# Approaching the Issue of Limited Annotation for Instance Segmentation

*Vishnu Doppalapudi*

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-82
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-82.html

May 14, 2021

# Approaching the Issue of Limited Annotation for Instance Segmentation

by Vishnu Doppalapudi

## Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

**Committee:**

DocuSigned by:
Trevor Darrell
50EBCB54F30D41B...

Professor Trevor Darrell
Research Advisor

5/13/2021

(Date)

* * * * * * *

DocuSigned by:
Joseph Gonzalez
07C27C781A3A4D9...

Professor Joseph Gonzalez
Second Reader

5/13/2021

(Date)

Approaching the Issue of Limited Annotation for Instance Segmentation

# Abstract

Approaching the Issue of Limited Annotation for Instance Segmentation

by

Vishnu Doppalapudi

Masters of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

Instance segmentation is a challenging task. It requires a model to localize each object in an image pixelwise, and unlike semantic segmentation requires the model to discern between different instances of objects with the same class label. Recent advances in instance segmentation, especially with deep learning models, are predicated on the availability of large datasets with high quality annotations. Without large datasets, the state of the art models with tens of millions of parameters face problems such as overfitting. However, constructing large labeled datasets is very expensive, and for many real-world applications it is not feasible. There have been many approaches to tackle this issue. One of these is semi-supervised instance segmentation, which is the use of abundant box annotations but limited mask annotations to learn an instance segmentation model. Another is few shot instance segmentation, the use of limited box and mask annotations to learn an instance segmentation model. This thesis introduces learning techniques for both of these approaches.

# Contents

# List of Figures

# List of Tables

# Acknowledgments

I would like to thank my parents for their support and encouragement of both my academic and non-academic goals. I would also like to thank Sharon Zhou, Dr. Xin Wang, Professor Fisher Yu, and Professor Trevor Darrell for their guidance and mentorship of my research at UC Berkeley.

# Chapter 1

# Introduction

Instance segmentation is a task which requires a model to classify and localize each object in an image pixelwise. It is used in many applications such as autonomous driving where a localization more precise than the bounding box obtained from object detection is required. However, the additional precision in instance localization makes this task even more challenging. Recent advances in instance segmentation, especially with deep learning models, are predicated on the availability of large datasets with high quality annotation. Without large datasets the state of the art models with millions of parameters face problems such as overfitting. However, constructing large labeled datasets is very expensive and for many real-world applications it is not feasible.

This thesis introduces learning techniques for instance segmentation which maximize performance with significantly less annotation than is typically required. When discussing reducing the amount of annotation for instance segmentation models, it is important to first understand that there are two types of annotation for each object, a bounding box and a pixelwise mask. There are two primary areas of research where we learn instance segmentation models with significantly less data. The first is few shot instance segmentation, where the model is given only a few (typically between 1 - 20) box and mask annotations per class. Chapter 2 outlines my work in this field. The second is semi-supervised instance segmentation, where the model is given 100% of the available box annotation and significantly reduced (typically between 0% - 10%) of mask annotation. Chapter 3 will outline my work in this field.

# Chapter 2

# Few Shot Instance Segmentation

## 2.1 Introduction

The goal of few shot instance segmentation (FSIS) is to learn to predict instance-level segmentation masks for a set of target classes given only a few box and mask annotations for the target classes and abundant box and mask annotation for a larger set of unrelated classes. This is an important problem because annotating the amount of data required to train a deep learning model such as Mask R-CNN [4] to accurately predict instance-level segmentation masks is very expensive. For many potential applications such as medical imaging, annotating the required amount of data is simply not feasible [9]. However, achieving the goal of FSIS with deep learning models like Mask R-CNN is very challenging. Unlike humans, who can generalize to unseen examples given only a few examples, deep learning models struggle to learn generalizable features when trained with a few examples and overfit to the examples they are trained with.

## 2.2 Related Work

That being said, FSIS models are largely based on Mask R-CNN. This model takes as input the image to segment. It first uses a large backbone network such as ResNet-50 [3], along with a Feature Pyramid Network (FPN) [5], to learn features from the input image. These features are then inputted into a region proposal network (RPN) to determine the probability that predefined anchor boxes contain an object. These anchor boxes and the predicted scores are then used to predict final detections and segmentation masks.

Due to the complexity of instance segmentation, FSIS is a very under-explored problem. The three prior works of note in this field are Siamese Mask R-CNN [7], Meta R-CNN [11], and Fully Guided Networks [1]. These are all meta-learning based methods and all follow the same fundamental approach. They have two inputs, one being the image to segment and the other being cropped instances of objects of the class they want to segment. For example, to segment cars the second input would be cropped instances of cars. As one can see, the function these models aim to approximate is very complex. The differences between each individual approach are fairly simple: Siamese Mask R-CNN has a unique backbone for each of the inputs and does not have a fine-tuning stage, Meta R-CNN uses the features from the cropped instances to guide the model prediction only at the RPN stage, and Fully Guided Networks uses the features from the cropped instances to guide the RPN, the detection head, and the segmentation head of the network. I hypothesize that the lack of fine-tuning in Siamese Mask R-CNN prevents its performance from scaling well with additional data.

