each image. Our final model uses a Mask R-CNN architecture with a 101-layer ResNeXt and Feature Pyramid Network (FPN) backbone, and achieves a mean Average Precision (mAP) of 89.42.

1 Background

1.1 Climate Change and the Mid-Piacenzian Age

The current phenomenon of global warming is unprecedented over the last few millennia due to its rapid timescale. However, climate change is a complex process that makes it difficult to quantify each driving force as well as project future trends. By studying periods of warming in the geological past, scientists can apply lessons learned from these periods to our understanding of anthropogenic climate change today.

The Mid-Piacenzian Age of the Pliocene Epoch, about 3 Ma, is a potential if imperfect analogue for near-future climate conditions. The global mean temperature of the Piacenzian Earth is estimated to have been approximately 2–3°C warmer than at present[1], which is within the range of warming predicted for the end of the twenty-first century[2]. At the same time, atmospheric CO_2 levels are similar to those today, at about 380-420 ppm[3]. Additionally, the similarity of the Mid-Piacenzian

[1] de la Vega et al., 2020
[2] Collins et al., 2013
[3] de la Vega et al., 2020

Earth to the modern one in terms of continental positions, land elevations and oceanic circulation patterns allows us to draw many parallels between climate-induced environmental changes then and now. Thus, understanding climate dynamics during the Pliocene is paramount to our ability to predict, adapt to, and mitigate the effects of future climate change.

1.2 The PRISM Project

The Pliocene Research, Interpretation and Synoptic Mapping (PRISM) Project[4] is a collaborative data analysis and climate modeling effort that aims to reconstruct this most recent warm period. The PRISM reconstruction is a high-resolution and multi-faceted description of the Pliocene, and is widely used to model ground-truth simulations of the past and to form the basis of future model projections.

Planktonic foraminifera serve as the primary source of data for deriving Pliocene ocean temperatures. Planktonic foraminifera are miniscule (0.05–1 mm), single-celled protists that live near the surface of marine environments. They possess several characteristics that make them ideal environmental and bio-chronological indicators. As planktonic organisms, they have a long geologic record and a widespread geographical distribution. Their shells, or tests, are made of predominantly calcium carbonate and are easily preserved; planktonic foraminifera thus exist in large abundance as microfossils through time and space. In addition, each species lives in a well-defined ecological niche, and hence the quantitative analysis of species assemblages is widely used to reconstruct environmental conditions.

As many Piacenzian species are extant, estimating paleotemperatures based on modern calibrations is possible[5]. Paleoceanographers measure the isotopic values in their tests to reconstruct records of climatic parameters such as sea-surface temperature. Data from fossil planktonic foraminifera can be used to determine how species that still exist today responded to the warmer environmental conditions of the Pliocene Epoch, or to investigate how the morphology of a species responds to varying climatic conditions over time. These insights can provide useful context for understanding modern global warming.

However, doing these types of analyses requires that individual specimens be identified to the species level, as different species exhibit different isotopic fractionation curves[6] (i.e., biological vital effects), occupy different ecological niches, and form different test morphologies. The species identification task is highly time-consuming to perform manually, and requires extensive taxonomic expertise.

1.3 Endless Forams

A challenge faced by researchers using planktonic foraminifer data is that the high intraspecies morphological variability, combined with subtle and ambiguous morphological species concepts/boundaries, makes it difficult to correctly identify an organism's species without extensive taxonomic training. To address this issue, the Hull lab at Yale previously released Endless Forams[7], a database of more than 34,000 planktonic foraminifera, for the purposes of capturing intraspecific

[4] USGS, 2016
[5] Dowsett et al., 2013
[6] Ezard et al., 2015
[7] Hsiang et al., 2019

morphological variability and aiding in taxonomic training. As part of the group's data collection efforts, they developed an image processing software, AutoMorph[8], to segment and extract 2D and 3D shapes of planktonic foraminifera from slides images viewed under light microscopy using traditional image processing algorithms.

The PRISM images[9] are a valuable addition that will allow researchers to document far, far more of the Pliocene species. However, the color, lighting and background variation of the PRISM dataset pose a challenge to AutoMorph's non-neural net approach. More specifically, AutoMorph employs RGB and gamma filters as well as user-defined threshold values that are not easily generalized over all the objects in the PRISM images (see Figure 2 for a brief description of the challenging characteristics found in the PRISM dataset). Hence, the goal of this project is to train an instance segmentation model that is able to robustly identify and segment all instances of planktonic foraminifera in a microscope slide image in spite of image variability. By using a machine learning approach, the model should not only perform well on the PRISM images, but also be easily extendable to future datasets.

2 Instance Segmentation Models

Instance segmentation is the computer vision task of identifying the boundaries of each individual occurrence of objects in an image. Strategies that have emerged in recent years can be divided into two-stage or single-stage procedures. Each category can be subdivided further: two-stage strategies adopt either an instance-first approach or a segmentation-first approach, and single-stage strategies use either local masks or global masks to represent instances, based on whether the masks fit to the size of each instance or preserve the spatial dimensions of the input image, respectively.

