Weakly Supervised Instance Segmentation

Kevin Awoufack
awoufack@mit.edu

Arman Dave
armdave@mit.edu

Abstract

Instance segmentation is a common and increasingly necessary task in the field of computer vision but is also considered to be long and arduous since it requires precise human input to generate segmentations. While supervised methods to annotate new images do exist, they require a large pre-labeled dataset, which is a luxury the medical field does not often have. We propose a weakly supervised segmentation model to create instance segmentations of Common Objects in Context. It will use a limited set of annotated images to improve the self-supervised FreeSOLO model.

1. Introduction

Instance segmentation seeks to create annotated image masks where objects of the same class can each be identified, as opposed to semantic segmentation that masks all objects of the same class together. However, both methods, like any supervised learning model, require large amounts of annotated data. This is a painstakingly long process where people must decide the pixel level boundaries of various objects in an image in order to create a rich and diverse database. Many are forced to use large generic annotated databases, such as MS COCO [7], as the base of their models, even though these databases are not perfectly extensive. This holds especially true in the fields such as medicine and education, where access to images and other data is already sparse due to laws impeding collection or public access to such information.

A weakly supervised learning model is thus proposed as a way to create new annotations for images, specifically in the medical domain. The term "weakly supervised" implies a model that uses imprecise or sparsely labeled data and is resilient to noise in its dataset. We hope to adapt the FreeSOLO model [11], a self-supervised learning algorithm, into a weakly supervised version by injecting a few labeled data points at the mask generation step, which is then compared to the model's output to learn the features of just data. Rather than just fine tune the self learned model with annotations, we seek to use these annotations as part of the learning process.

We had originally hoped to work with the PanNuke dataset [3], which contains nuclei instance segmentation and classification dataset across 19 different tissue types, as in our original proposal. For the sake of data formatting and compatibility we decided to use the COCO dataset as it's widely used and encompasses many different types of objects with labeled segmentations. The idea of using a limited set of annotated points is agnostic to the dataset (although datasets of niche subjects yield interest specifically for their small scope).

2. Related Work

2.1. Instance Segmentation

Instance segmentation is an incredibly common task in computer vision yet remains challenging due to the arbitrary number of instances. Most approaches are either top down (starting with detectors and working down to pixel level) or bottom up (grouping the pixels into an arbitrary number of objects). Mask R-CNN [4] is an example of the former, which extends Faster R-CNN by adding a branch to predict the object mask in parallel to the task of creating a bounding box. A more recent adaptation of similar methods is BoxInst [9], which the segmentations from the bounding box annotation by introducing projection loss and pairwise loss terms to the CondInst model [8]. By contract, a bottom up method such as [2] uses a loss function that encourages the network to push away to a further point in feature space the pixels belonging to different instances and pull closer in feature space the pixels in the same instance. SOLO [10] introduces the notion of "instance categories", a combination of approximating the location of the object center of an instance and determining the size through a feature pyramid network, converting instance segmentation into a single-shot classification-solvable problem.

2.2. Weakly Supervised Learning

In supervised learning, each pixel is given a unique class as a label. However, in weakly supervised learning, a dataset consists of images and labels or annotations associated with each image, as opposed to a pixel-level correspondence. This partially solves the problem of the high cost of pixel-level labeled data by allowing weakly supervised models to use large-scale datasets such as ImageNet.

One option is to use data labelled with bounding boxes, but bounding boxes can still be costly to obtain. Constraints can be further relaxed by using image-level class labels. These image-level class labels are often used to derive Class Attention Maps (CAMs), which estimate the areas of each class in an image. Yet, even CAMs suffer from several issues - most prominent of which is that CAMs cannot distinguish different instances of the same class.

Latest state of the art, such as IRNet [5], overcome the issues of CAM by using two additional pieces of data: a class-agnostic instance map and pairwise semantic affinities. By combining the instance-agnostic CAMs with the class-agnostic instance map, IRNet gets instance-wise CAMs. The data from pairwise semantic affinities is used to propagate attention scores in the areas around instance wise CAMs, and thus generate a pseudo instance segmentation label. IRNet significantly outperformed previous efforts on the PASCAL VOC 2012, a standard dataset for weakly supervised models.

2.3. Unsupervised Segmentation

The model FreeSOLO [11] is the current state of the art for unsupervised segmentation - FreeSOLO achieves 9.8% AP_{50} without using any sort of annotated data during training. FreeSOLO is built on top of the simple SOLO architecture, with its novelty coming from the Free Mask part of its architecture (which we describe in greater detail in Proposed Methods). FreeSOLO uses Free Mask to generate coarse masks and feeds the masks as inputs to the regular SOLO model for segmentation.

