Improving Fine-Grain Segmentation via Interpretable Modifications: 
A Case Study in Fossil Segmentation

Indu Panigrahi, Ryan Manzuk, Adam Maloof, and Ruth Fong
Princeton University
{indup, rmanzuk, maloof, ruthfong}@princeton.edu

Abstract

Most interpretability research focuses on datasets containing thousands of images of commonplace objects. However, many high-impact datasets, such as those in medicine and the geosciences, contain fine-grain objects that require domain-expert knowledge to recognize and are time-consuming to collect and annotate. As a result, these datasets contain few annotated images, and current machine vision models cannot train intensively on them. Thus, adapting interpretability techniques to maximize the amount of information that models can learn from small, fine-grain datasets is an important endeavor.

Using a Mask R-CNN to segment ancient reef fossils in rock sample images, we present a general paradigm for identifying and mitigating model weaknesses. Specifically, we apply image perturbations to expose the Mask R-CNN’s inability to distinguish between different classes of fossils and its inconsistency in segmenting fossils with different textures. To address these shortcomings, we extend an existing model-editing method for correcting systematic mistakes in image classification to image segmentation and introduce a novel application of the technique: encouraging a greater separation between positive and negative pixels for a given class. Through extensive experiments, we find that editing the model by perturbing all pixels for a given class in one image is most effective (compared to using multiple images and/or fewer pixels). Our paradigm may also generalize to other segmentation models trained on small, fine-grain datasets.

1. Introduction

Today, most computer vision models are trained on large-scale datasets (e.g. ImageNet [30] and Microsoft COCO [18]) that contain thousands of annotated images of commonplace scenes and objects. This is in part because deep neural networks, the current state-of-the-art in machine learning, require large amounts of data in order to learn complex, highly predictive patterns that enable them to outperform classical machine learning methods. Consequently, most work in interpretability focuses on understanding networks trained on these large-scale datasets, typically for object classification. However, many high-impact domains, such as those in the natural and life sciences, involve fine-grain objects that require domain-expert knowledge to recognize and are time-consuming to collect and annotate [3, 31, 37]. As a result, these datasets contain few annotated images; however, deep neural networks cannot be sufficiently trained on small datasets to segment objects in images well (Tab. [1]). Furthermore, there is a lack of insight into how well interpretability techniques extend to fine-grain segmentation tasks trained on small datasets.

In this paper, we apply several interpretability techniques to expose and address the shortcomings of a Mask R-CNN model [14] when segmenting ancient reef fossils in a small set of annotated rock sample images. Each image depicts a cross section of a rock sample and contains pixels that represent red mud and the embedded remains of different types of fossils (Fig. [1]). We are interested in the pixels that
represent archaeocyathids (Fig. 1), an extinct reef-building sponge [29].

515 million years ago, archaeocyathids became the first animals to build reefs with complex structures similar to today’s reefs [8]. Around the same time, an event known as the Cambrian radiation occurred in which animals rapidly became much more diverse and complex than they had been before. Modern coral reefs are extremely important for biodiversity. They have been likened to tropical rainforests both in terms of the tree-like function of their branching skeletal structures and the nurturing habitats that they provide for other organisms [34]. Considering the overall importance of reefs to marine biodiversity, it has been hypothesized that the development of large reefs could have caused the shift in animal complexity [21]. Discovering the roles that ancient archaeocyathid reefs played in the development of complex animals, and in Earth’s history more generally, will inform our understanding of the impact that dwindling coral reefs will have on Earth’s future climate and biosphere [28].

Constraining the roles of past reefs involves understanding their three-dimensional branching structure. In many cases, as with our rock sample, embedded specimens cannot be physically isolated from the surrounding material. Thus, we instead generate 3D models of the specimens through serial sectioning and imaging of samples [23]. To build a model, we need to segment the pixels of archaeocyathids in each image and then stack the resulting masks (Fig. 2).

Due to the fine-grain appearance of archaeocyathid fossils and the domain knowledge needed to recognize them, manually segmenting each image is time-consuming. For reference, it took approximately two hours to annotate the single image shown in Fig. 1 (Tab. 1). The full image stack contains 3,454 images, so manually segmenting all the images would take an unreasonable amount of time. Furthermore, this image stack is only one of many such rock sample datasets, and each stack can have a vastly different visual appearance. Thus, we seek to automate the segmentation process by utilizing a Mask R-CNN model [14] to segment an image stack from a small amount of training data (10 annotated images: 6 training, 2 validation, and 2 test images).

|                       | MS COCO | Ours |
|-----------------------|---------|------|
| Number of labeled images | > 200K  | 10   |
| Time to annotate an image (hrs) | 0.66    | 2    |
| Domain expert knowledge needed | ✓       | ✓    |

Table 1: Comparison between COCO dataset [18] and ours.