2.1 Mask R-CNN

Mask R-CNN[10] is a classic, breakthrough instance segmentation framework that continues to serve as a foundation for newer variants and achieve competitive results on various benchmarks. The network is part of the Region-based CNN (R-CNN) family of models, which generate a finite number of candidate Regions of Interest (RoI)s to process. Mask R-CNN adapted Faster R-CNN[11] to the task of instance segmentation by adding a mask prediction branch on top of the existing class prediction and bounding box prediction branches, and replacing the RoIPool operation with a more spatial-location-preserving RoIAlign layer.

The baseline model for this paper is a Mask R-CNN that uses a ResNet-50+FPN backbone for feature extraction, followed by the standard convolutional (conv) and fully connected (FC) heads for mask and box prediction, respectively. The ResNet-50+FPN backbone consists of a 50-layer residual network[12] (ResNet) followed by a Feature Pyramid Network[13] (FPN) to extract RoI features at

[8] Hsiang et al., 2017
[9] Dowsett et al., 2015
[10] He et al., 2017
[11] Ren et al., 2017
[12] He et al., 2016
[13] Lin et al., 2017

different scales. The model achieves a mask AP of 35.9[14] on the Microsoft COCO 2017 dataset[15], for the task of instance segmentation. COCO is a large-scale image dataset (330K images, 1.5 million object instances, 80 object categories) that is the gold standard benchmark for evaluating the performance of state-of-the-art computer vision models on the tasks of object detection, segmentation, and captioning. An explanation of the AP metric can be found in the Section 5.1.

2.2 SOLOv2

In contrast, SOLOv2[16] is a single-stage model that directly outputs a global mask. A key idea is that the model should predict the "instance category" of each pixel, analogous to the semantic category for semantic segmentation. The instance category is defined by location and size, as a proxy for shape. To find the location of each instance, the center position of any instance is approximated by dividing the input image into a grid of S x S cells, and predicting the class of each cell; to find the size, the model assigns each instance to different levels of a FPN and generates a full-resolution global mask for each cell. Thus, instance segmentation is reduced to two pixel-level classification problems. SOLOv2 is one of the few methods that operates on global masks, and is truly bounding-box-independent and hence regression-free.

The SOLOv2 model in this paper also uses a ResNet-50+FPN architecture. The model achieves a mask AP of 37.5[17] on the COCO 2017 dataset, and was shown by the authors to generate higher-quality masks than Mask R-CNN.

3 Dataset

The images used in this project are taken from a global planktic foraminifera census data set of the Pliocene ocean[18], curated for the PRISM Project. The census consists of 593,676 individuals identified to the species level in 1,957 Pliocene age ocean sediment samples. The foraminifera are grouped by species and fixed in place on micropaleontology slides with a 60-cell numbered grid. The slides are physically archived at the US Geological Survey in Reston, Virginia, USA.

Images of the slides were taken using a 5-megapixel Leica DFC450 digital camera mounted on a Leica Microsystems DM6000M compound microscope with a drive focus and motorized x-y scanning stage. The microscope system is controlled by Surveyor Software (Version 7.0.1.0, Objective Imaging Ltd) run on a Dell computer (3 TB Solid-State Drive, 3.7 GHz processor) coupled to an OASIS-blue 3 Stage Controller (Objective Imaging Ltd) and a 5-megapixel Leica DFC450 digital camera. The microscope software creates an Extended Depth of Focus (EDOF) image, a composite image that combines the in-focus areas from a stack of image slices captured at discrete focal depths. Each slide has multiple image stacks and hence multiple EDOF images, one per species.

A subset of slides from a small subset of sites were selected to give a global snapshot of species diversity during the mid-Piacenzian Warm Period. Out of this subset, 11 slides were further

[14] https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md#coco-instance-segmentation-baselines-with-mask-r-cnn

[15] Lin et al., 2014

[16] Wang et al., 2020

[17] https://github.com/WXinlong/SOLO#models

[18] Dowsett et al., 2015

selected to represent the diversity of image characteristics that the model should be able to handle. A brief description and example of each case is shown in Figure 2 in the results section.

Images were labeled using the data annotation platform Segments.ai. Segments.ai was selected from several image annotation tools because of the accuracy and ease of use enabled by its features, such as automated segmentation and model-assisted labeling. Images were labeled with only one class, that of planktonic foraminifera, even though a small subset of instances are in fact benthic foraminifera. This approach keeps the segmentation task as simple as possible. Differentiating benthic and planktonic foraminifera, as well as identifying different species of planktonic foraminifera, are future extensions of this work.

All training images were annotated by Nadia Irwanto, and then reviewed by Allison Hsiang or Elizabeth Sibert to verify that the ground truth masks and labels were accurate. The resulting dataset generated from these 11 slides has 188 images, containing 2,203 instances of planktonic foraminifera. The dataset was then split with a train/val/test ratio of 70/15/15; i.e., 70% of the images were allocated for training, 15% for validation, and 15% for testing.

