Improving Data-Efficient Fossil Segmentation via Model Editing

Anonymous L3D-IVU submission

Paper ID 18

Abstract

Most computer vision 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 labeled images, and current machine vision models cannot train intensively on them. Originally introduced to correct large-language models, model-editing techniques in machine learning have been shown to improve model performance using only small amounts of data and additional training. Using a Mask R-CNN to segment ancient reef fossils in rock sample images, we present a two-part paradigm to improve fossil segmentation with few labeled images: we first identify model weaknesses using image perturbations and then mitigate those weaknesses using model editing.

Specifically, we apply domain-informed 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 with no additional labeled data needed and show its effectiveness in decreasing confusion between different kinds of fossils. We also highlight the best settings for model editing in our situation: making a single edit using all relevant pixels in one image (vs. using multiple images, multiple edits, or fewer pixels). Though we focus on fossil segmentation, our approach may be useful in other similar fine-grain segmentation problems where data is limited.

1. Introduction

Today, most computer vision models are trained on large-scale datasets (e.g. ImageNet [29] and Microsoft COCO [16]) 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. 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, 30, 35]. As a result, these datasets contain few labeled images. However, deep neural networks often cannot be sufficiently trained on small datasets to segment objects in images well (Tab. 1).

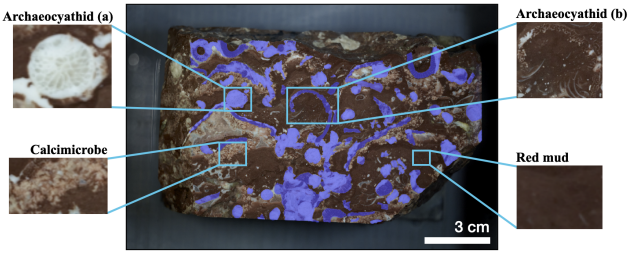


Figure 1. **Overview of rock sample.** Sample annotated image from our rock sample dataset with magnified examples of archaeocyathids, a calcimicrobe, and red mud. There also exist other classes that are omitted from this paper for simplicity. We assign labels **(a)** and **(b)** to two primary textures of archaeocyathids; **(a)** has a discernible pitted texture, while **(b)** is filled with red mud and consequently blends in with the surrounding mud. Archaeocyathid pixels in the full image are colored in purple with 80% transparency; magnified examples show the original coloring.

Model-editing methods have recently emerged in natural language processing (NLP) [22, 23, 43] and now computer vision [12, 31, 33] as a way to correct for systematic mistakes in deep learning models. These techniques differ from related works in domain adaptation and continual learning because of their focus on correcting mistakes in models (as opposed to adapting to a domain shift in inputs and/or retaining memory of the original distribution) by using limited data and additional training. This combination of limited data and focus on correcting errors makes model editing particularly promising in domains like geosciences, where data is hard to come by yet a small amount of domain

L3D-IVU
#18

expertise can be accessed to correct and improve a model.

In this paper, we present a two-part framework for improving a fossil segmentation model. First, we use domain-informed image perturbations to **identify model weaknesses**. Second, we adapt a model-editing method to **mitigate those weaknesses**. We focus our work on improving a Mask R-CNN [11] trained on a small set of annotated rock sample images to segment ancient reef fossils (Fig. 1).

We are interested in segmenting *archaeocyathids* (Fig. 1), an extinct reef-building sponge [28]. Studying these ancient reef fossils would allow us to understand their influence on past oceanic biodiversity [19] and consequently inform our understanding of the impact that dwindling coral reefs today will have on Earth's future climate and biosphere [27]. In many cases, as with our rock sample, embedded specimens are too delicate to be physically isolated from the surrounding material. One solution is to generate 3D models of the specimens from serial sectioning and imaging of samples [21]. To build such models, 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 archaeocyathids and the domain knowledge needed to recognize them, manually segmenting each image is time-consuming, with one image taking around 2 hours to annotate (Tab. 1). Given that a single, full image stack contains over 3,000 images, manually segmenting all images is very labor intensive. Furthermore, there exist many rock samples archived in museums and universities, and the image stacks generated for each can vary significantly in terms of visual appearance. Thus, we seek to automate the segmentation process by utilizing the Mask R-CNN model [11] to segment an image stack from a limited amount of labeled data ($\approx$ 10 images). Because of this data constraint, simply fine-tuning our model did not produce high-quality segmentation masks that are needed when forming 3D models for such specimens.