2.4. Contrastive Loss

Contrastive loss has become popular in the spheres of unsupervised and self supervised learning. Clusters of data points belonging to the same class are pulled closer together in latent embedded space, while pushing away those of other classes. SimCLR [1] is one such paper that improved self-supervised and semi-supervised accuracy by increasing the number of parameter, in particular the batch size, and employing data augmentation to images. It's interesting because SimCLR found significantly better results with extremely large batch sizes and longer training periods, compared to conventionally trained supervised models that do well with few parameters. Contrastive loss also has benefits in supervised contexts. SupCon [6] explores how the use of labels can boost top-1 accuracies. In a supervised context it is possible to leverage the entire class as positives rather than just a single data point at a time as in most self supervised approaches. Thus, embedding are pushed closer together. In this paper we use contrastive loss as proposed by SimCLR.

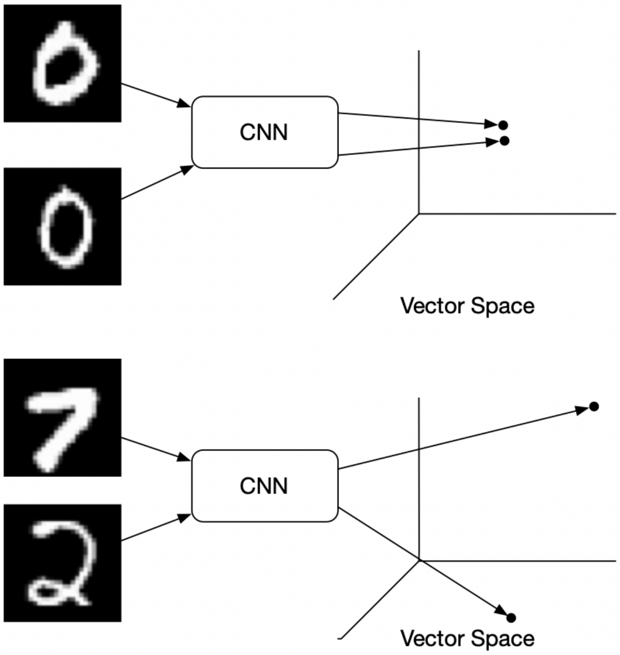


Figure 1. TODO: caption

3. Method

3.1. Background

We propose an adaptation of the FreeSOLO [11] network for the medical field that trains only on tissue to detect and segment instances of cancer nuclei. The visual fields are extremely similar so we inject a small sample of annotated data points.

We would train the Free Mask network, which uses a pretrained convolutional model like ResNet to generate a coarse segmentation map. This backbone network is used to construct queries Q and keys K. The keys are convolved by the queries by taking their cosine similarity. Their convolution is denoted S = Q ⊛ K, where S is the score map. It generates N = H' × W' queries, where H' and W' denote the down sampled spatial size, which are normalized, scored by maskness as defined in the paper [11], and then filtered for redundancies.

3.2. Weakly Supervised Contrastive Loss

After Free Mask has run on input data, for the small set of annotated training data within the set, the embedding of the highest scoring masks obtained is used to calculate the contrastive loss from the ground truth annotation. Distance can be computed through the cosine similarity of a vectorized bit mask, or more simply the intersection over union (IoU) of the polygon mask. We choose to implement the latter. The goal of contrastive loss is to discriminate the features of the input vectors, so the better masks are as close to the ground truth as possible. The contrastive loss function [1] is

$$L_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=0}^{2N} \mathbb{1}_{[k\neq i]} exp(sim(z_i, z_k)/\tau)}$$

where z_j/z_k is the embedding for 2N possible annotations, z_i is the embedding of a generated mask, and τ is a temperature normalization factor. There are 2N annotations compared against because contrast loss finds the distance from both predicted and labeled data to enable the push-pull of clusters in embedded space. The distance metric sim is defined as the cosine similarity between two vectors. It's important to note that as in SimCLR we're implementing as a modified version of cross-entropy loss because it's not a clear binary classification, more so calculating the likelihood two masks are the same instance. The closer the masks are in embedding, the lower the loss (-log(1) = 0). This loss function will be used to further train the backbone model of Free Mask to obtain better coarse masks that are then fed into the SOLOv2 model.