Few shot object detection (FSOD) is a more widely explored problem where the model only predicts a bounding box for each object rather than a pixelwise mask, as it does for FSIS. Many methods of FSOD follow the same approach as the previous approaches for FSIS. However, unlike these previous approaches, [10] propose an approach for FSOD that is rooted in transfer learning instead of meta learning. This method uses the Faster R-CNN architecture [8] and first trains a model, denoted the "base model", on the abundant data that is available for the base classes. Then, it fine-tunes that model with the limited annotated data available for the novel classes. This approach performs better than the previous meta learning methods for FSOD, so we seek to adapt it to FSIS.

ShapeProp [13] is the state of the art model for semi-supervised instance segmentation, a problem in which the model has abundant box and limited mask annotation. The model is an extension of Mask R-CNN and uses the additional box annotation to learn a saliency map where salient regions within a detection are activated. These regions are then propagated with a message passing module to learn the shape activation, which is a prior for the final mask prediction. Due to the success of this model in semi-supervised instance segmentation we seek to adapt it to FSIS.

## 2.3 My Approach

Due to the success of the aforementioned transfer learning method over previous methods of FSOD, I apply that transfer learning method to FSIS. I first train a base model with the abundant data in the base classes, then I fine tune this base model on the limited data I have for the novel classes. During the fine-tuning phase, I update the parameters of a variable number of layers based on the number of examples available for each class. I use the Mask R-CNN and ShapeProp models for my experiments, and as discussed below I make further improvements to both models to adapt them to FSIS.

### 2.3.1 Unfreezing Layers

The more layers that are unfrozen during fine-tuning the more parameters in the model are updated. Typically updating a large number of parameters is desirable, as much of the success of deep learning methods is from the use of overparameterized models with large datasets [12]. However, when the dataset is small, as is the case for the dataset of novel examples (i.e. novel set) in FSIS, updating a large number of parameters is known to cause overfitting. As a result, when only a few number of examples (i.e. shots) are available per class we only update the parameters of the last layer of the model. As the number of shots increases, the number of layers for which we update the parameters can be increased. For different numbers of shots, the number of layers to unfreeze can be treated as a hyperparameter and the optimal number can be found by performing a basic grid search.

### 2.3.2 Proposal Sampling

Empirically, I found that fine-tuning without additional improvements performs well on some novel classes and poorly on others. The poor performance on these other classes is due to the inability of the model to detect instances of these classes. This is despite the RPN being able to detect some of these instances and classify them as objects. This suggests that the issue is that the classification head is unable to classify these instances correctly, and this is likely caused by the significant class imbalance present in traditional R-CNN models. In the original setup I train the classification head with all available foreground proposals, or proposals that have been matched to a ground truth class, from the RPN (typically less than 20) and more than 450 background proposals, or proposals that have not been matched to a ground truth class. This severe class imbalance makes it easy for the model to misclassify instances from more difficult classes as background rather than foreground. To mitigate this issue I randomly sample a reduced number of background proposals such that there is less than 200 background proposals per image. This reduces the class imbalance issue considerably and allows the model to correctly detect many more difficult classes. Although this significantly reduces the number of false negatives, as shown in Figure 2.1, the number of false positives increases slightly, as shown in Figure 2.2. There is a tradeoff between the number of false positives and false negatives. For larger numbers of sampled negative proposals the number of false negatives goes up and the number of false positives go down, whereas for smaller values of sampled negative proposals the number of false negatives goes down and the number of false positives goes up. The optimal number of negative proposals can be found through a grid search.

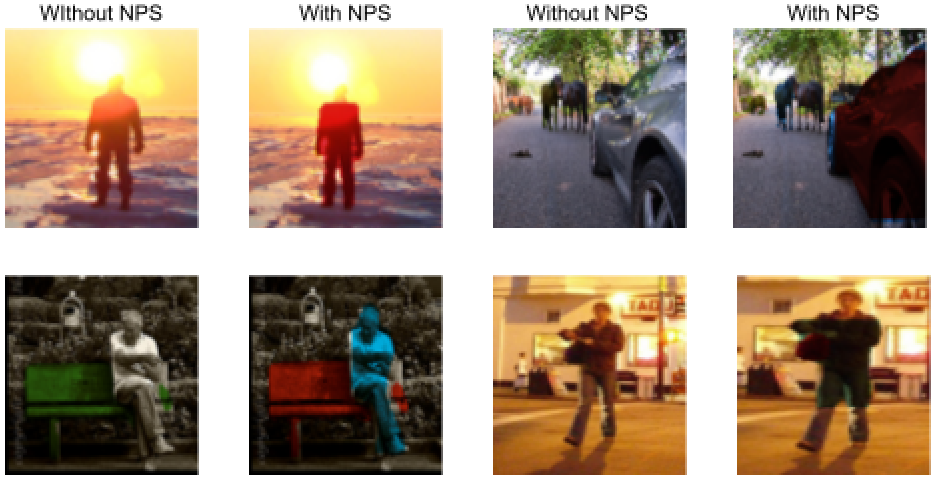

Figure 2.1: Examples of Mask RCNN with and without negative proposal sampling in the 20 shot FSIS setting

