One-Shot Instance Segmentation

Claudio Michaelis Ivan Ustyuzhaninov Matthias Bethge Alexander S. Ecker
University of Tübingen
claudio.michaelis@uni-tuebingen.de

Abstract

We tackle one-shot visual search by example for arbitrary object categories: Given an example image of a novel reference object, find and segment all object instances of the same category within a scene. To address this problem, we propose Siamese Mask R-CNN. It extends Mask R-CNN by a Siamese backbone encoding both reference image and scene, allowing it to target detection and segmentation towards the reference category. We use Siamese Mask R-CNN to perform one-shot instance segmentation on MS-COCO, demonstrating that it can detect and segment objects of novel categories it was not trained on, and without using mask annotations at test time. Our results highlight challenges of the one-shot setting: while transferring knowledge about instance segmentation to novel object categories not used during training works very well, targeting the detection and segmentation networks towards the reference category appears to be more difficult. Our work provides a first strong baseline for one-shot instance segmentation and will hopefully inspire further research in this relatively unexplored field.

1. Introduction

Humans do not only excel at acquiring novel concepts from a small number of training examples (*few-shot learning*), but can also readily point to such objects (*object detection*) and draw their outlines (*instance segmentation*). In recent years, machine vision has made substantial advances in one-shot learning [38, 79, 24] with a strong focus on image classification in a discriminative setting. Similarly, a lot of progress has been made on object detection and instance segmentation [29, 59], but both tasks are still very data-hungry and the proposed approaches perform well only for a small number of object classes, for which enough annotated examples are available.

In this paper, we work towards taking the one-shot setting to real-world instance segmentation: We learn to detect and segment arbitrary object categories (not necessarily included in the training set) based on a single visual example (Fig. 1). That is, given an arbitrary query image and a single reference instance, the goal is to generate a bounding box and an instance mask for every instance in the image that is of the same object category as the reference. This type of visual search task creates new challenges for computer vision algorithms, as methods from metric and few-shot learning have to be incorporated into the notoriously hard tasks of object identification and segmentation.

Figure 1. One-shot visual search. Given a query image and a reference image showing an object of a novel category, we seek to detect and segment all instances of the corresponding category ('person' on the left, 'car' on the right). Note that no ground truth annotations of reference categories are used during training.

Our approach is based on taking ideas from metric learning (*Siamese networks*) and combining them with Mask R-CNN, a state-of-the-art object detection and segmentation system (Fig. 2). Our main contributions are as follows:

- We present *Siamese Mask R-CNN* for performing one-shot instance segmentation. It extends Mask R-CNN [29] with a Siamese backbone and a matching procedure to perform visual search.
- We introduce a novel one-shot visual search task, requiring object detection and instance segmentation based on a single visual example.
- We establish an evaluation protocol for this task and evaluate our model on MS-COCO [44]. We show that segmenting novel object categories works well even without mask annotations at test time, while targeting the detection towards the reference category is the main challenge.
- We will make code and pre-trained models available.

2. Related work

Our approach lies at the intersection of few-shot/metric learning, object detection/visual search, and instance segmentation. Each of these aspects has been studied extensively, as we review in the following. The novelty of our approach is the combination of all these aspects into a new problem.

Object detection. Object detection is a classical computer vision problem [22, 31, 82, 4]. Modern work can be split broadly into two general approaches: Single stage detectors [47, 66, 67, 68, 43] are usually very fast, while multi-stage detectors [26, 25, 71, 29] perform a coarse proposal step followed by a fine-grained classification, and are usually more accurate. Most state-of-the-art systems are based on Faster R-CNN [71], a two-step object detector that generates proposals, for each of which it crops features out of the last feature map of a backbone. Feature Pyramid Networks [42] are a popular extension that uses feature maps at multiple spatial resolutions to increase scale invariance.

Instance segmentation. In contrast to semantic segmentation [49, 55, 73, 60, 90, 9, 15, 48], where every pixel is classified into a category, instance segmentation additionally requires to discriminate between individual object instances [27, 18, 28, 62, 19, 39, 63, 72, 5, 14, 23, 29, 45, 70, 37]. Most current state-of-the-art systems are based on Mask R-CNN [29, 46, 1], an extension of Faster R-CNN [71] performing joint object detection and instance segmentation.