	MS COCO	Ours
Number of labeled images	$>$ 200K	10
Time to annotate an image (hrs)	0.66 [1]	2
Domain expert knowledge needed		✓

Table 1. Comparison between COCO dataset [16] and ours.

Rather than annotating more images, we focused on leveraging a model-editing technique [31] to improve our baseline model. This technique requires no additional labelled data, making it particularly well-suited for specialized datasets like ours that require domain knowledge expertise and significant time to annotate.

In this work, we present a two-part, data-efficient paradigm that combines domain-informed image perturbations with a model-editing method to first **identify** and then **mitigate weaknesses** in our model. Our main contributions are summarized as follows:

- We first identify model weaknesses via image perturbations and texture synthesis. From these experiments, we found two main weaknesses. First, our model often confuses archaeocyathids with other visually-similar types of fossils (e.g. **interclass confusion**). Second, our model is not robust to the visual diversity that archaeocyathids can have (e.g. **intraclass variation**).
- We then mitigate the identified model weaknesses by extending and evaluating an existing, model-editing technique [31] for correcting systematic mistakes in image classification to image segmentation. In particular, we find that certain edits improve the model's ability to distinguish between archaeocyathids and other types of fossils.
- Lastly, we gain several insights on how to effectively use the editing method. We show that performing a single edit using one image (vs. using multiple images or multiple, sequential edits) is sufficient and, further, that editing by using all relevant pixels (vs. a smaller subset of pixels) yields the best results.

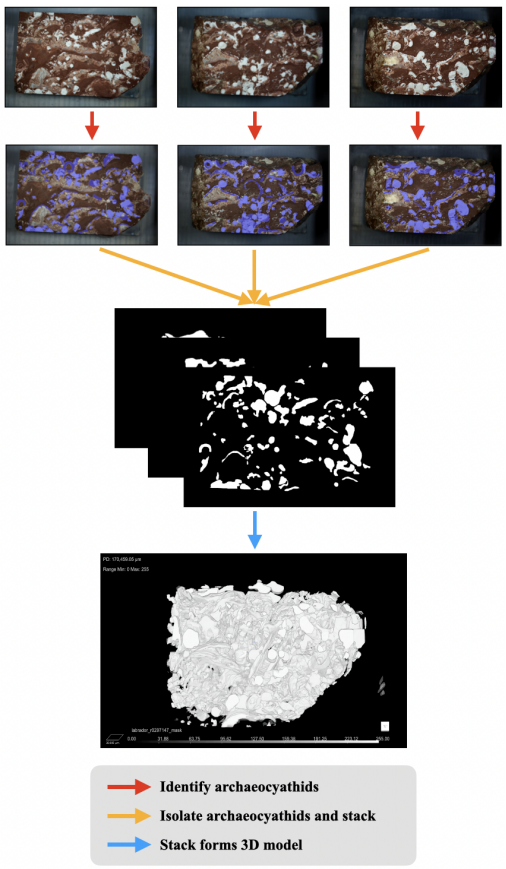


Figure 2. **Reef modeling process.** From top to bottom: The archaeocyathids in each image in the stack are segmented, and the segmented portions are stacked to form a 3D model.

2. Related Work

In this section, we first discuss related work in the field of *connectomics*, which tackles a similar problem of building a 3D structure from 2D data. Then, we discuss relevant literature for the two core parts of our work: identifying model weaknesses via image perturbations (e.g. *image occlusion* and *texture synthesis*) and mitigating weaknesses via *model editing*.