Figure 2: Reef modeling process. From top to bottom: The archaeocyathids in each image in the stack are segmented, and the segmented portions are stacked to form a 3D model.

In this work, we utilize a small, geosciences dataset and the Mask R-CNN model to outline a general paradigm for identifying and mitigating shortcomings of our model that may be generally applicable to other segmentation models trained on small, fine-grain datasets. Our main contributions are summarized as follows:

- We identify model weaknesses via image perturbations and texture synthesis. Specifically, we find that our model is not robust to intraclass variation within the archaeocyathid class and cannot clearly distinguish between different classes (i.e. types) of fossils.
- We mitigate known model weaknesses by extending an existing model-editing technique [32] for correcting systematic mistakes in image classification to im-
age segmentation. Specifically, we demonstrate a novel application of the method for enforcing a pixel-level supercategorization of classes (i.e. encouraging the model to distinguish archaeocyathids from non-archaeocyathids).

- Through extensive experiments, we glean several insights on how to use the editing method: We show that a single editing operation using one image (vs. multiple images or editing operations) is sufficient and that editing using all relevant pixels (vs. a smaller subset) in an image is most impactful.

2. Related Work

Dataset. The rock sample dataset is obtained by alternately grinding and imaging cross sections of a rock sample\cite{23}. Each image in this rock sample contains several types of fossils. In this paper, we focus on the fossils of archaeocyathids and calcimicrobes for simplicity (Fig. 1).

We annotate the images by tracing an individual instance (polygon) for each archaeocyathid\cite{4}. Red mud is annotated as one conglomerate.

Data augmentation. Typically, data augmentation techniques are used to artificially enlarge computer vision datasets, such as applying geometric transformations to input images (e.g. rotation, scale, cropping, etc.)\cite{24} and the design and use of synthetic images\cite{6,5,36}. We use standard geometric transformations during training; however, synthetic images are less relevant to our task. Synthetic images are typically used to help models generalize to novel scenarios (e.g. improving autonomous driving on unseen roads); however, we are primarily concerned with improving segmentation on images from the same rock sample.

Connectomics. A similar problem occurs in the connectomics field when recovering the 3D structure of neurons from 2D brain scan images. The reconstruction process involves delineating boundaries around regions in the scans just as we segment archaeocyathids\cite{41}. However, most work in connectomics has been directed towards creating novel network architectures\cite{11,19,22} rather than using interpretability techniques to understand and mitigate model failures. Similar to approaches in connectomics, we modified a Mask R-CNN model to leverage the fact that the archaeocyathids remain in similar locations between layers in the stack by influencing the ranking of the proposal boxes generated by the Region Proposal Network. However, we found that the under-trained state of the model hinders its ability to take advantage of our modifications. Thus, we primarily focus our work on understanding and addressing the failures of the unmodified model in our specific task.

Image occlusion. A common interpretability technique involves occluding part of an input image and observing the resulting effect on a model’s output decision. Several works utilize image occlusions to generate attribution heatmaps that visualize the most important image regions for a model’s decision\cite{40,27,43,9,20}. Others partially occlude images during training as a data augmentation technique to improve model robustness\cite{33,7,10} and/or localization performance\cite{39}. Our work is more similar to those using occlusions to generate attribution heatmaps, as we selectively occlude all pixels from certain classes and observe the effect on the model to identify its shortcomings.

Texture synthesis. Texture synthesis is another class of methods that is often used to characterize a model and involve generating a realistic, synthetic texture\cite{12,15}. It can be used for inpainting a corrupted image\cite{16}, visualizing what kind of visual features most activates a channel in a network\cite{25}, or studying a network’s relative bias towards texture vs. shape\cite{13}. More similar to feature visualization, we generate textures for the different kinds of fossils in order to study how our model performs with respect to the visual appearance of archaeocyathids.

Model editing. There have been a number of methods proposed for editing a model after an original training period. From the machine learning fairness literature, several works have proposed to debias a model so that sensitive demographic information (e.g. race and gender) do not inform model predictions\cite{2,42,32,5,20,38}. However, not all model errors relate to a societal bias. One recent work by Santurkar et al.\cite{42} proposes editing an object classifier to correct for systematic mistakes, like misclassifying vehicles on snow. In this example, they map a synthesized snow texture underneath vehicles to a more typical asphalt road pattern such that the edited model classifies vehicles on snow as it classifies vehicles on asphalt. Their method can edit a model using a single image and a corresponding
perturbed version, so it can be adapted to models trained on small datasets.