4. Experiment

4.1. Settings

Implementation. We largely implemented the parameter so FreeSOLO as was defined in the paper. However, for every generated mask in the first training pass through, we find the closest matching ground truth mask from the annotated data using the IoU as the distance metric. This chosen labeled mask is then encoded and saved to the COCO style database built by Free Mask as input to SOLOv2. Once the first loop is complete the backbone model of Detectron2 is extracted from the cfg file. FreeSOLO uses pretrained ResNet50 model but because we don't want a softmax classification we extract the res5 layer. The output of the layer, a tensor of (batch_size, 2048, 4, 4), is vectorized as the embedding used in our contrastive loss functions. Resnet requires a three channel input and the contrastive loss function requires that batches have a uniform number of masks in each input. To overcome this we set the batch size as twice the most number of polygons in one segmentation and repeat data points along the missing dimension. We choose this approach because the contrastive loss function benefits from larger batch sizes but we are limited in the amount of data available by nature of the problem.

We set a temperature of τ = 0.5 for our the contrastive loss function, as SimCLR does in their paper. Smaller temperature values generally have higher benefit than large having large ones but temperature values that are too low become statistically unstable and lose meaning as loss values blow up. The most important parameters for training the backbone model were the batch size and learning rate, which were 8 and 0.005. The former was limited by the computational power of our GPU but ideally would be 32 so as to learn the most features within a batch. The latter was an experimental parameter based on the fact that the pretrained weights already produced a model with very low loss so large changes in the learning rate caused the model to deviate from the local minima. There was little benefit from longer training epochs as well.

Datasets. FreeSOLO expects COCO annotated data as input and in its original implementation used the COCO train2017 and COCO unlabeled2017 as the set of unlabeled images, which exceeds 200k images. We only use the COCO train2017 dataset to save time, save our limited computational resources, and replicate an environment where there is a limited amount of labeled instance segmentation data. Since FreeSOLO is a two-tiered architecture (Free Mask and self-supervised SOLO), we approach our data in similar manner: Within the Free Mask training loop, Free Mask builds coarse masks from the train2017 dataset and then randomly splits them into train and test sets. There is never more than 50% of the coarse masks in the training dataset to avoid becoming a supervised learning problem.

Next, we use these coarse masks as input for SOLO to train on the COCO Val2017 dataset. Since we did not have powerful enough GPUs to train on the entire 5000 image set (the model would take over 24 hours to run), we randomly partitioned a sample of 500 images. We use a random partition of 500 of the remaining 4500 for the evaluation dataset.

4.2. Evaluation

We first used Free Mask to generate coarse masks, achieving a loss of 0.013525506. Here are some photos, alongside the masks they generated:

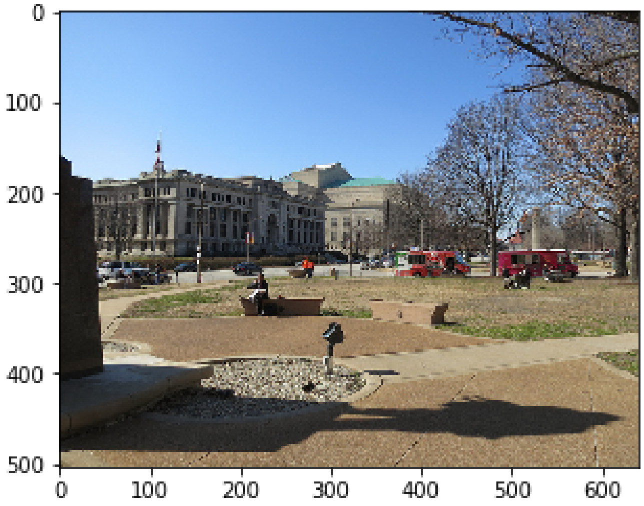


Figure 2. Image1

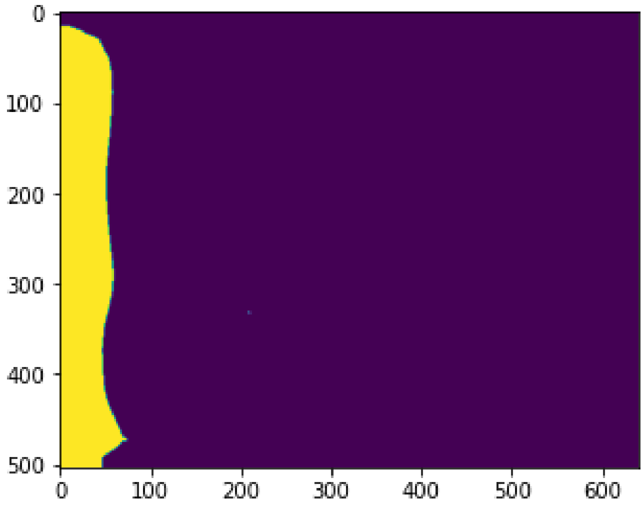


Figure 3. Image1Mask

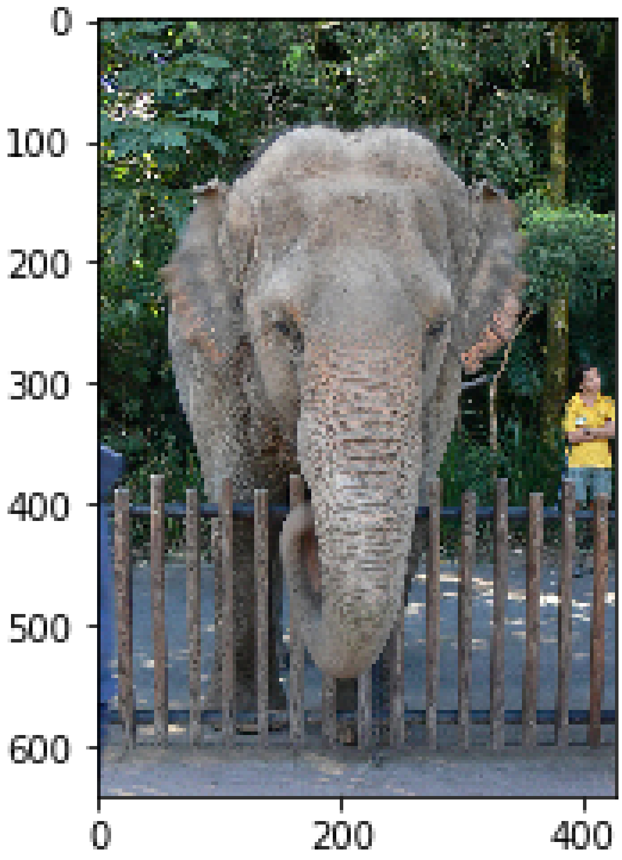


Figure 4. Image2

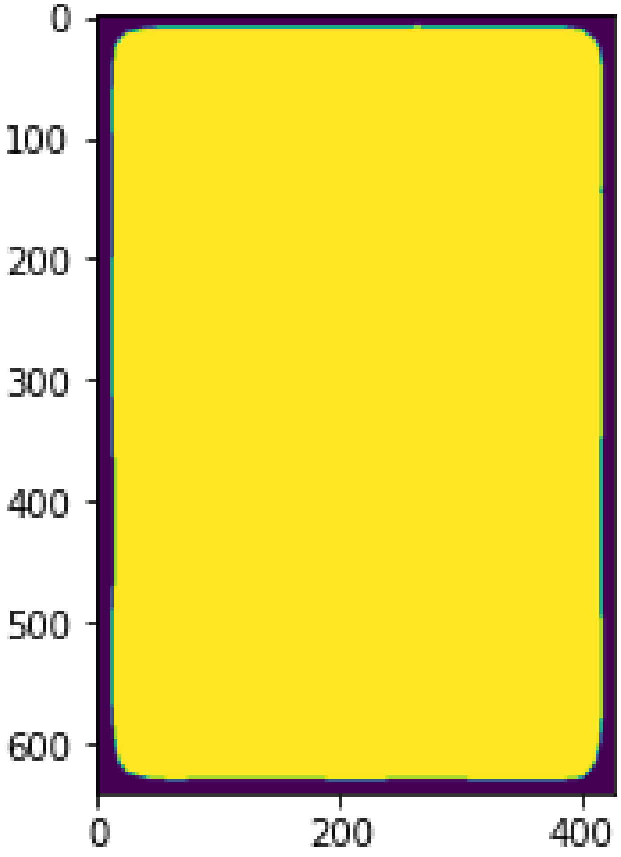


Figure 5. Image2Mask

For masks, we saw two common themes: Either the mask would cover the entire image, as seen in Image2, or the mask would pick up on objects in the left corner, as see in Image1. The first case is pretty intuitive in that Free Mask was only able to learn the entirety of the image. The second case is not as intuitive, but we suspect it may have to do with iteration of the image beginning on the left and proceeding column by column. As such, Free Mask would learn the left-hand-side of images better.