Weakly supervised object detection and segmentation. Labeled data is hard to obtain for instance-level tasks like object detection, and even more so for pixel-level tasks like segmentation [44, 12, 3]. Therefore, various weakly and semi-supervised approaches have been explored [32, 88, 57, 35, 92]. Weak supervision is a promising direction for annotation-heavy tasks, hence it has been explored for semantic segmentation [58, 57, 61, 17, 88, 7, 41], object detection [56, 91, 67] and instance segmentation [35, 33, 92].

Visual search. Visual search has a long history in perceptual psychology (reviewed, *e.g.*, by [75]), although typically with simple visual patterns, while search for arbitrary objects in real scenes has been addressed only recently [89, 87], and often using a natural language cue [87].

Few-shot learning. Few-Shot learning has seen great progress over the last years. A classic approach is based on metric learning using Siamese neural networks [8, 16, 36], which – due to its simplicity – is also the approach we use. The metric learning approach has seen a number of improvements in recent years [36, 84, 79, 85, 86]. Other approaches are based on generative models [38, 76], ideas from information retrieval [81] or employ meta learning [24, 40, 52, 51, 53, 54, 74, 80, 69].

Few-shot segmentation. Closely related to our work is one-shot semantic segmentation of images using either an object instance as reference [78, 65, 20, 50] or a texture [83]. However, the key difference is that these systems perform pixel-level classifications and cannot distinguish individual instances. The only work on one-shot instance segmentation we are aware of tracks an object instance across a video sequence based on a small number of annotated frames [10, 11], which differs from our setup in that a single object is to be tracked, for which ground-truth annotations are available.

Few-shot object detection. There is related, but not directly comparable work on few-shot object detection. Some work focuses on settings with few (more than one) annotated training images per category [13, 21], while others tackle the zero-shot setting based on only a textual description of the reference [6, 64]. Most closely related to our work is concurrent work based on Siamese networks for one-shot detection on an Omniglot-based dataset and for audio data [34] as well as work on fine-grained bird classification and localization in ImageNet images [77], which tend to have only one or few instances per image. In contrast, we work on potentially cluttered real-world images.

3. One-shot object detection and instance segmentation on MS-COCO

We define a one-shot object detection and instance segmentation task on MS-COCO: Given a *reference image* showing a close-up of an example object, find all instances of objects belonging to the same category in a separate *query* image, which shows an entire visual scene potentially containing many objects. To work in a one-shot setting, we split the 80 object categories in MS-COCO into *background* and one-shot *evaluation* splits[1], containing 60 and 20 categories, respectively. We generate four such background/evaluation splits by starting with the first, second, third or fourth category, respectively, and including every fourth category into the one-shot evaluation split. We call those splits S_1–S_4; they are given in Table 3 in the Appendix.

Note that this one-shot visual search setup differs from earlier, purely discriminative one-shot learning setups: At training time, the query images can contain objects from the one-shot evaluation categories, but they are neither selected as the reference nor are they annotated in any way.

[1] Following the terminology of Lake *et al.* [38].