Specifically, for a given layer, they refer to its input as keys and its output as values (Fig. 3). Then, they use an L1 loss function to tune the weights of the selected layer such that the keys for the image with the snow-covered road map to the values for the image with the paved road after the snow-covered road passes through the convolutional layer. We extend this model-editing method to image segmentation and experiment with several ways to improve segmentation of archaeocyathids.

3. Addressing Interclass Confusion

3.1. Identify model weakness

Archaeocyathid vs. non-archaeocyathid fossil confusion. For our dataset, the Mask R-CNN sometimes labels instances of another fossil called calcimicrobe (Fig. 1) along with a few other non-archaeocyathid fossils as archaeocyathids. To analyze this trend, we occlude all archaeocyathids from an image by inpainting them with a shade of red mud that we extract from a manually-selected red mud pixel. While the Mask R-CNN ideally should identify no archaeocyathids in the perturbed image, it instead identifies large portions of calcimicrobe as archaeocyathids (Fig. 4). Thus, the Mask R-CNN cannot clearly distinguish between archaeocyathids and calcimicrobes.

Archaeocyathid vs. red mud separability. As a complementary occlusion, we inpaint all non-archaeocyathid pixels with a shade of red mud (Fig. 7b) in our 6 training images and run inference. The quality of the instance masks for each image drastically improves (mean instance-level IoU across the 6 training images increases from 0.63 ± 0.29 to 0.78 ± 0.24, mean instance-level precision increases from 0.78 ± 0.22 to 0.89 ± 0.17, mean instance-level recall increases from 0.78 ± 0.25 to 0.86 ± 0.20) (Fig. 5). Thus, the model generally can distinguish between archaeocyathids and a simplified version of red mud.

Since we only need to isolate the archaeocyathid pixels, we have a binary segmentation task with archaeocyathids as positive pixels and non-archaeocyathids as negative ones. Thus, it would be ideal if the model associated all negative pixels with a concept it already recognizes, namely red mud.

3.2. Mitigate model weakness

Mapping non-archaeocyathids to red mud to reduce archaeocyathid vs. non-archaeocyathid confusion. To enforce this supercategorization, we apply the model-editing method [32] to one training image such that the model is encouraged to associate all non-archaeocyathid pixels with red mud (Fig. 6). Specifically, our k∗ (Fig. 3) is the input representation of the original, unperturbed image (Fig. 7a), and our v∗ is the output representation of the same image with all non-archaeocyathids pixels inpainted with red mud (Fig. 7b). We perform 20k rewriting steps at a learning rate of 10−4. Furthermore, we try tuning with each of the 6 training images individually.
that we perform the tuning at different resolutions and can consequently target objects of various sizes in the image.

**Experimental details.** We evaluate the mapping with instance-level metrics on our validation and test images (4 images). Specifically, we obtain the mean instance-level precision, recall, and IoU across all 4 images using a confidence threshold of 0. We use this threshold because we wish to evaluate whether or not the Mask R-CNN classifies a pixel as an archaeocyathid at all. Since an archaeocyathid sometimes has several, overlapping predicted masks (Fig. 8), we match each ground truth instance to the predicted instance mask with the highest IoU and take the mean across the matched predicted masks (i.e. the identified archaeocyathids) rather than across all predicted masks to avoid inflating our results.

Precision of archaeocyathid masks improves. We find that the precision of archaeocyathid masks improves significantly (Tab. 2); however, the tuned models generally produce masks with lower IoUs and mean recall scores. We also evaluate the mapping on the image used for tuning to verify that the mapping was successful and observe a similar trend except that the mean IoU improves for some images. The performance on the test images shows that the mapping does indeed reduce interclass confusion for the tuning image, even if it decreases the coverage of archaeocyathid pixels. For our application, this result demonstrates an improvement because the false identification of extraneous fossils as archaeocyathids interferes with measurements on the rendered 3D model. The choice of training image does seem to impact the performance; for example, tuning with $C$ improves the precision more than tuning with $E$ (Tab. 3). Additionally, we try mapping different amounts of non-archaeocyathid pixels to red mud; we use image $C$ for this experiment because it produces the best tuned model. We find that mapping more non-archaeocyathid pixels to red mud produces better masks (Tab. 3). This trend suggests that mapping all the non-archaeocyathid pixels at once is more effective than mapping a small portion.