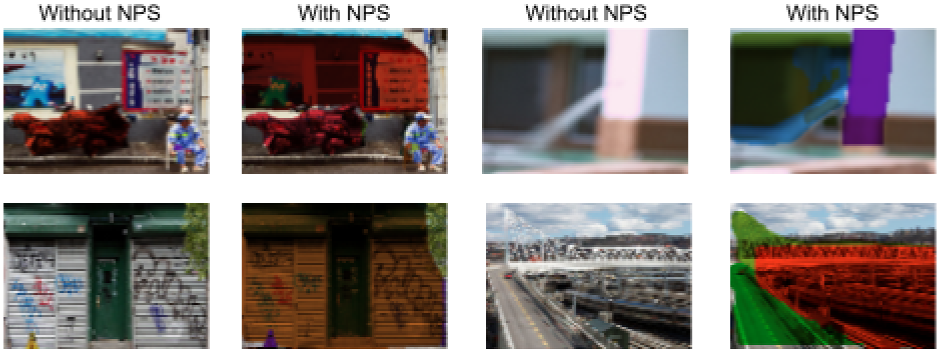

Figure 2.2: Examples showing the increase in false positives for ShapeProp with and without negative proposal sampling

### 2.3.3 Proposal Relocation

For low shot models, the detections used for mask prediction often fail to localize the instance properly, either being too big or too small. Although mask performance suffers in both cases, it especially suffers in the latter case because the mask prediction is constrained to be within the detection, and if the detection is too small the mask prediction will be forced to miss some areas of the target object. To mitigate this issue, I propose to use the ShapeProp module

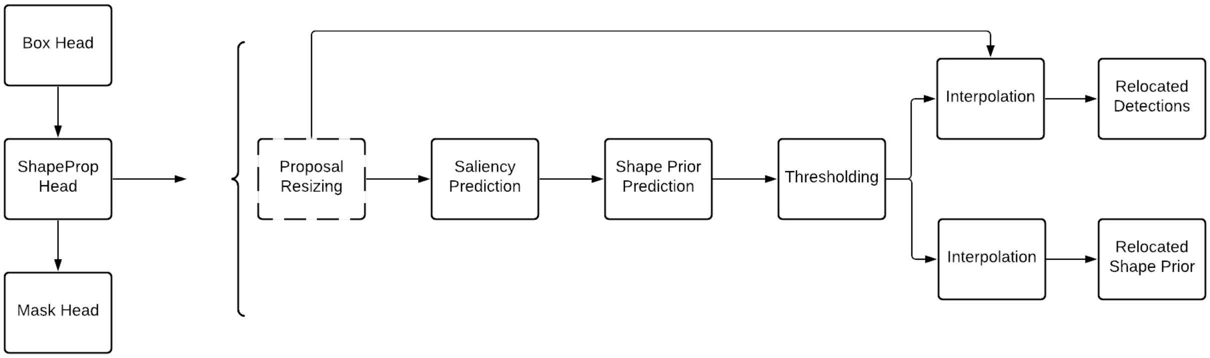

Figure 2.3: Adjustments made to ShapeProp module for Proposal Relocation

to jointly optimize the box and mask predictions by relocating the box prediction. I achieve this by first scaling the width and height of each box by a constant factor. Then, using a predetermined threshold, I identified a sub-region within the shape activation that contains the object. After this region is located, I use interpolation to recover a new detection and shape activation. I call this Proposal Relocation, and it is illustrated in Figure 2.3. Because proposal relocation does not require the addition of any new parameters, it does not require training a new model. Rather, it can be used with a model checkpoint from the original approach and be added during inference.

### 2.3.4 Class Agnostic Saliency

In the original ShapeProp model the predicted saliency was class specific and it assumed a large amount of box annotation for supervision during training. However, in the few shot setting there are not nearly as many box annotations for the novel classes, so I instead propose to learn a class agnostic saliency. Suppose there are $N$ novel classes and $K$ examples per novel class. With class-specific saliency there are $K$ examples to learn the correct saliency for that class. However, for the class agnostic case, all the base annotation and the $N * K$ boxes from the novel classes are available to learn the saliency. Although this approach may prevent the saliency from capturing some of the fine features of each individual class, a class agnostic saliency would better capture the coarse features of the target instance which could then be iteratively refined by the shape activation and final mask prediction to capture the fine features of target instances of novel classes.

## 2.4 Experiments

For the experiments I used the MS COCO [6] and LVIS [2] datasets. Since the MS COCO dataset is not itself a few shot dataset I manually selected 20 of its 80 categories, randomly selected $K$ examples for each of these classes (novel classes), and discarded the rest to create a few shot dataset. This poses an additional yet realistic challenge because many instances of the novel classes are left without annotation when training the model. This missing annotation would further incline the model to classify these instances as background rather than foreground. I use mean average precision (mAP) as the quantitative metric to assess the performance of each method. I refer to FS + MRCNN as the fine tuning approach applied to Mask RCNN and FS + SP as the fine tuning approach applied to ShapeProp. Further, CAS refers to Class Agnostic Saliency, PR refers to Proposal Relocation, and NPS refers to Negative Proposal Sampling.