Next, we ran a baseline model without injecting our masks. Unfortunately, due to physical machine constraints, we were unable to run our model for the required time to reproduce or beat FreeSOLO's results. As is standard in instance segmentation, we use AP and AR metrics, combined with the IoU theshold. Given that tp is true positive, fp is false positive, and fn is false negative, average precision is defined as

$$precision = \frac{tp}{tp + fp},$$

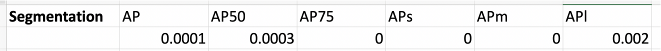
and average recall is defined as

$$precision = \frac{tp}{tp + fn}.$$

The intersection over union (IoU) is defined as

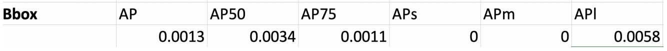
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

AP_x and AR_x are then defined as their respective measures with the IoU threshold set as x. We now present our baseline and mask-injected results on AP and AR metrics:



Segmentation	AP	AP50	AP75	APs	APm	APl
	0.0001	0.0003	0	0	0	0.002

Figure 6. Baseline Segmentation Results



Bbox	AP	AP50	AP75	APs	APm	APl
	0.0013	0.0034	0.0011	0	0	0.0058

Figure 7. Baseline Bbox Results

In the given time, it appears that our model was only able to get to the point where it learned to capture the entire image as an object. We give the following example images:

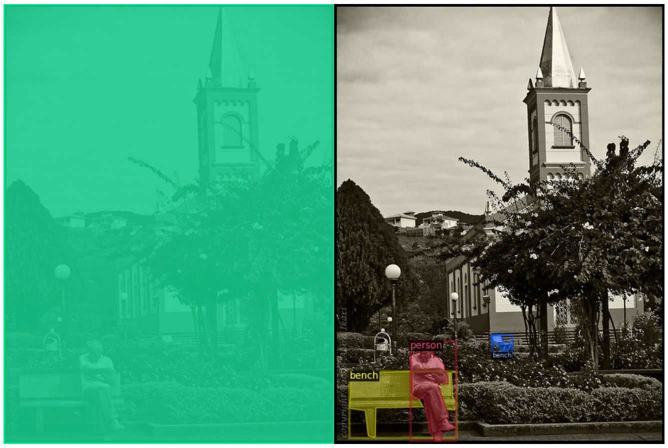


Figure 8. BaselineSeg1

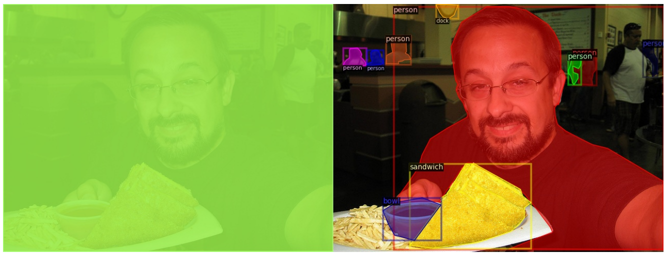
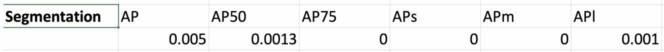


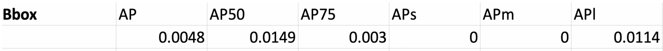
Figure 9. BaselineSeg2

Our next step was to inject our earlier masks into the training process. Unlike our baseline model, which seemed to get trapped in a local minima very quickly (loss at 2), the mask-injected training appeared to start off at a very high loss and consistently improved till we got loss of 0.4. We achieved the following results:



Segmentation	AP	AP50	AP75	APs	APm	APl
	0.005	0.0013	0	0	0	0.001

Figure 10. Baseline Segmentation Results



Bbox	AP	AP50	AP75	APs	APm	APl
	0.0048	0.0149	0.003	0	0	0.0114

Figure 11. Baseline Bbox Results

Our mask-injected model performed about the same as our baseline model, which is not surprising, given the sparsity of data on which we train. The results of the model suggest that the dense feature maps of Free Mask's backbone could not minimize the pairwise similarity loss of local features when running the ResNet50 model. Essentially all pixels are being treated as similar even if they are physically far in the image or embedded space. One thing to note is that FreeSOLO got a 12% AP score, but used both the COCO train2017 and COCO unlabeled2017 datasets as input. Another point of consideration for where our model may have failed is in training the pretrain backbone model. One reason for low AP scores was that we were training on too few data points. The low contrastive loss that hardly fluctuated may have been an indication that the model was overfitting the masks in embedded space. This is simply an inherent problem of machine learning and having too little data for a powerful model without properly set regularization metrics. Another point of issue was that the masks the training loop was fed was bad data. As we can see by the fact that most masks simply covered the image, which would produce very similar embeddings in feature space and low loss.