**3.3. Tuning with Multiple Images**

In addition to tuning with different images individually, we test the effect of tuning with more than one image.

**Experimental details.** We experiment with five additional mappings, each of which incorporates a new image for tuning. We again use a learning rate of $10^{-4}$ and perform $20k$ rewriting steps for each image. For example, $A, B$ uses image $A$ for $20k$ steps with $lr = 10^{-4}$ followed by image $B$ for an additional $20k$ steps at the same learning rate. We add images in order of increasing percentage of archaeocyathid pixels, so image $A$ contains the lowest percent...
of archaeocyathid pixels and image E contains the highest percent of archaeocyathid pixels. Furthermore, we test sequences in increasing and decreasing order of precision, recall, and IoU (without incrementally adding images).

### 4. Addressing Intraclass Inconsistencies

#### 4.1. Identify model weakness

**Archaeocythids can have different textures.** There exists a fair amount of intraclass variation among archaeocythids. For example, there are recrystallized (white/gray) and red mud filled (red/brown) archaeocythids, irregular (long) and regular (round) archaeocythids, and more. We use labels (a) and (b) for the two primary textures (Fig. [1]). The Mask R-CNN segments (a) (archaeocythids with pitted textures; mean precision = 0.84; mean recall = 0.67) better than it segments (b) (archaeocythids filled with red mud; mean precision = 0.56; mean recall = 0.25) (Fig. [2]).

| Image | Precision | Recall | IoU |
|-------|-----------|--------|-----|
| None  | 0.86 ± 0.17 | 0.75 ± 0.26 | 0.63 ± 0.28 |
| A     | 0.90 ± 0.13  | 0.59 ± 0.26 | 0.52 ± 0.27 |
| B     | 0.88 ± 0.17  | 0.59 ± 0.25 | 0.52 ± 0.27 |
| C     | 0.91 ± 0.12  | 0.63 ± 0.26 | 0.56 ± 0.27 |
| D     | 0.89 ± 0.15  | 0.64 ± 0.25 | 0.56 ± 0.27 |
| E     | 0.87 ± 0.17  | 0.64 ± 0.26 | 0.56 ± 0.28 |
| F     | 0.90 ± 0.15  | 0.65 ± 0.26 | 0.57 ± 0.27 |

Table 2: Mapping non-archaeocythids to red mud. Metrics computed on 4 test images (mean and standard deviation reported) when tuning on each of the training images. The “Image” column denotes the training image that was used for tuning. The top row indicates the original model’s performance on the test images (no tuning). Precision improves when tuning on any training image, while recall and IoU decrease.

#### Tuning on one image is sufficient.

The performance of the model tuned on a combination loosely corresponds to the performance of the model tuned under the last image in the combination. For example, the performance of the model tuned on A, B, C is identical to that of the model tuned on just C (Tabs. [2] and [4]). More generally, the mean instance-level IoU and recall are almost identical to those under the model tuned with the last image in the combination. In addition, the mean instance-level precision is similar to that under the model tuned with the last image in the combination. One exception is the model tuned on A, B which performs worse overall. Thus, we find that tuning the model with one inpainted image is sufficient.

| % Pixels | Precision | Recall | IoU |
|----------|-----------|--------|-----|
| None     | 0.86 ± 0.17 | 0.75 ± 0.26 | 0.63 ± 0.28 |
| 1        | 0.86 ± 0.16  | 0.75 ± 0.26 | 0.63 ± 0.28 |
| 35       | 0.90 ± 0.14  | 0.65 ± 0.29 | 0.57 ± 0.31 |
| 100      | 0.91 ± 0.12  | 0.63 ± 0.26 | 0.56 ± 0.27 |

Table 3: Mapping different amounts of non-archaeocythid pixels to red mud. Metrics computed on 4 test images (mean and standard deviation reported). “% Pixels” indicates the percent of non-archaeocythid pixels in image C that were inpainted with red mud. The first row shows performance of the original model (no tuning). The second row is when pixels for one calcimicrobe are replaced. The third row is when non-archaeocythid pixels on the rock face (i.e. excluding the sides of the rock and the platform on which the rock sits) are replaced. The last row is when all non-archaeocythid pixels are replaced. When more non-archaeocythid pixels are replaced, the precision of the archaeocythid masks improves.