### 2.4.1 1 shot

| Method | Novel Box AP | Novel Segm AP |
|---|---|---|
| Meta R-CNN | – | – |
| **Siamese Mask R-CNN** | **5.309** | 3.899 |
| FS + MRCNN (mine) | 2.397 | 2.767 |
| FS + MRCNN + NPS (mine) | 3.323 | 3.667 |
| FS + SP (mine) | 2.116 | 2.743 |
| FS + SP + NPS (mine) | 2.795 | 3.459 |
| FS + SP + PR (mine) | 3.019 | 3.273 |
| FS + SP + PR + NPS (mine) | 3.653 | 3.729 |
| FS + SP + CAS (mine) | 2.429 | 3.113 |
| FS + SP + CAS + NPS (mine) | 3.001 | 3.714 |
| **FS + SP + CAS + NPS + PR (mine)** | 3.706 | **3.933** |

Table 2.1: 1 shot FSIS results

Table 2.1 summarizes the results of the experiments in the 1 shot setting. Although ShapeProp on its own achieves a very similar segmentation AP as Mask R-CNN, the additions of class agnostic saliency, proposal relocation, and negative proposal sampling improve performance considerably and outperforms Mask R-CNN in both box and segmentation AP. Figure 2.4 shows qualitative examples of Mask R-CNN, ShapeProp, and ShapeProp w/ Proposal Relocation and demonstrates how Proposal Relocation improves performance in the 1 shot setting. However, despite the gains achieved with proposal relocation, my best-performing method of ShapeProp with Class Agnostic Saliency, Proposal Relocation, and Negative Pro-

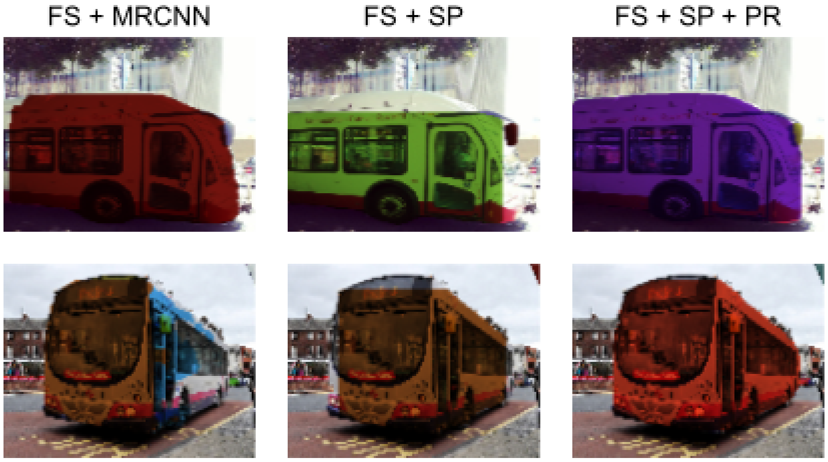

Figure 2.4: Examples of Mask RCNN, ShapeProp, and ShapeProp w/ Proposal Relocation in the 1 shot FSIS setting

posal Sampling does not achieve significantly higher segmentation AP than the baseline of Siamese Mask-RCNN and achieves a lower box AP.

### 2.4.2 5 shot

| Method | Novel Box AP | Novel Segm AP |
|---|---|---|
| Meta R-CNN | 3.5 | 2.8 |
| Siamese Mask R-CNN | 5.436 | 4.06 |
| FS + MRCNN (mine) | 6.722 | 6.945 |
| FS + MRCNN + NPS (mine) | 7.25 | 7.489 |
| FS + SP (mine) | 5.985 | 6.867 |
| FS + SP + CAS (mine) | 5.806 | 6.527 |
| FS + SP + PR (mine) | 6.881 | 7.128 |
| FS + SP + NPS (mine) | 6.437 | 7.416 |
| **FS + SP + NPS + PR (mine)** | **7.701** | **7.879** |

Table 2.2: 5 shot FSIS results

Table 2.2 summarizes the results in the 5 shot setting. As in the 1 shot setting, FS + ShapeProp achieves a similar segmentation AP for the novel classes as FS + Mask R-CNN, both with and without negative proposal sampling, and with the addition of Proposal Relocation the box and segmentation AP achieved by ShapeProp exceeds that of Mask R-CNN. Class