4 Experiments

4.0 Model Implementation and Default Configuration

The instance segmentation models used in this study are implemented using Facebook AI Research's Detectron2[19] framework. All models were pre-trained on the COCO 2017 dataset, with a 3x training schedule (~37 COCO epochs). Unless otherwise stated, the training procedure for the PRISM dataset was configured to run for 500 iterations, with a warm up period of 200 iterations to reach a base learning rate of 0.0025, and a 10x decay at iteration 400. The selected optimizer was Stochastic Gradient Descent with a momentum coefficient of 0.9, and the mini-batch size was fixed to 2 images.

These settings generally follow the default configurations in Detectron2, except for a few adjustments. The learning rate was set at an appropriate scale, to ensure that the model converges. The number of iterations and mini-batch size were significantly lowered to take advantage of the PRISM dataset being simpler than the COCO dataset, and hence save on computation cost.

4.1 Mask R-CNN Backbone Architecture

The backbone network of the Mask R-CNN extracts feature maps from the input image. Four different backbone combinations were tested:

1. ResNet-50+FPN: 50-layer ResNet with a FPN.
2. ResNet-101+FPN: 101-layer ResNet with a FPN. This is a deeper version of the first backbone, and tests the commonly held belief that more layers leads to better performance, as the model is able to learn more complicated features.
3. ResNeXt-101+FPN: 101-layer ResNet, enhanced with the “cardinality” dimension. The cardinality defines the size of the set of transformations, and greatly reduces the number of model parameters while even being more effective than going deeper or wider. More details can be found in the original paper.[20]

[19] https://github.com/facebookresearch/detectron2

[20] Xie et al., 2017

4. ResNet-50+DC5: 50-layer ResNet that uses dilated convolutions, also known as atrous convolutions, in conv5, i.e. Dilated-C5 (DC5). The filter's dilation rate is increased from 1 to 2, and this increases the receptive field size while keeping the number of parameters and operations the same. This is used by the Deformable ConvNet paper,[21] which found that accuracy increased for all tasks when using larger dilation values. It builds on previous research that found that the effective receptive field of units in deep convolutional networks is only a portion of the theoretical receptive field, and hence tends to be too small.[22]

The original Mask R-CNN paper experimented with the first three backbones, and found expected gains from both going deeper as well as using ResNeXt over ResNet.[23]

4.2 Hyperparameter Tuning

When building machine learning models, the values of the hyperparameters can significantly impact model metrics. Therefore, an important step in the machine learning workflow is to identify the best hyperparameters for a model. Hyperparameters are variables that determine how the network is trained, such as the learning rate, batch size, momentum, and weight decay. Whereas model parameters are learned during training, hyperparameters are fixed and set manually before training. Because the model before tuning already performs very well, we focus on optimizing just the learning rate, one of the more fundamental hyperparameters, and employ grid search with a log scale to test learning rates of [0.025, 0.0025, 0.00025, 0.000025].

4.3 Training Dataset Size

Annotating images to train an instance segmentation model is extremely time consuming and tedious. For instance, generating the relatively modest training set for this project took 1.5 months. Thus, the question of how the number of training images affects model performance is of great interest. In this set of experiments, the baseline model was trained on variable-sized subsets of the full training data set. The training data sample ratios tested were [0.1:0.9:0.1] (i.e., 0.1 to 0.9 inclusive, in intervals of 0.1), where 0.1 represents sampling 10% of all training images, in order to simulate data scarcity and create a smaller training dataset.

For each sample ratio, five repeated training runs were conducted, using a different random seed to sample the training images for each run. The final metrics take the average across the five training runs; this ensures that any difference in performance can be attributed to a change in the ratio, rather than to a change in the distribution of the data in the training set. For similar reasons, the same validation and test sets were used across all runs and sample ratios, so as not to make the evaluation metrics dependent on the distribution of the validation or test sets.

5 Results

5.1 Evaluation Metrics

The mean Average Precision (mAP) is the standard quantitative evaluation metric for instance segmentation models.

[21] Dai et al., 2017
[22] Luo et al. 2017
[23] He et al., 2017

The first step of deriving the mAP is to calculate the intersection over union (IOU) of each prediction. The IOU takes the predicted and ground truth mask, and divides the area of their intersection by the area of their union:

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

The IOU score of each prediction is compared to a threshold to identify true positives (TP), false positives (FP), and false negatives (FN). These values are used to calculate the model's precision and recall:

$$Precision = \frac{TP}{TP+FP}; \quad Recall = \frac{TP}{TP+FN}$$

Precision measures how accurate the model predictions are, while recall measures how well the model detects everything it should detect. Setting a lower threshold increases the precision but decreases the recall, and vice versa, due to the inherent tradeoff between the two metrics. The precision and recall can be plotted in relation with each other, and the Average Precision (AP) is defined as the area under the precision-recall curve. In practice, the AP is calculated as the average of precision values, taken at equally spaced recall levels. In COCO evaluation, the AP is the mean of 101 precision values that correspond to recall values of [0:.01:1] (i.e., 0 to 1, in intervals of 0.01).