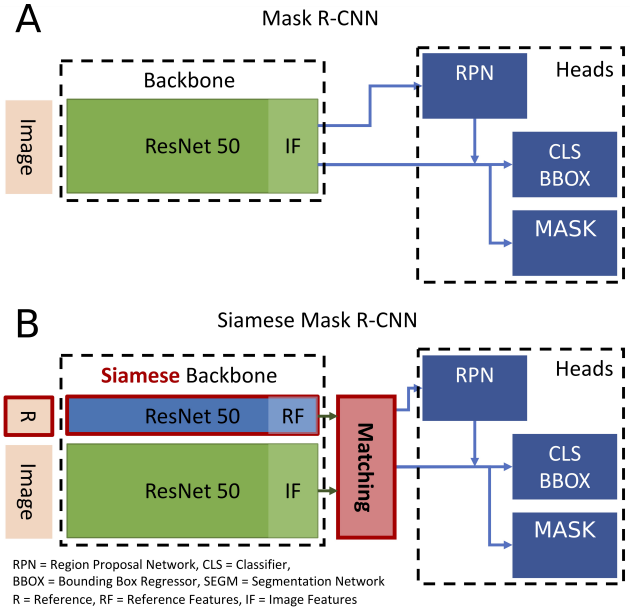


Figure 2. Comparison of Mask R-CNN (A) and Siamese Mask R-CNN (B). The differences between the two models are the addition of a Siamese backbone which encodes the reference and the matching step in the Siamese model (marked in red).

We therefore still refer to this setting as one-shot, because no label information is available for these categories during training. Conversely, at test time, the query images contain both known and novel object categories. Taken together, we consider this setup to be a realistic scenario in the real world of an autonomous agent, which would typically encounter new objects alongside the known objects and may encounter unlabeled objects multiple times before they become relevant and label information is provided (think of a household robot seeing a certain type of toy in various parts of the apartment multiple times before you instruct it to go pick it up for you). This setup also produces a number of challenges for evaluation, which we discuss in Section 5.2.

4. Siamese Mask R-CNN

The key idea behind Siamese Mask R-CNN is to detect and segment object instances based on a single visual example of some object category. Thus, it must deal with arbitrary, potentially previously unseen object categories, rather than with a fixed set of categories. We base Siamese Mask R-CNN on Mask R-CNN [29] with feature pyramid networks [42]. To adapt it to the visual search task, we turn the backbone into a Siamese network – hence the prefix *Siamese* –, which extracts features from both the reference image and the scene and computes a pixel-wise similarity between the two. The image features and the similarity score form the input to three heads: (1) the Region Proposal Network (RPN), (2) the bounding box classification and regression head and (3) the segmentation head. In the following, we briefly review the key components of Mask R-CNN and then introduce our extensions.

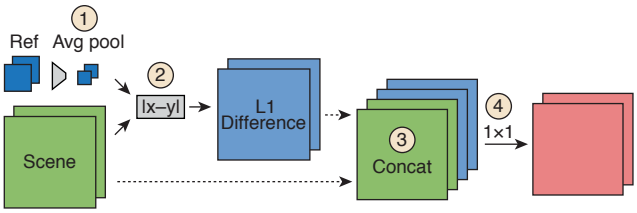


Figure 3. Sketch of the matching procedure. The reference encoding is reduced to a vector by average pooling (1) and the point by point absolute difference to the scene encoding is computed (2). The concatenated (3) scene encoding and reference features are reduced by a 1×1 convolution (4) before feeding them to the network heads.

4.1. Mask R-CNN

Mask R-CNN is a two-stage object detector that consists of a backbone feature extractor and multiple heads operating on these features (see Fig. 2A). We choose a ResNet50 [30] with Feature Pyramid Networks (FPN) [42] as our backbone. The heads consist of two stages. First, the region proposal network (RPN) is applied convolutionally across the image to predict possible object locations in the scene. The highest scoring region proposals are then cropped from the backbone feature maps and used as inputs for the bounding box classification (CLS) and regression (BBOX) head as well as the instance masking head (MASK).

4.2. Siamese feature pyramid networks

In the conventional object detection/instance segmentation setting, the set of possible categories is known in advance, so the task of the backbone is to extract useful features for the subsequent detection and segmentation stages. In contrast, in the one-shot setting the information on which objects to detect and segment is provided in the form of a reference image, which can contain an object category the system has not been trained on. To adapt to this situation, our backbone does not only extract useful features from the scene image, but also computes a similarity metric to the reference at each possible location. To do so, we follow the basic idea of Siamese networks [36] and apply the same backbone (ResNet50 with FPN) with shared weights to extract features from both the reference and the scene. These features are then matched pixel-wise as described below.