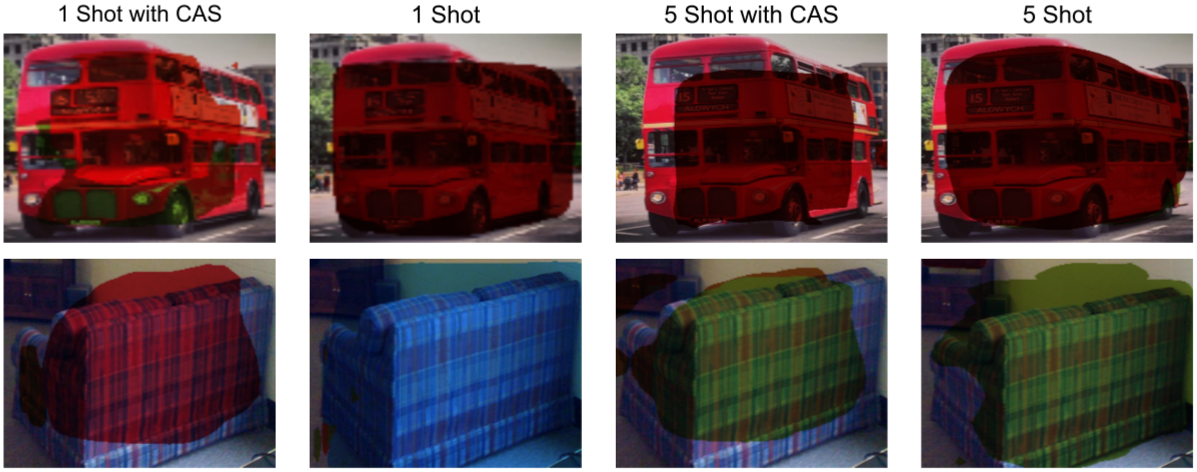

Figure 2.5: How the performance of the class specific vs class agnostic saliency varies from the 1 shot to the 5 shot settings

agnostic saliency does not improve performance in this setting, which implies that 5 boxes are enough supervision for the ShapeProp module to learn fine image features with a class specific saliency which convey more information than the improved coarse features present with a class agnostic saliency. This can be seen in Figure 2.5, as the quality of the class specific saliency improves considerably from 1 to 5 shots. As in the 1 shot setting, negative proposal sampling provides a significant boost in performance by reducing the number of false negatives, and Figure 2.1 shows some examples of this from the MS COCO validation set. It can also be seen that in the 5 shot setting my method significantly outperforms both baselines of Siamese Mask R-CNN and Meta R-CNN.

### 2.4.3 10shot

| Method | Novel Box AP | Novel Segm AP |
|---|---|---|
| Meta R-CNN | 5.6 | 4.4 |
| Siamese Mask R-CNN | 5.492 | 4.076 |
| FS + MRCNN (mine) | 8.37 | 8.318 |
| FS + MRCNN + NPS (mine) | 8.809 | 8.765 |
| FS + SP (mine) | 7.634 | 8.376 |
| FS + SP + CAS (mine) | 7.907 | 8.314 |
| FS + SP + PR (mine) | 8.432 | 8.582 |
| FS + SP + NPS (mine) | 8.313 | 9.175 |
| **FS + SP + PR + NPS (mine)** | **9.318** | **9.324** |

Table 2.3: 10 shot FSIS results

Table 2.3 summarizes the results in the 10 shot setting. The findings from these results are the same as those of the 5 shot setting except that my best performing method of ShapeProp with Proposal Relocation and Negative Proposal Sampling outperforms the baselines even more.

### 2.4.4 20 shot

| Method | Novel Box AP | Novel Segm AP |
|---|---|---|
| Meta R-CNN | 6.2 | 6.4 |
| Siamese Mask R-CNN | 5.509 | 4.094 |
| FS + MRCNN (mine) | 9.669 | 9.437 |
| FS + MRCNN + NPS (mine) | 10.732 | 10.365 |
| FS + SP (mine) | 9.232 | 9.911 |
| FS + SP + CAS (mine) | 9.161 | 9.514 |
| FS + SP + PR (mine) | 10.0 | 10.123 |
| FS + SP + NPS (mine) | 9.95 | 10.827 |
| **FS + SP + NPS + PR (mine)** | **11.18** | **11.155** |

Table 2.4: 20 shot FSIS results

Table 2.4 summarizes the results in the 20 shot setting. The findings from these results are the same as those of the 5 and 10 shot settings, except in this setting FS + ShapeProp achieves a higher segmentation AP for the novel classes than FS + Mask R-CNN, which is likely due to the improved quality of the shape activation resulting from increased annotation.

### 2.4.5 LVIS

| Method | Novel Segm AP | Novel Box AP |
|---|---|---|
| Mask R-CNN | 19.917 | 20.181 |
| ShapeProp | 21.478 | 20.542 |
| ShapeProp + PR (mine) | 21.526 | 21.12 |

Table 2.5: Overall LVIS Results

| Method | Novel Segm APr | Novel Segm APc | Novel Segm APf |
|---|---|---|---|
| Mask R-CNN | 7.814 | 18.218 | 27.134 |
| ShapeProp | 9.91 | 19.443 | 28.833 |
| ShapeProp + PR (mine) | 9.837 | 19.631 | 28.779 |

Table 2.6: LVIS segmentation results broken down by class frequency

| Method | Novel Box APr | Novel Box APc | Novel Box APf |
|---|---|---|---|
| Mask R-CNN | 7.563 | 17.55 | 28.665 |
| ShapeProp | 9.227 | 17.964 | 28.391 |
| ShapeProp + PR (mine) | 9.591 | 18.98 | 28.575 |

Table 2.7: LVIS box results broken down by class frequency

In addition to the experiments on the few shot MS COCO dataset, I also evaluated this method on LVIS, a dataset with a long-tail data distribution. The classes in this dataset are divided into three subsets based on the number of images they appear in. Rare classes appear in $< 10$ images, common classes appear in between $10 - 100$ images, and frequent classes appear in $> 100$ images. The AP of the model on rare classes is denoted APr, AP on common classes is denoted APc, and AP on frequent classes is denoted APf. The challenge this dataset poses is that it has many classes, and as a result of its long-tail data distribution some classes occur with high frequency and others occur with very low frequency. Another challenge in this dataset is that as a result of having many classes some classes are very similar to others, and the model must discern between these very similar classes (e.g. mandarin orange and clementine). To evaluate my method on this dataset I train a ShapeProp model without proposal relocation as a baseline and then evaluate the model with proposal relocation. Because there are many ground truth instances in every image and thus many foreground proposals I do not include negative proposal sampling.