The standard COCO metric for AP calculates the average precision score over an IOU range of [.5:.05:.95]. Other metrics include the AP$_{50}$ and AP$_{75}$, where the subscript denotes the value of the fixed IOU threshold. The mAP is simply the AP scores averaged over different classes. AP and mAP are often used interchangeably, as the difference is clear from context.

5.2 Quantitative Results

	AP	**AP$_{50}$**	**AP$_{75}$**
Model Architecture			
Mask R-CNN*	**88.19**	98.94	98.78
SOLOv2	0.62	0.98	0.56
Mask R-CNN Backbone Architecture			
ResNet-50+FPN*	89.10	98.96	97.93
ResNet-101+FPN	88.86	98.00	97.90
ResNeXt-101+FPN	**89.42**	99.00	97.98
ResNet-50+DC5	86.66	98.99	97.79
Training Size			
0.1 (13 images)	86.29	95.15	94.09
0.2	87.96	97.18	96.09
0.3	88.23	97.51	96.80
0.4	88.40	97.58	96.68
0.5 (65 images)	88.36	97.59	96.69
0.6	88.81	93.37	97.08
0.7	88.91	98.36	97.31
0.8	88.80	98.17	97.09
0.9 (117 images)	**89.00**	97.99	96.89
Learning Rate			

0.025	87.99	98.02	96.99
0.0025*	**88.68**	98.00	97.96
2.50E-04	87.71	96.10	95.87
2.50E-05	75.81	93.71	89.80

Table 1. **Instance segmentation AP** on the PRISM dataset. The asterisk marks the baseline model. The highest AP for each experiment is bolded.

5.3 Qualitative Results

Visualizing the outputs of any model is important to obtain more nuanced insights into its performance, insights that are lost in an aggregated metric like the AP. This section focuses on results from the best performing model: the Mask R-CNN with the ResNeXt-101+FPN backbone, a learning rate of 0.0025, and a train/val/test split ratio of 70/15/15.