Connectomics. Connectomics tackles a similar problem to ours when learning 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 [40]. However, most work in connectomics has been directed towards creating novel network architectures [8, 17, 20] rather than using interpretability or model-editing techniques to understand and mitigate model failures. Similar to approaches in connectomics, we modified our Mask R-CNN to leverage similarities between neighboring images in the stack; however, we found that our model performed poorly despite this modification.

Image occlusion. Image occlusion 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 [6, 25, 26, 39, 42]. Others partially occlude images during training as a data augmentation technique to improve model robustness [5, 7, 32] and/or localization performance [37]. 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. However, we further use the perturbed images to edit our model.

Texture synthesis. Texture synthesis refers to methods that generate a synthetic, often realistic-looking texture [9, 13]. It can be used in a variety of ways: from inpainting a corrupted image [14], to visualizing what kind of visual features most activates a channel in a network (i.e. feature visualization) [24], to studying a network's relative bias towards texture vs. shape [10]. More similar to feature visualization, we generate certain textures in order to study how our model responds to the visual appearance of archaeocyathids.

Model editing. There have been a number of methods proposed for editing a model after a pretraining period in computer vision and NLP. From the machine learning fairness literature, several works have proposed to debias a model so that sensitive demographic information (e.g. race and gender) does not inform model predictions [2, 18, 34, 36, 41]. However, not all model errors relate to a societal bias.

In addition to correcting for model biases, model-editing techniques have been used to update the knowledge encoded in large language models to remove outdated information and/or to introduce new information [22, 23, 43].

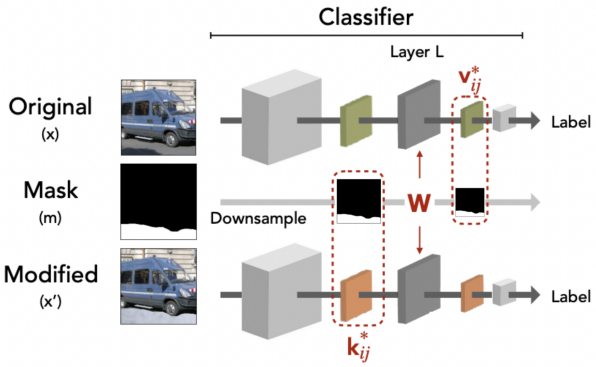


Figure 3. **Overview of model-editing method [31].** Layer L is the targeted convolutional layer, k^* is the input representation (keys) of the snow-covered road, and v^* is the output activation (values) of the paved road. The method maps k^* to v^* by tuning the weights w of the convolutional layer L. A mask m can be used to restrict the tuning to affect the road pixels only; we do not use a mask for any of our tunings. Figure adapted from [31].

Similarly, in computer vision, model-editing techniques have been introduced for correcting mistakes in image classification [12, 31, 33]. However, little work has been done towards editing image segmentation models.

One recent work by Santurkar et al. [31] 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 accurately 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 selected convolutional 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 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 layer. Using a rock sample dataset, we extend this method to image segmentation by first applying domain-informed image perturbations to identify systematic mistakes that the Mask R-CNN makes and then editing the model to correct those mistakes.

3. Experimental Setup

Dataset. We use images of the Labrador rock sample from [19] which were shared by their authors. The dataset was collected by alternately grinding and imaging cross sections of a rock sample, using a methodology similar to [21]. Each image depicts a cross section of the rock sample and contains pixels that represent red mud and the embedded remains of different types of fossils (Fig. 1). In this paper, we focus on archaeocyathid and calcimicrobe fossils (Fig. 1).

We annotate a total of 10 images by tracing an individual instance (polygon) for each archaeocyathid [4]. We split our 10 annotated images into the following subsets: 6 training, 2 validation, 2 test.