As shown in Tables 2.5, 2.6, and 2.7, although proposal relocation does not hurt overall performance, it does not provide a significant improvement either. I believe this is because there are many instances in the LVIS dataset that are directly adjacent to each other, and after resizing the shape activation the model struggles to localize the target instance due to the presence of many adjacent instances. This can make the model prediction worse, and some examples of this can be seen in Figure 2.7. However, given the low number of shots for rare classes many instances have improved box and mask predictions as a result of proposal relocation, and this can be seen in Figure 2.6.

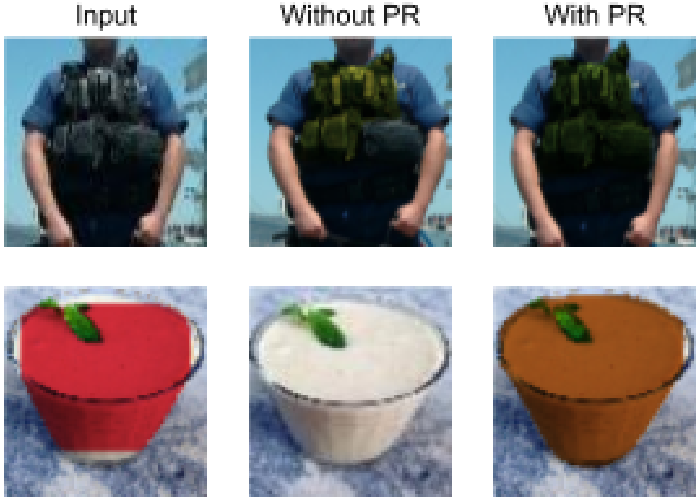

Figure 2.6: Positive examples of Proposal Relocation in LVIS

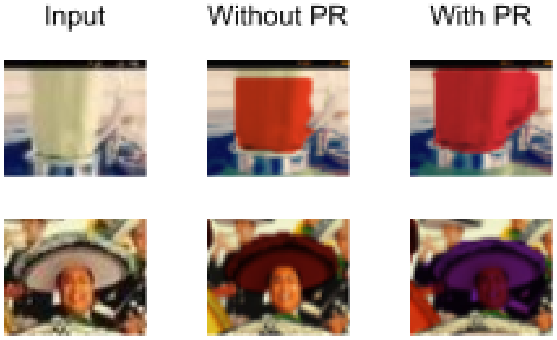

Figure 2.7: Negative examples of Proposal Relocation in LVIS. Note that the target instance in the second example is the smoothie inside the blender, not the blender itself.