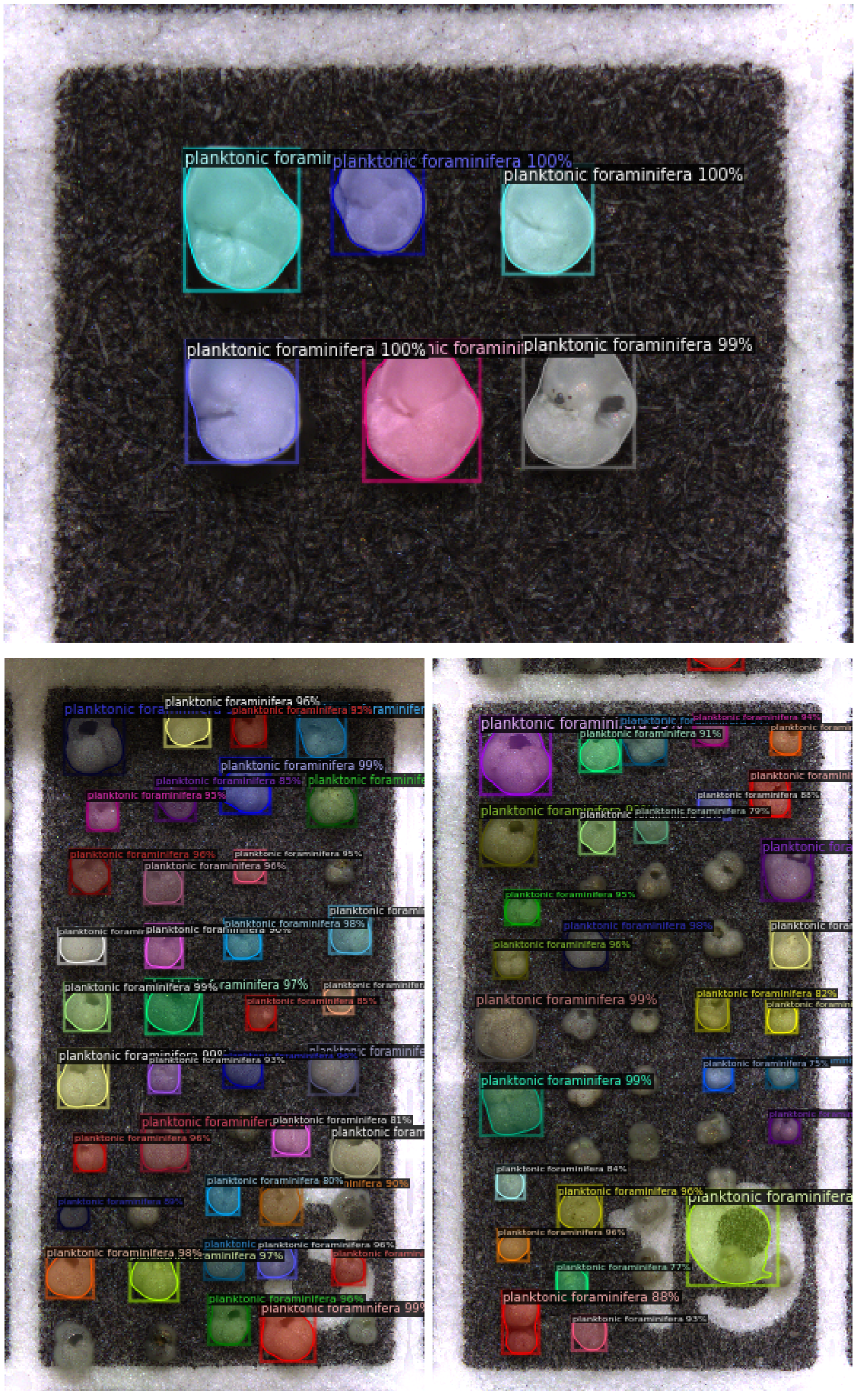

planktonic foramir
planktonic foraminifera 100%
planktonic foraminifera 100%
planktonic foraminifera 100%
ic foramini
planktonic foraminifera 99%
planktonic foraminifera 96%
planktonic foraminifera 99%
planktonic foraminifera 99%
planktonic foraminifera 99%
planktonic foraminifera 99%

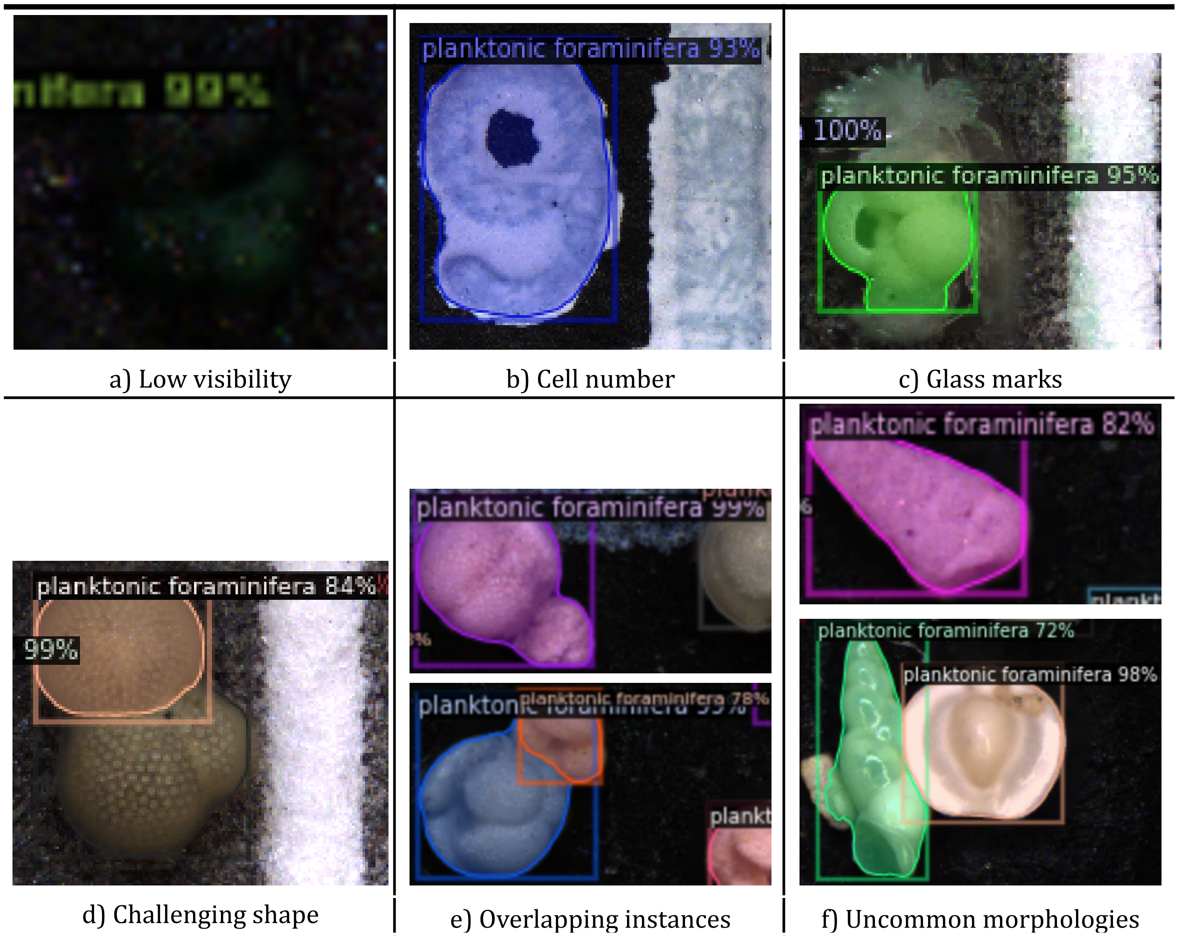
Figure 1a and 1b. **Example of model predictions**. Visualization of instance segmentation masks, object detection bounding boxes, and classification scores for a) a typical image (*left*), and b) an image with many more instances (*right*) from the test dataset.