| Sequence | Precision | Recall | IoU |
|----------|-----------|--------|-----|
| None     | 0.86 ± 0.17 | 0.75 ± 0.26 | 0.63 ± 0.28 |
| A.B      | 0.85 ± 0.24  | 0.39 ± 0.25 | 0.35 ± 0.24 |
| A.B.C    | 0.91 ± 0.12  | 0.63 ± 0.26 | 0.56 ± 0.27 |
| A,B.C,D   | 0.90 ± 0.14  | 0.64 ± 0.25 | 0.56 ± 0.27 |
| A,B,C,D,F | 0.89 ± 0.16  | 0.64 ± 0.26 | 0.56 ± 0.28 |
| A,B,C,D,F,E | 0.87 ± 0.18  | 0.67 ± 0.25 | 0.56 ± 0.28 |
| E,B,D,A,F,C | 0.91 ± 0.13  | 0.62 ± 0.27 | 0.55 ± 0.28 |
| C,F,A,D,B,E | 0.88 ± 0.15  | 0.65 ± 0.26 | 0.57 ± 0.27 |
| A,B,C,E,D,F | 0.90 ± 0.14  | 0.63 ± 0.26 | 0.57 ± 0.27 |
| F,D,E,C,B,A | 0.89 ± 0.15  | 0.59 ± 0.26 | 0.51 ± 0.27 |
| B,A,E,D,C,F | 0.89 ± 0.15  | 0.64 ± 0.25 | 0.57 ± 0.27 |
| F,C,D,E,A,B | 0.87 ± 0.16  | 0.62 ± 0.25 | 0.52 ± 0.27 |

Table 4: Mapping non-archaeocythids to red mud using multiple, sequential tuning images. Metrics computed on 4 test images (mean and standard deviation reported). The “Sequence” column denotes the sequence of training images used for each tuning. The first set of 6 sequences corresponds to adding one image at a time. The next set of 2 sequences is in order of increasing and decreasing precision when using a single image (Tab. [2]). The next two sets are in order of increasing and decreasing recall and IoU respectively. Most combinations are comparable to tuning on the last image only (Tab. [2]) and generally improve on precision while decreasing recall and IoU.

**Test effect of optimal texture.** To test the effect of the pitted texture on the segmentation quality, we stitch copies of a square crop of the texture from one such pitted-textured archaeocythid by copying and pasting copies of the crop...
We tune with each of the 6 training images individually and characteristics of the archaeocyathid class (i.e. we apply supercategorization, we use the method to strengthen the pitted texture equivalent. Unlike our previous approach of replacing the texture in for together to form a continuous textured image of the same size as our images. We then substitute the texture in for all the archaeocyathids in the training images (Fig. 7c) and run inference on the modified images. The quality of the masks improves (mean IoU increases from $0.63 \pm 0.29$ to $0.66 \pm 0.31$) though to a lesser extent than the predicted masks for the unperturbed non-archaeocyathids. Further analysis shows that there is an increase both in instance-level precision (mean increases from $0.78 \pm 0.22$ to $0.80 \pm 0.17$) and recall (mean increases from $0.77 \pm 0.25$ to $0.87 \pm 0.23$). Qualitatively, we find that the segmentation of previously poorly-segmented archaeocyathids improves (Fig. 10).

4.2. Mitigate model weakness

Mapping archaeocyathids to the optimal texture to improve segmentation. Since the modified images seem to solicit an improved segmentation, we apply the model-editing method to map poorly-performing archaeocyathids to the pitted texture equivalent. Unlike our previous application of the model-editing method to enforce a binary supercategorization, we use the method to strengthen the characteristics of the archaeocyathid class (i.e. we apply the method for the same reason as the original work [32]). We tune with each of the 6 training images individually and

![Figure 9: Inconsistent segmentation of archaeocyathids. Examples of output archaeocyathids masks with varying segmentation quality. The Mask R-CNN fails to segment the irregular, red mud filled (b) archaeocyathid but produces a complete mask for the regular, recrystallized (a) archaeocyathid that contains a pitted texture.](image)

![Figure 10: Example of improvement due to texture replacement. An example of a red mud filled archaeocyathid which the model misses in the unperturbed image (left) and perfectly segments when replaced with a pitted texture (right). A similar trend occurs for other red mud filled archaeocyathids.](image)

| Image | Precision | Recall | IoU  |
|-------|-----------|--------|------|
| None  | 0.86 ± 0.17 | 0.75 ± 0.26 | 0.63 ± 0.28 |
| A     | 0.85 ± 0.16 | 0.70 ± 0.30 | 0.56 ± 0.32 |
| B     | 0.84 ± 0.16 | 0.70 ± 0.29 | 0.55 ± 0.32 |
| C     | 0.85 ± 0.18 | 0.73 ± 0.29 | 0.59 ± 0.31 |
| D     | 0.84 ± 0.15 | 0.72 ± 0.28 | 0.55 ± 0.32 |
| E     | 0.80 ± 0.20 | 0.69 ± 0.32 | 0.52 ± 0.34 |
| F     | 0.85 ± 0.15 | 0.70 ± 0.30 | 0.57 ± 0.32 |