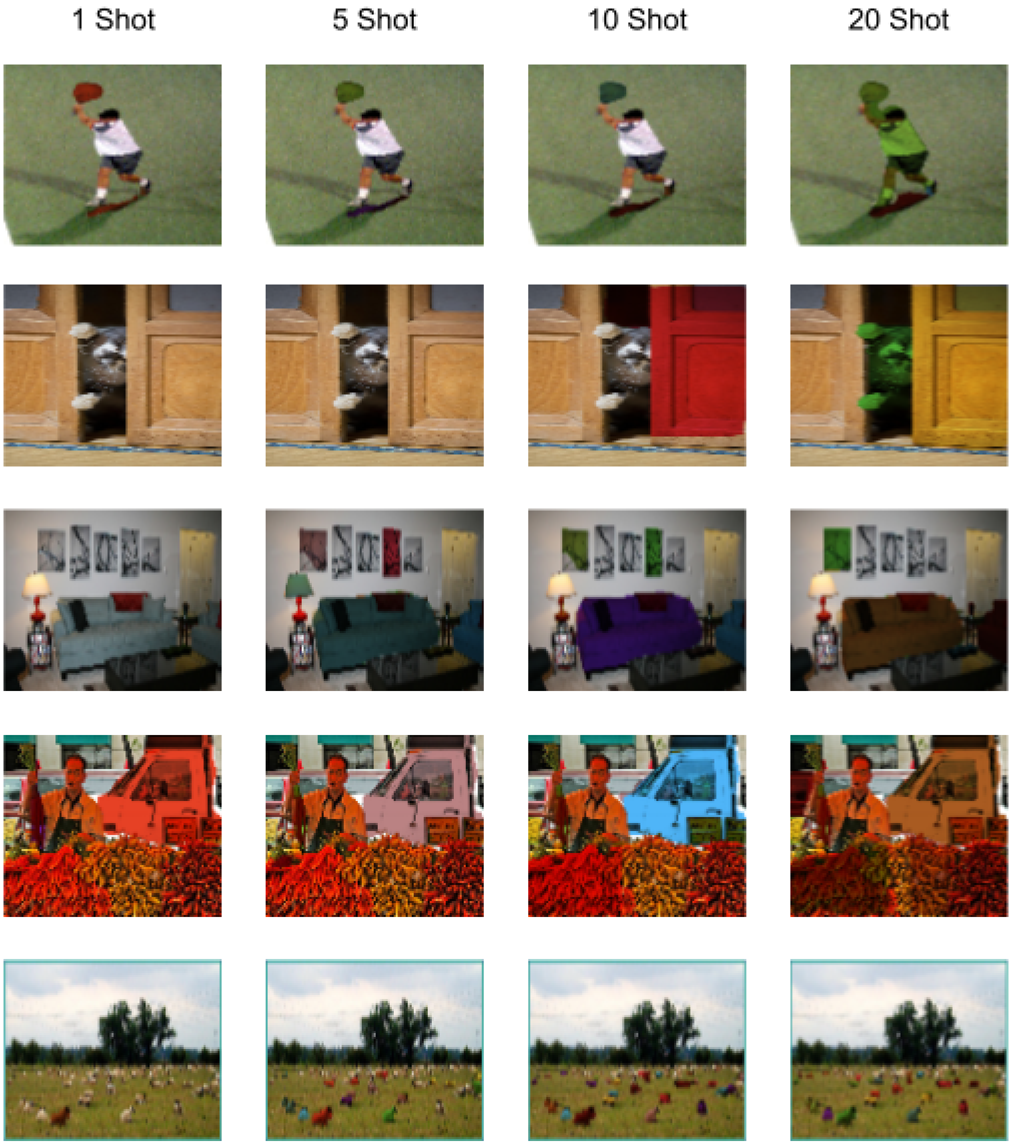

Figure 2.8: Examples from the COCO validation set of ShapeProp w/ NPS and PR as number of shots increases

## 2.5 Conclusions

I discussed three approaches to improve model performance: Class Agnostic Saliency (ShapeProp Only), Negative Proposal Sampling, and Proposal Relocation (ShapeProp Only). Class agnostic saliency was successful in improving segmentation AP in the one-shot setting but as the number of shots increased it's efficacy decreased. This is likely because the model does not require many boxes to encode important fine features in the saliency prediction for each class. Negative Proposal Sampling was successful in improving segmentation AP for all numbers of shots, as demonstrated by the quantitative results in Tables 2.1, 2.2, 2.3, and 2.4. Although it did increase the number of false positives, as one would expect when the number of background proposals used to supervise the box head decreases, it also decreased the number of false negatives, so much so that there was a substantial improvement in the box and segmentation AP. These findings can be seen in the Figures 2.2 and 2.1. Proposal Relocation with the ShapeProp module also led to substantial improvements in both the box and segmentation AP for all numbers of shots without introducing any additional model parameters. As I hypothesized, it allowed the model to re-adjust predictions from the box head that were too small, as can be seen in the examples in Figure 2.4.

It can be concluded that my methods are able to utilize the additional data as the number of shots increases much more effectively than the meta-learning baselines of Siamese Mask R-CNN and Meta R-CNN. See Figure 2.8 for some examples from the MS COCO validation set which show how the model prediction evolves as the number of shots increases. This table shows that the number of shots necessary to obtain a good quality prediction is highly dependent on the example. As I hypothesized, Siamese Mask R-CNN was not able to effectively utilize the additional data available as the number of shots increased because it did not supervise the model with the additional data from the novel classes. However, despite its use of fine-tuning, Meta R-CNN performed significantly worse than my methods in all settings. This suggests that the current meta learning based methods for FSIS are limited in their representational capacity.

# Chapter 3

# Semi-supervised Instance Segmentation

## 3.1 Introduction

From the results in the previous setting I hypothesize that the limiting factor in performance was the detector. Although some segmentation masks are low quality, particularly in the one-shot setting, as the number of shots increases the model is able to learn very accurate masks given an accurate detection. From this hypothesis a natural question arises: can accurate segmentation masks be predicted given abundant box and limited mask annotation? This is a question of importance for commercial applications, as box annotation is much cheaper to obtain than mask annotation. To explore this question and see how the techniques I introduced in the previous chapter can impact performance, I experimented with two settings: the traditional semi-supervised setting, where mask annotations are available for only a fraction of all box annotations for every class, and the zero shot transfer learning setting, where abundant box and mask annotations are available for a subset of classes and abundant box and 0 mask annotations are available for the other classes.

## 3.2 Related Work

The state-of-the-art method for semi-supervised instance segmentation is ShapeProp [13]. For the zero shot transfer learning setting it is able to achieve within 10% of the fully-supervised oracle's performance. For the traditional semi-supervised setting it is able to consistently achieve a higher segmentation AP than the baseline of Mask R-CNN for several possible fractions of mask annotation that have been made available to the model.