Model. We fine-tune a Mask R-CNN pretrained on ImageNet and COCO [38] on our 6 training images to perform instance segmentation for individual archaeocyathids. Similar to approaches in connectomics [17], we modified our model to leverage the fact that the archaeocyathids remain in similar locations between close layers in the stack by influencing the ranking of the proposal boxes generated by the Region Proposal Network in the Mask R-CNN. However, we found that the model still classified several non-archaeocyathid fossils as archaeocyathids and generally did not produce precise masks for archaeocyathids.

The precisions of the archaeocyathid masks are particularly important because the identification of spurious non-archaeocyathid pixels interferes with measurements on the rendered 3D model and consequently provides misleading information about the reef's structure. We would prefer that a few archaeocyathids are missed rather than being fully segmented in an instance containing non-archaeocyathid pixels. In other words, we prioritize precision over recall. Thus, in this work, we focus on leveraging model-editing, with a specific goal of improving the precision of segmentation masks.

4. Addressing Interclass Confusion

4.1. 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 classifies 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 drastically improves (mean instance-level IoU across all 443 archaeocyathids from 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.

4.2. Mitigate model weakness

Mapping non-archaeocyathids to red mud to reduce interclass confusion. To enforce this binary supercategorization, we apply the model-editing method [31] 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 image (Fig. 7a), and our v^* is the output representation of the same image with all non-archaeocyathid 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.

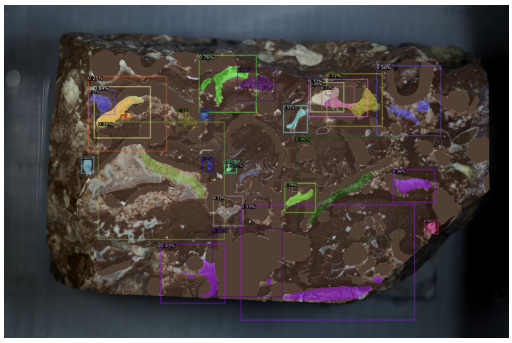


Figure 4. **Example of interclass confusion.** Resulting segmentation when archaeocyathids are inpainted with a shade of red mud. The Mask R-CNN misclassifies several instances of calcimicrobe (boxed and filled with various colors) as archaeocyathids.

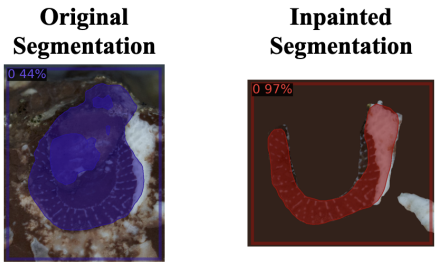


Figure 5. **Example of improved masks from inpainted image.** The mask of this archaeocyathid improves when the original image (left) is inpainted with red mud (right).

The model-editing method applies to feature maps, so we tune the weights of *each* of the 5, 3x3 output convolutional layers in the ResNet-101 FPN backbone [15] that produce the feature maps for the Mask R-CNN. Doing so means that we perform the tuning at different resolutions and can consequently target objects of various sizes in the image.

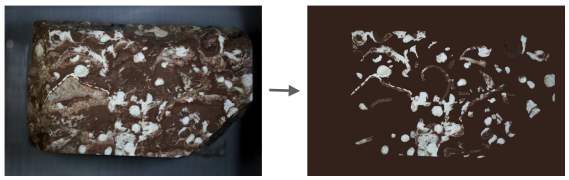
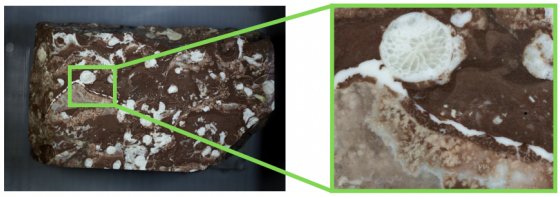
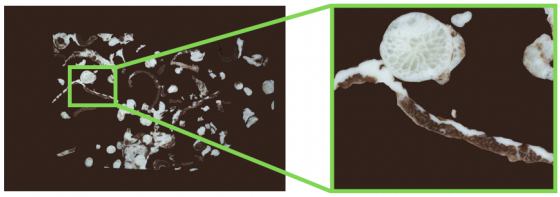