4.3. Feature matching

The feature pyramid network produces image features at multiple scales, hence we perform the following matching procedure at each scale of the pyramid (Fig. 3):

1. Pool the features of the reference image over space using average pooling to obtain a vector embedding of the category to be detected and segmented.
2. At every spatial position of the scene representation, compute the absolute difference between the features of the reference and that of the scene.
3. Concatenate the scene representation and the pixel-wise distance between the two.
4. Reduce the number of features by 1×1 convolution.

The resulting features are then used as a drop-in replacement for the original feature pyramid as they have the same dimensionality. The key difference is that they do not only encode the content of the scene image, but also its similarity to the reference image, which forms the basis for the subsequent heads to generate object proposals, classify matches vs. non-matches and generate instance masks.

4.4. Head architecture

We use the same region proposal network (RPN) as Mask R-CNN, changing only its inputs as described above and the way examples are generated during training (described below). We also use the same classification and bounding box regression head as Mask R-CNN, but change the classification from an 80-way class discrimination to a binary match/non-match discrimination. Similarly, for the mask branch we generate only a single instance mask instead of one per potential class.

4.5. Implementation details

Our system is based on the Matterport implementation of Mask R-CNN [2]. We provide all details in Appendix 1.

5. Experiments

We train Siamese Mask R-CNN jointly on object detection and instance segmentation in the visual search setting. We evaluate the trained models both on previously seen and unseen (one-shot) categories using splits of MS-COCO.

5.1. Training

Pre-training backbone. We pre-train the ResNet backbone on image classification on a reduced subset of ImageNet, which contains images from the 687 ImageNet categories without correspondence in MS-COCO – hence we refer to it as *ImageNet-687*. Pre-training on this reduced set ensures that we do not use any label information about the one-shot classes at any training stage.

Training Siamese Mask R-CNN. We train the models using stochastic gradient descent with momentum for 160,000 steps with a batch size of 12 on four NVIDIA P100 GPUs in parallel. We use an initial learning rate of 0.02 and a momentum of 0.9. During the first 1,000 steps, we train only the heads. After that, we train the entire network, including the backbone and all heads, end-to-end. After 120,000 steps, we divide the learning rate by 10.

Construction of mini-batches. During training, a mini-batch contains 12 sets of reference and query images. We first draw the query images at random from the training set and pre-process them in the following way: (1) we resize an image so that the longer side is 1024 px, while keeping the aspect ratio, (2) we zero-pad the smaller side of the image to be square 1024×1024, (3) we subtract the mean ImageNet RGB value from each pixel. Next, for each image, we generate a reference image as follows: (1) draw a random category among all categories of the background set present in the image, (2) crop a random instance of the selected category out of any image in the training set (using the bounding box annotation), and (3) resize the reference image so that its longer side is 192 px and zero-pad the shorter side to get a square image of 192×192. To enable a quick look-up of reference instances, we created an index that contains a list of categories present in each image.

Labels. We use only the annotations of object instances in the query image that belong to the corresponding reference category. All other objects are treated as background.

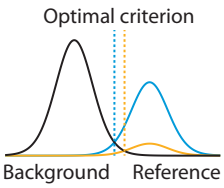
Loss function. Siamese Mask R-CNN is trained on the same basic multi-task objective as Mask R-CNN: classification and bounding box loss for the RPN; classification, bounding box and mask loss for each RoI. There are a couple of differences as well. First, the classification losses consist of a binary cross-entropy of the match/non-match classification rather than an 80-way multinomial cross-entropy used for classification on MS-COCO. Second, we found that weighting the individual losses differently improved performance in the one-shot setting. Specifically, we apply the following weights to each component of the loss function: RPN classification loss: 2, RPN bounding box loss: 0.1, RoI classification loss: 2, RoI bounding box loss: 0.5 and mask loss: 1.

Mask R-CNN. For comparison, we also trained the original Mask R-CNN on MS-COCO on all 80 classes for 320,000 steps using the same hyper parameters as for Siamese Mask R-CNN but without the adjustments to the loss function weights described above.

5.2. Evaluation

General procedure. We evaluate the performance of our model using the MS-COCO *val 2017* set as a test set (it was not used for training). We do one evaluation run per class split S, using the following procedure:

Figure 4. Object scores can be thought of as posterior probabilities, i.e. the product of image evidence and category prior. Thus, the optimal criterion depends on the prior, but in a one-shot setting, there is no information about the prior.