## 3.3 My Approach

As abundant box annotation is available in this setting there is no longer a need for negative proposal sampling, as its objective was to make the model more sensitive to instances of difficult-to-detect classes where there were few region proposals matched to these objects. Proposal relocation on the other hand may still be effective, so I introduce it exactly as discussed in the previous section on FSIS to see how effective it is in this setting.

## 3.4 Experiments

### 3.4.1 Zero Shot Transfer Learning Setting

| Method | Non-VOC Box AP | Non-VOC Segm AP | VOC Box AP | VOC Segm AP |
|---|---|---|---|---|
| ShapeProp | 35.635 | 34.722 | 41.791 | 34.521 |
| ShapeProp w/ PR | 35.881 | 34.808 | 41.929 | 34.598 |

Table 3.1: Non VOC → VOC Results

| Method | VOC Box AP | VOC Segm AP | Non-VOC Box AP | Non-VOC Segm AP |
|---|---|---|---|---|
| ShapeProp | 43.419 | 39.577 | 35.308 | 31.258 |
| ShapeProp w/ PR | 43.85 | 39.713 | 35.488 | 31.268 |

Table 3.2: VOC → Non VOC Results

I ran experiments in two zero shot transfer learning settings with the COCO dataset. In the first, there is abundant box and mask annotations for a subset of 60 classes known as the non-voc classes and abundant box and zero mask annotation for the other 20 classes, known as the voc classes. I denote this setting as non-voc → voc. In the second, there is abundant box and mask annotations for the 20 voc classes and abundant box and zero mask annotations for the 60 non-voc classes. I denote this setting as voc → non-voc. Table 3.1 shows the results for the non voc → voc setting, and Table 3.2 shows the results for the voc → non voc setting. In both settings proposal relocation does not produce a significant improvement in performance. This is likely the case for two reasons. The first is that when there is abundant box annotation for all classes box predictions become very accurate and thus additional relocation can only improve performance slightly. The second is that when there is zero mask annotation the shape activation, which is used for proposal relocation, becomes significantly less accurate. This reduces the efficacy of relocation, as relocation uses

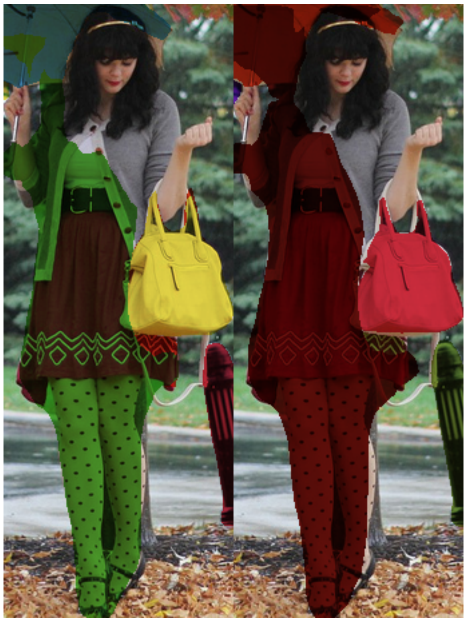

Figure 3.1: On the left is the mask prediction without proposal relocation and on the right is the prediction with proposal relocation.

the shape activation to relocate the detection. This is illustrated in Figure 3.1, where the original box prediction covers the whole object but the shape activation is of poor quality and thus proposal relocation is ineffective.

### 3.4.2 Traditional Semi-Supervised Instance Segmentation

| Method | Percent of Mask Annotations | Box AP | Segm AP |
|---|---|---|---|
| ShapeProp | 1% | 36.727 | 31.018 |
| ShapeProp w/ PR | 1% | 36.691 | 31.044 |
| ShapeProp | 2% | 35.998 | 31.414 |
| ShapeProp w/ PR | 2% | 36.201 | 31.47 |
| ShapeProp | 10% | 35.323 | 32.603 |
| ShapeProp w/ PR | 10% | 35.607 | 32.705 |

Table 3.3: Results for traditional semi-supervised experiments

For the traditional semi-supervised setting, I ran experiments where 1%, 2%, and 10% of box annotations had a corresponding mask annotation. The results for these experiments can be seen in Table 3.3. The improvement in performance in this setting is slightly more pronounced than in the zero shot transfer learning setting, likely due to the presence of mask annotation which made the shape activation accurate enough for effective relocation. However because the detections were all very accurate due to the abundant box annotation, proposal relocation still did not produce a significant improvement in performance in this setting.

## 3.5 Conclusion

The results of these experiments confirm my hypothesis that the limiting factor for model performance in FSIS is the detector. The significant improvement in performance in the zero shot transfer learning setting when compared to FSIS shows that given abundant box annotation, the model is able to detect instances of the zero shot classes and able to achieve a significantly higher segmentation AP than the FSIS model despite having zero mask annotations. Another conclusion that can be drawn from the experiments in both the traditional semi-supervised and zero shot transfer learning settings is that when abundant box annotation is available, proposal relocation does not significantly improve the box and segmentation AP of the model. The main reason for this is that with abundant box annotation the quality of the box prediction is very good and does not have much room for improvement. For the zero shot transfer learning setting the poor quality of the shape activation for zero shot classes is also a reason for why proposal relocation did not produce a significant improvement in performance.