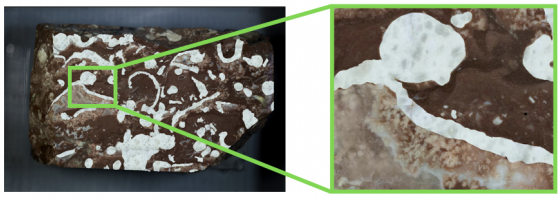
Figure 6. **Mapping non-archaeocyathids to red mud.** We apply the model-editing method to map the input representation of the unperturbed image (left) to the output representation of the inpainted image (right) to enforce a binary supercategorization of archaeocyathids and non-archaeocyathids.



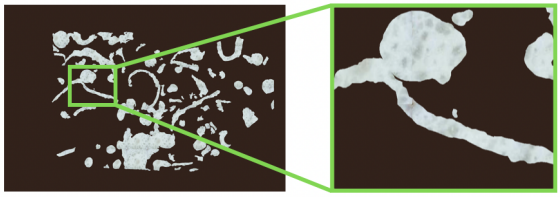
(a) Original image.



(b) Inpaint all non-archaeocyathids with a shade of red mud.



(c) Substitute pitted texture for all archaeocyathids.



(d) Inpaint (b) and (c) simultaneously.

Figure 7. **Image perturbations.** (a) shows the original training image with magnified archaeocyathids, (b) shows the version of the image with all non-archaeocyathid pixels inpainted with a manually-extracted solid shade of red mud, (c) shows the version with only the archaeocyathids replaced with a pitted texture, and (d) shows the version with (b) and (c) simultaneously.

Evaluation details. Since the validation set was not used when applying the model-editing method, we combine our validation and test sets to evaluate the mapping on more archaeocyathids. Specifically, we obtain the mean instance-level precision, recall, and IoU across all 215 archaeocyathids in the 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.

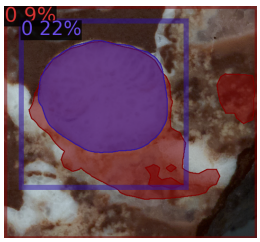


Figure 8. **Example of overlapping instance masks.** The original model predicts two instance masks (one colored in purple and another colored in red) for this archaeocyathid.

Precision of archaeocyathid masks improves. We find that the precisions of archaeocyathid masks improve significantly, while the IoU and recall scores decrease (Tab. 2). We also separately evaluate the mapping on each tuning image and observe a similar trend except that the mean IoU improves for some images. For our application, precision is the most important metric because the false identification of extraneous fossils as archaeocyathids interferes with measurements on the rendered 3D model. Thus, the mapping does indeed reduce interclass confusion, even if it decreases the coverage of archaeocyathid pixels.

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. 2). Additionally, we try mapping different amounts of non-archaeocyathid pixels to red mud, using image C because it produces the best tuned model. We find that mapping more non-archaeocyathid pixels to red mud produces more precise masks (Tab. 3). This trend suggests that mapping all the non-archaeocyathid pixels at once is more effective than mapping a small portion.

4.3. 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

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

Table 2. **Mapping non-archaeocyathids to red mud.** Metrics computed on 215 archaeocyathids from 4 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.

tunes on 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).

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 metrics are similar to those 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.