Table 5: Mapping all archaeocyathids to pitted texture. Metrics computed on 4 test images (mean and standard deviation reported) when tuning on each of the training images. The top row shows the original model’s performance (no tuning). None of the tunings produce an improvement over the original model.

| % Pixels | Precision | Recall | IoU  |
|---------|-----------|--------|------|
| None    | 0.86 ± 0.17 | 0.75 ± 0.26 | 0.63 ± 0.28 |
| 6       | 0.86 ± 0.16 | 0.75 ± 0.26 | 0.62 ± 0.29 |
| 49      | 0.85 ± 0.16 | 0.74 ± 0.26 | 0.61 ± 0.29 |
| 100     | 0.84 ± 0.15 | 0.72 ± 0.28 | 0.55 ± 0.32 |

Table 6: Mapping different amounts of archaeocyathid pixels to pitted texture. Metrics computed on 4 test images (mean and standard deviation reported). “% Pixels” indicates the percent of archaeocyathid pixels in image $D$ that were replaced with the pitted texture. The first row shows the original model’s performance (no tuning). The second row is when pixels for one, and perfectly segments when replaced with the pitted texture. The third row is when pixels for 18 archaeocyathids are replaced. The last row is when all archaeocyathid pixels are replaced. The performance for 6% of replaced pixels is nearly identical to that under the original model; none of the tunings show an improvement over the original model.

produce two additional tuned models for image $D$ to test the effect of replacing different amounts of archaeocyathid pixels with the pitted texture.

Results Although mapping all the archaeocyathids to the pitted texture at once sometimes improves the segmentation of the tuning image itself, it generally produces masks with lower IoUs and does not improve the precision or recall for unseen images (Tab. 4). Furthermore, mapping fewer archaeocyathid pixels to the pitted texture does not significantly change the performance from the original model (Tab. 4). Thus, mapping the archaeocyathids to the pitted texture does not seem to be an effective approach. This may be because archaeocyathids filled with red mud are visually confusing (i.e. they resemble red mud rather than archaeocyathids with the pitted texture).
5. Combinations of Mappings

5.1. Simultaneous Mapping

When we run inference on perturbed training images where both non-archaeocyathids are inpainted with red mud and archaeocyathids are replaced with the pitted texture, the segmentation improves (mean IoU improves from 0.63 ± 0.29 to 0.71 ± 0.30, mean precision improves from 0.79 ± 0.22 to 0.85 ± 0.17, and mean recall improves from 0.77 ± 0.25 to 0.88 ± 0.21).

However, when we tune (20k steps; \( lr = 10^{-4} \)) an unperturbed image to a version where both the non-archaeocyathids are inpainted with red mud and the archaeocyathids are inpainted with the pitted texture (Fig. 7c), the tuned model produces lower quality masks for both the tuning image and the unseen images (Tab. 7). Thus, this mapping is not an effective approach.

![Table 7: Mapping simultaneously. Mean and standard deviation are computed on 4 test images when tuning on each of the training images. The top row indicates the original model’s performance on the test images. None of the tunings produce an improvement over the original model.](image)

5.2. Sequential Mapping

In addition to mapping to both perturbations simultaneously, we try mapping the non-archaeocyathids to red mud and then mapping the archaeocyathids to the pitted texture. This procedure is identical to tuning on multiple images for the non-archaeocyathids to red mud edit (Sec. 3) except we tune (20k steps; \( lr = 10^{-4} \)) to an image with the non-archaeocyathids inpainted followed by an image with the archaeocyathids replaced (and vice versa). We perform this mapping only with image \( C \) since it produces the most improvement in the non-archaeocyathids to red mud tuning. We find that the performance of each, sequentially tuned model corresponds to the performance of the last mapping in isolation. For example, the model tuned with the non-archaeocyathid mapping followed by the archaeocyathid mapping produces masks of similar quality to the model tuned with the archaeocyathid mapping only (Tab. 8). This trend seems reasonable given the results from earlier multiple tuning images experiment.