For each predicted instance, the Mask R-CNN generates a segmentation mask, bounding box, and classification along with the confidence score. Figure 1 shows two examples of the output for a test image: a) an image similar to most other images in terms of image characteristics and model output, and b) an image with many more instances and relatively poor model performance.

Generally, the qualitative results reflect the high AP; the instance masks align very closely with the actual planktonic foraminifera outlines, and the model correctly classifies each instance as a foram with high scores close to 100%. However, Figure 1b shows an example of an image where the model fails to detect many instances.

While inspecting the visualizations of the model outputs, certain image characteristics were paid particular attention. These image characteristics represent cases that were hypothesized to pose a challenge to the model, and were difficult for traditional non-neural-net approaches to segment. Table 2 lists each case and an example(s) of how the model fails to perform as desired.

Model failure when handling challenging cases


nifera 99%
planktonic foraminifera 93%
100%
planktonic foraminifera 95%
planktonic foraminifera 82%
planktonic foraminifera 99%
planktonic foraminifera 84%
99%
planktonic foraminifera 78%
plankt
planktonic foraminifera 72%
planktonic foraminifera 98%

a) Low visibility

b) Cell number

c) Glass marks

d) Challenging shape

e) Overlapping instances

f) Uncommon morphologies

Figure 2. **Challenging cases and model failure**. Certain image characteristics of the PRISM dataset make the instance segmentation task harder, and the model performs less well on them. From left to right, top to bottom: a) low visibility: the model fails to detect an instance that suffers from poor contrast against the background; b) cell number: the model identifies a number on the slide as a foraminifera c) glass marks: the model segments the foraminifera poorly, because of scratches or other marks that occlude its boundaries; d) challenging shape: certain morphologies are tricky to segment correctly, as a subpart of it may be identified as a foraminifera itself; e) overlapping instances: overlapping or adjacent instance have less explicit boundaries, and may be segmented as a single instance, or otherwise incorrectly; f) uncommon morphologies: certain taxa and their

morphologies occur infrequently in the dataset, and may have lower mask quality and classification scores.

6 Discussion

6.1 Network Architecture

The baseline model performed extremely well. While state-of-the-art models typically yield an AP score of 40-50 on the COCO dataset, the default Mask R-CNN configuration consistently achieves a mask AP of near 90 on the PRISM dataset used in this paper. Furthermore, the training loss (Figure 2) and AP scores stabilize after ~100 iterations, showing that the model converges quickly. Therefore, not only can the model be trained over a shorter time, but also that it can be easily extended to new datasets in the future.

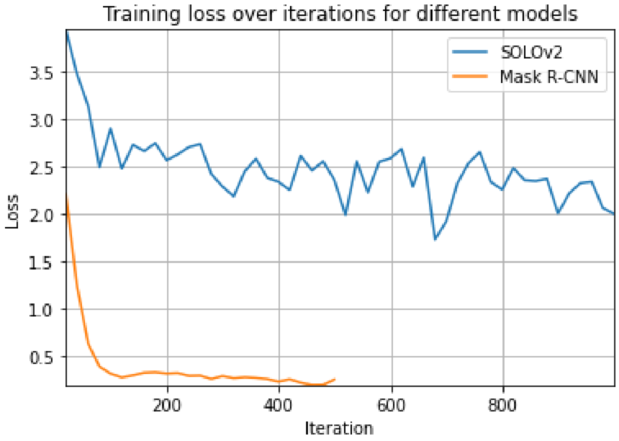


Figure 3. **Training loss** for Mask R-CNN baseline (*orange*) and SOLOv2 (*blue*) model architectures. The training process for SOLOv2 was extended to 1000 iterations to attempt to achieve convergence.

In contrast, SOLOv2 performed extremely poorly. The AP of 0.62 from using SOLOv2 was drastically lower than when using Mask R-CNN. Moreover, the model was hard to train; the loss often became undefined, and the training process often had to be monitored and restarted, even after attempting to adjust the learning rate. Training loss remained high even after increasing the number of iterations to 1000, and oscillated significantly throughout (Figure 2). However, the huge discrepancy in model performance cannot be conclusively attributed to whether the network uses a two-stage or single-stage instance segmentation approach. Model training is a time-consuming process, and deep networks have many hyperparameters that can contribute to model convergence. Given the satisfactory performance of the baseline model, we decided not to further optimize the learning process of SOLOv2 for the purpose of this project, and proceed with tuning the Mask R-CNN network.