5. Addressing Intraclass Inconsistencies

5.1. Identify model weakness

Archaeocyathids can have different textures. There exists a fair amount of intraclass variation among archaeocyathids. For example, there are recrystallized (white/gray) and red mud filled (red/brown) archaeocyathids, irregular (long) and regular (round) archaeocyathids, and more. We use labels **(a)** and **(b)** for the two primary textures (Fig. 1). The Mask R-CNN segments **(a)** (archaeocyathids with pitted textures; mean precision $= 0.84$; mean recall $= 0.67$) better than it segments **(b)** (archaeocyathids filled with red mud; mean precision $= 0.56$; mean recall $= 0.25$) (Fig. 9).

Test effect of optimal texture. To test the effect of the pitted texture on the segmentation quality, we stitch copies

% Pixels	Precision	Recall	IoU
None	0.86 ± 0.17	$\mathbf{0.75 \pm 0.26}$	$\mathbf{0.63 \pm 0.28}$
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	$\mathbf{0.91 \pm 0.12}$	0.63 ± 0.26	0.56 ± 0.27

Table 3. **Mapping different amounts of non-archaeocyathid pixels to red mud.** Metrics computed on 215 archaeocyathids from 4 images (mean and standard deviation reported). "% Pixels" indicates the percent of non-archaeocyathid 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-archaeocyathid 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-archaeocyathid pixels are replaced. When more non-archaeocyathid pixels are replaced, the precision of the archaeocyathid masks improves.

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

Table 4. **Mapping non-archaeocyathids to red mud using multiple, sequential tuning images.** Metrics computed on 215 archaeocyathids from 4 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 precision while decreasing recall and IoU.

of a square crop of the texture from one such pitted-textured archaeocyathid 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 ± 0.29 to 0.66 ± 0.31) though to a lesser extent than the predicted

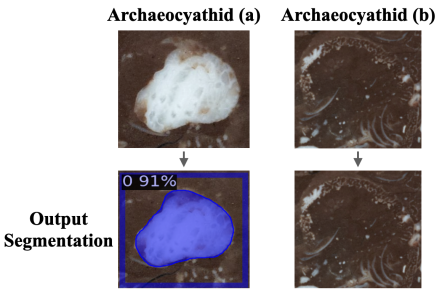


Figure 9. **Inconsistent segmentation of archaeocyathids.** Examples of output archaeocyathids masks with varying segmentation quality. The Mask R-CNN fails to segment the irregular, red mud filled (b) archaeocyathid but produces a complete mask for the regular, recrystallized (a) archaeocyathid that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

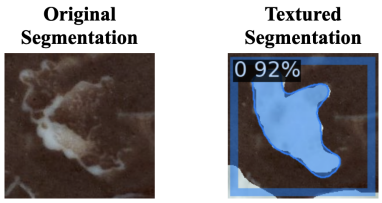


Figure 10. **Example of improvement due to texture replacement.** An example of a red mud filled archaeocyathid which the model misses in the unperturbed image (left) and perfectly segments when replaced with a pitted texture (right). A similar trend occurs for other red mud filled archaeocyathids.

masks for the inpainted non-archaeocyathids. Further analysis shows that there is an increase both in instance-level precision (mean increases from 0.78 ± 0.22 to 0.80 ± 0.17) and recall (mean increases from 0.77 ± 0.25 to 0.87 ± 0.23). Qualitatively, we find that the segmentation of previously poorly-segmented archaeocyathids improves (Fig. 10).

5.2. Mitigate model weakness

Mapping archaeocyathids to the optimal texture. 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 extension of the model-editing method to enforce a binary supercategorization, we use the method to strengthen the characteristics of the archaeocyathid class (i.e. a similar reason as the original work [31]). We tune with each of the 6 training images individually and produce two additional models tuned on 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 the unseen images (Tab. 5). Furthermore, mapping fewer archaeocyathid pixels to the pitted texture does not significantly change the performance from the original model (Tab. 6). Thus, mapping the archaeocyathids to the pitted texture does not seem to be an effective approach. This may be because the substituted pitted texture does not provide as drastic a visual contrast between archaeocyathids and non-archaeocyathids as does inpainting with red mud (Sec. 4).