![Table 8: Mapping sequentially. Metrics computed on 4 test images (mean and standard deviation reported); all tunings were done using only image \( C \). \( Ar, No \) represents mapping archaeocyathids to pitted texture followed by mapping non-archaeocyathids to red mud; \( No, Ar \) represents the reverse sequence. The performance of each sequential mapping is similar to the performance when tuning with the last mapping only (i.e. \( Ar, No \) is similar to \( No \).](image)

6. Conclusion

In this work, we use the Mask R-CNN model and a small, fine-grain geosciences dataset as a case study that may be generally applicable for understanding a segmentation model’s behavior and improving its performance on any small, fine-grain dataset.

First, we show how inpainting and texture synthesis can identify model weaknesses such as interclass confusion (e.g. our model confused a different type of fossil, calcimicrobes, for archaeocyathids) or intraclass inconsistencies (e.g. performance varied for archaeocyathids with different textures). Second, we extend a model-editing technique \([32]\) for image classification to image segmentation and show how to best apply it to mitigate identified model weaknesses. We show that one tuning image is sufficient, that mapping all relevant pixels is more effective than mapping smaller portions of relevant pixels, and that sequentially performing tuning operations typically yields the same performance as tuning using the last operation alone.

We also demonstrate that the tuning technique may not work in challenging circumstances when the visual appearance of an object or texture is very similar to another, non-relevant object/texture (e.g. archaeocyathids filled with red mud look similar to red mud, and mapping them to pitted-textured archaeocyathids does not work). Lastly, we demonstrate that the tuning technique can improve precision when designed to mitigate interclass confusion (e.g. treating non-archaeocyathid pixels as red mud); however, it can negatively impact IoU and recall.

Limitations. Given that our work focuses on understanding and editing a Mask R-CNN trained on a single, small dataset of rock samples, the main limitation is that these findings may not generalize well to other segmentation models trained on small, fine-grain datasets. To the best of our abilities, we ran extensive experiments to substantiate our findings, but work in novel domains should validate our findings on their own models and datasets.
References

[1] Fluid Annotation: An Exploratory Machine Learning-Powered Interface for Faster Image Annotation. https://ai.googleblog.com/2018/10/fluid-annotation-exploratory-machine.html. Accessed: 2022-06-24.

[2] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. arXiv, 2017.

[3] Laura Brenskelle, Rob P Guralnick, Michael Denslow, and Brian J Stucky. Maximizing human effort for analyzing scientific images: A case study using digitized herbarium sheets. Applications in plant sciences, 8(6):e11370–e11370, July 2020.

[4] Justin Brooks. COCO Annotator. https://github.com/jsbroks/coco-annotator/ 2019.

[5] Tom Bu, Xinhe Zhang, Christoph Mertz, and John M. Dolan. Carla simulated data for rare road object detection. In Proc. IEEE International Intelligent Transportation Systems Conference (ITSC), pages 2794 – 2801, September 2021.

[6] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In CVPR.

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv, 2017.

[8] Douglas H Erwin and Sarah Tweedt. Ecological drivers of the Ediacaran-Cambrian diversification of Metazoa. Evolutionary Ecology, 26(2):417–433, 2012.

[9] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In ICCV, 2017.

[10] Ruth Fong and Andrea Vedaldi. Occlusions for effective data augmentation in image classification. In ICCV Workshop, 2019.

[11] Jan Funke, Fabian Tschopp, William Grisaitis, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Serge Belongie. Feature Pyramid Networks for Object Detection, 2016.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft COCO: Common Objects in Context. In ECCV, 2014.

[13] Xuejing Lei, Ganning Zhao, and C. C. Jay Kuo. Nites: A non-parametric interpretable texture synthesis method. arXiv, 2020.

[14] Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. Deep image prior. In CVPR, 2018.
[33] K.K. Singh and Y.J. Lee. Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization. In *ICCV*, 2017.

[34] Peter K. Swart. Coral Reefs: Canaries of the Sea, Rainforests of the Oceans. *Nature Education Knowledge*, 4(3):5, 2013.

[35] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *arXiv*, 2018.

[36] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *CVPR*, 2019.

[37] Shanshan Wang, Cheng Li, Rongpin Wang, Zaiyi Liu, Meiyun Wang, Hongna Tan, Yaping Wu, Xinfeng Liu, Hui Sun, Rui Yang, Xin Liu, Jie Chen, Huihui Zhou, Ismail Ben Ayed, and Hairong Zheng. Annotation-efficient deep learning for automatic medical image segmentation. *Nature Communications*, 12(1):5915, 2021.

[38] Tianlu Wang, Jiuy Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. *arXiv*, 2018.

[39] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017.

[40] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[41] Tao Zeng, Bian Wu, and Shuiwang Ji. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics*, 33(16):2555–2562, aug 2017.