6.2 Model Tuning

The choice of model backbone did not significantly affect the AP; all four yielded similarly good results. However, ResNet-50+DC5 performed noticeably worse, except when using a lower IOU threshold. The lower AP (~2 points) of ResNet-50+DC5 compared to ResNet-50+FPN highlights the importance of using a FPN to detect objects of different sizes. The AP scores for ResNet-50+FPN and

ResNet-101+FPN are too similar to warrant using a 101-layer ResNet over a 50-layer ResNet, as deeper models are more difficult to train and require more computational costs. On the other hand, the cardinality dimension of ResNeXt allows deeper ResNeXt models to be trained without increasing the number of parameters, as compared to their ResNet counterparts. Hence, ResNeXt-101+FPN was chosen as the final backbone architecture, as it has the highest AP, while still providing relatively fast training and inference.

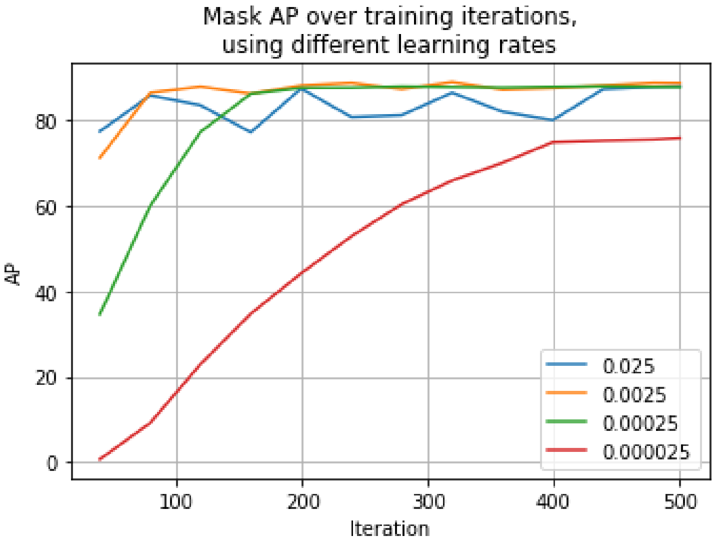


Figure 4. **Mask AP over training, using different learning rates**: 0.025 (blue), 0.0025 (orange), 2.50E-04 (green), 2.50E-05 (red). All four curves follow the same general trend. Using a learning rate of 0.0025 yields a smooth curve with fast convergence. AP oscillates at a suboptimal value when the learning rate is 0.025. The model takes significantly longer to converge when the learning rate is 2.50E-04 or 2.50E-05.

The learning rate controls the step size at each iteration of gradient descent. If the learning rate is too small, the model will need many more iterations to converge; if the learning rate is too large, the model may not converge on the global optimum. The results of adjusting the learning rate are as expected. When using the smallest learning rate (2.50E-05), the AP is significantly lower than with higher values, and the red training curve in Figure 3 confirms that this is because the model is learning, but very slowly. For the other three learning rates, although the final AP scores are similar, the training curves in Figure 3 show that the smaller learning rate (2.50E-04) takes noticeably more iterations for the AP to reach near 90, while the larger learning rate (0.025) results in unstable learning. The default value of 0.0025 achieves a balance between fast learning and model convergence, and hence is the best choice of learning rate.

6.3 Dataset Preparation

As the size of the training dataset grew, the AP showed an approximately increasing trend. Sampling 90% of the training images improved the AP by ~3 percentage points, compared to sampling 10% of images. However, as the sample ratio increases past 0.3, the metrics show much less change over each increment, and it is not clear that any increase in performance is statistically significant. Conducting a significance test given the number of runs and variance of the metrics is outside the scope of this project.

The fact that the metrics seem to plateau for the larger sample ratios still lead to two important implications. Firstly, the results indicate that the amount of training data generated was sufficient for our model to achieve optimal performance. Secondly, even though there is no clear cut-off for the minimum number of training images needed, it is likely that the size of the dataset could have been greatly reduced without any significant drop in model performance. Thus, a recommended approach for practitioners generating their own training dataset is to alternate between data annotation and model training: to systematically grow the training dataset until model performance is satisfactory or no longer improves significantly. This iterative process allows practitioners to have a better idea of how much data they need, potentially saving time and labor.

6.4 Qualitative Results

Although the AP score is high for almost all the models, the model's shortcomings when handling the difficult image cases (Figure 2) suggest that the model may not generalize well to more challenging datasets. Future work will look into data augmentation techniques to improve performance on a wider variety of image characteristics. In addition, the results suggest that the distribution of the data may be more important than the size of the training dataset, in order to train a model that is more robust to challenging or unseen cases.

7 Conclusion

Motivated by the limitations of non-neural network approaches, this project looked into training an instance segmentation model on microscope slide images of planktonic foraminifera. Recent computer vision models have adopted various strategies to perform instance segmentation, and can broadly be categorized into two-stage or single-stage procedures. For this paper, we initially compared using Mask R-CNN and SOLOv2 models pre-trained on the MS COCO dataset. Training SOLOv2 was difficult and ultimately unsuccessful (~0.62 AP). In contrast, the baseline Mask R-CNN model was very effective (~88.19 AP). Hence, the next step of the project focused on refining the Mask R-CNN model.