Image	Precision	Recall	IoU
None	$\mathbf{0.86 \pm 0.17}$	$\mathbf{0.75 \pm 0.26}$	$\mathbf{0.63 \pm 0.28}$
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 215 archaeocyathids from 4 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	$\mathbf{0.86 \pm 0.17}$	$\mathbf{0.75 \pm 0.26}$	$\mathbf{0.63 \pm 0.28}$
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 215 archaeocyathids from 4 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, **(b)** archaeocyathid are replaced. The third row is when pixels for 18 **(b)** archaeocyathids (roughly 50% of archaeocyathid pixels) 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.

6. Combinations of Mappings

6.1. Simultaneous Mapping

When we run inference on training images where both non-archaeocyathids are inpainted with red mud and archaeocyathids are replaced with the pitted texture (Fig. 7d), 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; $lr = 10^{-4}$) on these images, 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.

Tuning Image	Precision	Recall	IoU
None	$\mathbf{0.86 \pm 0.17}$	$\mathbf{0.75 \pm 0.26}$	$\mathbf{0.63 \pm 0.28}$
A	0.82 ± 0.19	0.58 ± 0.29	0.41 ± 0.30
B	0.82 ± 0.18	0.58 ± 0.29	0.41 ± 0.31
C	0.81 ± 0.22	0.57 ± 0.32	0.44 ± 0.32
D	0.82 ± 0.20	0.61 ± 0.29	0.45 ± 0.32
E	0.76 ± 0.26	0.67 ± 0.30	0.43 ± 0.33
F	0.79 ± 0.24	0.66 ± 0.29	0.48 ± 0.33

Table 7. **Mapping simultaneously.** Mean and standard deviation are computed on 215 archaeocyathids from 4 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.

6.2. Sequential Mapping

In addition to simultaneously mapping to both perturbations, 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 with the non-archaeocyathids to red mud perturbation (Sec. 4.3) except we tune ($20k$ steps; $lr = 10^{-4}$) to an image with only the non-archaeocyathids inpainted followed by an image with only the archaeocyathids inpainted (and vice versa). We perform this mapping with image C since it produces the most improvement in the non-archaeocyathids to red mud tuning (Tab. 2). 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 alone (Tab. 8). This trend seems reasonable given the results from the earlier multi-image tuning experiments.

7. Conclusion

In this work, we focus on improving a fossil segmentation model first by identifying its model weaknesses via image perturbations and second by mitigating those weaknesses using model editing. Specifically, we study a Mask R-CNN trained on a small, fine-grain rock sample dataset to segment instances of archaeocyathid fossils.

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 for archaeocyathids) and intraclass inconsistencies (e.g. performance varied for archaeocyathids with different textures). Second, we extend a model-editing technique [31] 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 fewer pixels, and that sequentially performing tuning operations typically yields the same performance as tuning on the last edit alone.

Order	Precision	Recall	IoU
Ar	0.85 ± 0.18	0.73 ± 0.29	0.59 ± 0.31
No	0.91 ± 0.12	0.63 ± 0.26	0.56 ± 0.27
Ar, No	0.91 ± 0.12	0.62 ± 0.27	0.56 ± 0.28
No, Ar	0.85 ± 0.16	0.73 ± 0.28	0.59 ± 0.31

Table 8. **Mapping sequentially.** Metrics computed on 215 archaeocyathids from 4 images (mean and standard deviation reported); all edits 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. *Ar* and *No* results are from Tabs. 2 and 5. The result of each sequential mapping is similar to the result when editing with the last mapping only (i.e. *Ar, No* is similar to *No*).

We also demonstrate that model editing may not work in challenging circumstances when the visual appearance of an object is very similar to that of another type of object in the tuning image. Lastly, we find that while the tuning technique can negatively impact IoU and recall, it can improve precision when designed to mitigate interclass confusion (e.g. treating non-archaeocyathid pixels as red mud).