[42] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AEIS*, 2018.

[43] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object Detectors Emerge in Deep Scene CNNs. *ICLR*, 2015.
Appendix

This appendix contains tables for metrics on the images used for tuning. We include these evaluations to check if the tuned models perform better than the original model on the image used for the corresponding tuning. The evaluations on the test images are included in the main paper. In general, we see that the performance of each mapping on the tuning images loosely corresponds to its performance on the test images. For example, the simultaneous mapping does not perform well on most of the tuning images Tab. 11 or on the test images (Tab. 7 in main paper).

### Table 9: Mapping non-archaeocythids to red mud: Performance on tuning image.

This table contains the mean instance-level metrics ± one standard deviation run on each image used for tuning non-archaeocythids to red mud. For example, the first row contains the mean instance-level precision, recall, and IoU of the identified archaeocythids in image A only. All the precisions increase, and some of the IoU scores increase as well. The recall scores decrease.

| Image | Precision | Recall | IoU |
|-------|-----------|--------|-----|
|       | Original  | Tuned  | Original | Tuned  | Original | Tuned  |
| A     | 0.82 ± 0.19 | 0.88 ± 0.19 | 0.78 ± 0.24 | 0.63 ± 0.28 | 0.66 ± 0.25 | 0.57 ± 0.29 |
| B     | 0.53 ± 0.28 | 0.54 ± 0.27 | 0.78 ± 0.24 | 0.67 ± 0.24 | 0.66 ± 0.27 | 0.61 ± 0.26 |
| C     | 0.83 ± 0.20 | 0.95 ± 0.09 | 0.77 ± 0.24 | 0.70 ± 0.23 | 0.64 ± 0.30 | 0.67 ± 0.24 |
| D     | 0.79 ± 0.23 | 0.91 ± 0.13 | 0.78 ± 0.28 | 0.74 ± 0.23 | 0.64 ± 0.30 | 0.66 ± 0.26 |
| E     | 0.85 ± 0.15 | 0.92 ± 0.10 | 0.70 ± 0.26 | 0.63 ± 0.28 | 0.55 ± 0.32 | 0.58 ± 0.29 |
| F     | 0.83 ± 0.16 | 0.90 ± 0.10 | 0.82 ± 0.22 | 0.72 ± 0.21 | 0.67 ± 0.26 | 0.64 ± 0.23 |

### Table 10: Mapping all archaeocythids to pitted texture: Performance on tuning image.

This table contains the mean instance-level metrics ± one standard deviation run on each image used for tuning archaeocythids to the pitted texture. For example, the first row contains the mean instance-level precision, recall, and IoU of the identified archaeocythids in image A only. There does not appear to be a clear trend in any of the metrics. For example, image D produces an improvement across all metrics, while image F does not.

| Image | Precision | Recall | IoU |
|-------|-----------|--------|-----|
|       | Original  | Tuned  | Original | Tuned  | Original | Tuned  |
| A     | 0.82 ± 0.19 | 0.82 ± 0.18 | 0.78 ± 0.24 | 0.81 ± 0.22 | 0.66 ± 0.25 | 0.67 ± 0.26 |
| B     | 0.53 ± 0.28 | 0.53 ± 0.25 | 0.78 ± 0.24 | 0.77 ± 0.24 | 0.66 ± 0.27 | 0.62 ± 0.29 |
| C     | 0.83 ± 0.20 | 0.84 ± 0.15 | 0.77 ± 0.24 | 0.75 ± 0.30 | 0.64 ± 0.30 | 0.60 ± 0.34 |
| D     | 0.79 ± 0.23 | 0.85 ± 0.16 | 0.78 ± 0.28 | 0.82 ± 0.26 | 0.64 ± 0.30 | 0.66 ± 0.28 |
| E     | 0.85 ± 0.15 | 0.80 ± 0.16 | 0.70 ± 0.26 | 0.73 ± 0.30 | 0.55 ± 0.32 | 0.48 ± 0.35 |
| F     | 0.83 ± 0.16 | 0.82 ± 0.14 | 0.82 ± 0.22 | 0.81 ± 0.26 | 0.67 ± 0.26 | 0.65 ± 0.28 |

### Table 11: Mapping simultaneously: Performance on tuning image.

This table contains the mean instance-level metrics ± one standard deviation run on each image used for tuning simultaneously. For example, the first row contains the mean instance-level precision, recall, and IoU of the identified archaeocythids in image A only. Only image F produces an increase in precision. None of the tuned models have higher recall or IoU scores than the original model.