5. Conclusion

Based on FreeSOLO, our paper has thus introduced how it could be possible to create new annotations from a dataset that is largely unlabeled. While we did not achieve results that surpassed FreeSOLO, it raises interesting questions about what work can be done to improve unsupervised models within a reasonable scale at the level of commodity machines. The process of generating an annotated dataset from raw images remains a largely human intensive task.

Limitations. The primary limitation of our implementation of the project is the computational resources. There are two views: theoretical and practical. On a theoretical level, most unsupervised and self supervised models benefit greatly from large batch sizes and many epochs of training, as in SimCLR. The same can be said about FreeSOLO. They used about 200k images between COCO train2017 and COCO unlabeled2017 datasets, trained for 300,000 epochs, and had batch sizes of 32 images. In the span and scope of our physical resources, it was not possible to implement a fully trained weakly supervised on Google Colab where there is at most one GPU available and training can be interrupted, whereas FreeSOLO ran on V100 GPU and used a pretrained backbone model without further training it. On our setup simply processing 500 images for 30,000 epochs would take well over 24 hours.

Some limitations came from of choice of FreeSOLO as our base model. The Free Mask portion that generates the coarse masks does so through unsupervised object discovery using the features found by the backbone model. This introduces inconsistency into the design of FreeSOLO as a whole because there is no guarantee that the same object and boundaries will be found between iterations of Free Mask, if any are found at all. At times not even 20 coarse masks could be generated and most of the time they were not useful masks. Without sufficient training data, it wasn't possible to train the backbone model, even in the context of weakly supervised training. Thus, there were many iterations simply trying to simply get enough annotations out of Free Mask. FreeSOLO in general is also a very recent paper that was published in the past few months, so not all its viewable code was completely up-to-date, which required us to spend a large chunk of our preliminary time getting the base model running.

Next Steps. A reasonable next step would be to train our model on a more powerful GPU cluster to accurately replicate the training environment of FreeSOLO. Once there are more concrete results, there are two approaches we propose to improve the coarse mask generation abilities. The first would be to introduce a bottom up pixel-wise recognition step. While FreeSOLO is interesting in the fact that it's a class agnostic approach, it also means that it requires intensive training to find meaningful features and, as many of our results show from masks that cover the entire image, pixels may not be properly contextualized as objects. While such a suggestion would sacrifice the class agnostic property of FreeSOLO, there exist many pretrained models which can then be contrasted with negative space as an indirect form of localization. The second course of improvement would be to introduce some form of attention into the model. Each image is currently processed individually into its queries and keys. Thus no learning occurs when coarse masks are generated.

6. Individual Contributions

I was primarily responsible for much of the implementation details. This included figuring out how the COCO dataset was formatted and how to use the API to stream chunks of data. I also had to effectively partition the data in order to work with manageable chunks given that COCO is 20 GB. When it came to running the model, I overcame several hurdles: The primary hurdle that took nearly 12 hours to overcome was tracing the root source of a segmentation fault. Since seg faults do not report tracebacks, and the model is asynchronous, this was an exhaustive process to trace the root of the problem to shape errors in the tensors. I further overcame an error where FreeSOLO appeared to modify tensors in-place during the backpropagation process, which is illegal. It appears that the authors of FreeSOLO fixed this in their implementation of SOLO (the model they wrote upon which FreeSOLO is based), but did not push these changes to the files of SOLO they included with FreeSOLO. When we ran the model, I oversaw the self-supervised SOLO process of the model, which included sanity checks on our results and tracking the loss. Overall, it required a lot of attention to detail and understanding the architecture of FreeSOLO to a high extent in order to get the model to run given our limited compute resources.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3

[2] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 1

[3] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology*, pages 11–19. Springer, 2019. 1

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[5] Dipendra Jha, Logan Ward, Zijiang Yang, Christopher Wolverton, Ian Foster, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Irnet: A general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2385–2393, 2019. 2

[6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 2

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[8] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision*, pages 282–298. Springer, 2020. 1

[9] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5443–5452, 2021. 1

[10] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020. 1

[11] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. *arXiv preprint arXiv:2202.12181*, 2022. 1, 2