While our work focuses on improving a fossil segmentation model, our methodology may be useful for tackling similar problems that involve training a segmentation model on a small, fine-grained dataset. Additionally, further work could investigate what properties of the images cause one training image to be more effective than another. Additionally, future research could work towards mitigating the negative impacts on IoU and recall and investigate other methods for tackling the more visually-challenging situations.

Limitations. Given that our work focuses on editing a Mask R-CNN trained on a few images from one rock sample, the main limitation is that our findings may not generalize well to other segmentation models trained on small, fine-grain datasets. Our goal is to present a novel combination of techniques for investigating and improving the performance of segmentation models with a limited amount of labeled data, and we ran extensive experiments to substantiate our decisions. Thus, 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

[2] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv*, 2017. 3

[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. 1

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

[5] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv*, 2017. 3

[6] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017. 3

[7] Ruth Fong and Andrea Vedaldi. Occlusions for effective data augmentation in image classification. In *ICCV Workshop*, 2019. 3

[8] Jan Funke, Fabian Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C. Turaga. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *PAMI*, 41(7):1669–1680, 2019. 3

[9] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *NeurIPS*, 2015. 3

[10] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[12] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2022. 1, 3

[13] Xuejing Lei, Ganning Zhao, and C. C. Jay Kuo. Nites: A non-parametric interpretable texture synthesis method. *arXiv*, 2020. 3

[14] Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. Deep image prior. In *CVPR*, 2018. 3

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection, 2016. 4

[16] 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. 1, 2

[17] Drew Linsley, Junkyung Kim, David M. Berson, and Thomas Serre. Robust neural circuit reconstruction from serial electron microscopy with convolutional recurrent networks. *arXiv*, 2018. 3, 4

[18] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *ICML*, 2018. 3

[19] Ryan A. Manzuk, Adam C Maloof, Jaap A Kaandorp, and Mark Webster. Branching archaeocyathids as ecosystem engineers during the Cambrian radiation. *Geobiology*, pages 1–20, 2022. 2, 3

[20] Brian Matejek, Daniel Haehn, Haidong Zhu, Donglai Wei, Toufiq Parag, and Hanspeter Pfister. Biologically-constrained graphs for global connectomics reconstruction. In *CVPR*, 2019. 3

[21] Akshay Mehra and Adam Maloof. Multiscale approach reveals that Cloudina aggregates are detritus and not in situ reef constructions. *PNAS*, 115(11):E2526, 2018. 2, 3

[22] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022. 1, 3

[23] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022. 1, 3

[24] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018. 3

[25] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018. 3

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" explaining the predictions of any classifier. In *KDD*, 2016. 3

[27] Eugene Rosenberg, Omry Koren, Leah Reshef, Rotem Efrony, and Ilana Zilber-Rosenberg. The role of microorganisms in coral health, disease and evolution. *Nature Reviews Microbiology*, 5(5):355–362, 2007. 2

[28] Stephen M. Rowland and Roland A. Gangloff. Structure and paleoecology of lower cambrian reefs. *PALAIOS*, 3(2):111–135, 1988. 2

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 1

[30] Jéssica S. Santos, Rodrigo S. Ferreira, and Viviane T. Silva. Evaluating the classification of images from geoscience papers using small data. *Applied Computing and Geosciences*, 5:100018, 2020. 1

[31] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. In *NeurIPS*, 2021. 1, 2, 3, 4, 7, 8

[32] 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. 3

[33] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *International Conference on Learning Representations*, 2020. 1, 3

[34] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *arXiv*, 2018. 3

[35] 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. 1

[36] Tianlu Wang, Jieyu 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. 3

[37] 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. 3

[38] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 3

[40] 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. 3

[41] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AEIS*, 2018. 3

[42] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object Detectors Emerge in Deep Scene CNNs. *ICLR*, 2015. 3

[43] Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models, 2020. 1, 3