To optimize the performance of our Mask R-CNN model, we experimented with different backbone architectures, as well as learning rates. In terms of the network backbone, using a FPN is crucial to model performance (~2 points improvement), while the ResNext is designed to allow for deeper, more powerful models without raising computational cost, and indeed performs slightly better than when using a ResNet. The learning rate is one of the most important hyperparameters of any machine learning model, and a learning rate of 0.0025 was found to be an appropriate magnitude to attain both fast training and global convergence. Our final model thus uses a Mask R-CNN architecture with a ResNeXt-101+FPN backbone and a learning rate of 0.0025, and achieves a AP of 89.42. However, the qualitative results indicate that the high AP might not extend to future datasets, as the model still performs poorly when given an image with many instances or challenging image characteristics. Making the model less brittle to more diverse images is a focus of future work.

The question of how much data is needed is of interest to researchers looking to apply machine learning techniques on their work. This is because neural networks, especially deep networks, need to learn from a sufficient number of examples to perform well on unseen data, but the data annotation process is extremely resource-intensive. In this paper, we showed how our model's performance varies with the amount of training data, and propose an iterative process to carry out dataset preparation and model training in tandem.

References

1. Collins, M., R. Knutti, J. Arblaster, J.-L. Dufresne, T. Fichefet, P. Friedlingstein, X. Gao, W.J. Gutowski, T. Johns, G. Krinner, M. Shongwe, C. Tebaldi, A.J. Weaver and M. Wehner. (2013). Long-term Climate Change: Projections, Commitments and Irreversibility. *Climate Change 2013: The Physical Science Basis*. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change . Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
2. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. *2017 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2017.89
3. De la Vega, E., Chalk, T. B., Wilson, P. A., Bysani, R. P., & Foster, G. L. (2020). Atmospheric CO2 during the Mid-Piacenzian Warm Period and the M2 glaciation. *Scientific Reports, 10*(1). doi:10.1038/s41598-020-67154-8
4. Dowsett, H. J., Robinson, M. M., Stoll, D. K., Foley, K. M., Johnson, A. L., Williams, M., & Riesselman, C. R. (2013). The PRISM (Pliocene palaeoclimate) reconstruction: Time for a paradigm shift. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371*(2001), 20120524. doi:10.1098/rsta.2012.0524
5. Dowsett, H., Robinson, M., & Foley, K. (2015). A global planktic foraminifer census data set for the Pliocene ocean. *Scientific Data, 2*(1). doi:10.1038/sdata.2015.76
6. Ezard, T. H., Edgar, K. M., & Hull, P. M. (2015). Environmental and biological controls on size-specific δ13C and δ18O in recent planktonic foraminifera. *Paleoceanography, 30*(3), 151-173. doi:10.1002/2014pa002735
7. Facebook AI Research (FAIR). (2020, November 14). Detectron2: COCO Instance Segmentation Baselines with Mask R-CNN. Retrieved May 13, 2021, from https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md#coco-instance-segmentation-baselines-with-mask-r-cnn
8. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2017.322
9. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.90
10. Hsiang, A. Y., Brombacher, A., Rillo, M. C., Mleneck-Vautravers, M. J., Conn, S., Lordsmith, S., . . . Hull, P. M. (2019). Endless Forams: 34,000 Modern Planktonic Foraminiferal Images for Taxonomic Training and Automated Species Recognition Using Convolutional Neural Networks. *Paleoceanography and Paleoclimatology, 34*(7), 1157-1177. doi:10.1029/2019pa003612
11. Hsiang, A. Y., Nelson, K., Elder, L. E., Sibert, E. C., Kahanamoku, S. S., Burke, J. E., . . . Hull, P. M. (2017). AutoMorph: Accelerating morphometrics with automated 2D and 3D image

processing and shape extraction. *Methods in Ecology and Evolution, 9*(3), 605-612. doi:10.1111/2041-210x.12915

12. Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.106
13. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV 2014,* 740-755. doi:10.1007/978-3-319-10602-1_48
14. Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2017). Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. *ArXiv Preprint ArXiv:1701.04128*.
15. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137-1149. doi:10.1109/tpami.2016.2577031
16. U.S. Geological Survey. (2016, December 16). PRISM4: Pliocene Research, Interpretation and Synoptic Mapping. Retrieved May 13, 2021, from https://geology.er.usgs.gov/egpsc/prism/index.html
17. Wang, X. (2020, August 07). SOLO: Models. Retrieved May 13, 2021, from https://github.com/WXinlong/SOLO#models
18. Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C. (2020). SOLOv2: Dynamic and Fast Instance Segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*. Retrieved from https://arxiv.org/abs/2003.10152.
19. Wu, Y., Kirillov, A., Lo, W., & Girshick, R. (2019). Detectron2. Retrieved May 13, 2021, from https://github.com/facebookresearch/detectron2
20. Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.634