Model	Obj. detection mAP50	Instance segm. mAP50
Mask R-CNN	42.5	40.1
Siamese Mask R-CNN	35.7	33.4

Table 1. Detection results on MS-COCO *val 2017*.

1. For each image in the test set and each one-shot category present in this image, extract a reference instance from another randomly chosen image in the test set.
2. For each (query, reference) image pair, compute predictions for bounding boxes and segmentation masks.
3. Assign the computed predictions to the category of the corresponding reference image (that allows us to use standard tools for MS-COCO evaluation),
4. Aggregating the predictions for all images, compute the AP50 value for each category in S, and obtain a mAP50 score by averaging the AP50 values over all categories in S.

The class splits S are either one of the four one-shot splits S_1–S_4 (one-shot evaluation) or the entire set of training categories (for comparison to regular Mask R-CNN).

Considerations for evaluation. Our evaluation scheme is similar to the standard evaluation of instance segmentation models on MS-COCO, allowing us to use existing tools for evaluation. However, the resulting mAP50 values are not directly comparable to earlier work on fixed-category detection and segmentation setups. The main difference is the way in which we select the reference images. We ensure that there is always at least one object of the reference category in the query image. The primary reason why we enforce this constraint is to simplify the task. The one-shot visual search task has two aspects that make it substantially harder than detection in a fixed-category setting or one-shot learning in a discriminative setting.

First, to perform one-shot learning in a discriminative setting, one does not need to normalize the scores in any way; one can simply pick the largest. In contrast, in the detection setting, we do not know a priori how many instances there are, so the scores need to be normalized such that applying the same threshold on the confidence scores across images actually makes sense.

Second, we can think of the scores for each object as a posterior, i.e. the product of the image evidence for the category and the prior probability of the category being present in an image (blue vs. orange in Fig. 4). However, in a one-shot setting, there is no information about the prior, so one would have to guess it for each novel object category.

Thus, to simplify the task and to keep the prior for each category roughly constant, we decided to change the evaluation in the way described above. As we show below, this task is still hard for systems that perform competitively on regular MS-COCO detection and instance segmentation, so we think it makes sense to use these simplifications in order to work in a regime where progress is realistic.

5.3. Baseline: random boxes

As a very naïve baseline, we evaluate the performance of a model predicting random bounding boxes and segmentation masks. To do so, we take ground-truth bounding boxes and segmentation masks for the category of the reference image, and randomly shift the boxes around the image (assigning a random confidence value for each box between 0.8 and 1). We keep the ground-truth segmentation masks intact in the shifted boxes. Such procedure allows us to get random predictions while keeping certain statistics of the ground-truth annotations (*e.g.* number of boxes per image, their sizes, etc.).

6. Results

6.1. Example-based detection and segmentation

We start by showing our results on the task of object detection and instance segmentation targeted to a single class, which is given by an example. This is essentially a metric learning problem: we learn a similarity metric between image regions and the reference image. This allows the detection and segmentation heads to produce bounding boxes and instance masks for matching objects. As discussed above, this problem is harder than training an object detector for a fixed set of classes, and we therefore simplified the training and evaluation process (see Section 5.2 above).

To put our one-shot results reported below in context, we first trained both Siamese Mask R-CNN as well regular Mask R-CNN on the entire MS-COCO data set (Table 1). Our Mask R-CNN implementation performed reasonably, achieving 42.5% mAP50 on detection and 40.1% on instance segmentation. These numbers are not state-of-the-art (due to limited availability of extendable code and pre-trained models), but that doesn't change the conclusions, since we are interested in relative performance differences to Mask R-CNN and not in absolute values.

Siamese Mask R-CNN achieved 35.7% mAP on detection and 33.4% on instance segmentation using the same backbone, training schedule, etc., but based on examples rather than trained on a fixed set of categories. Thus, we conclude that the proposed Siamese Mask R-CNN architec-

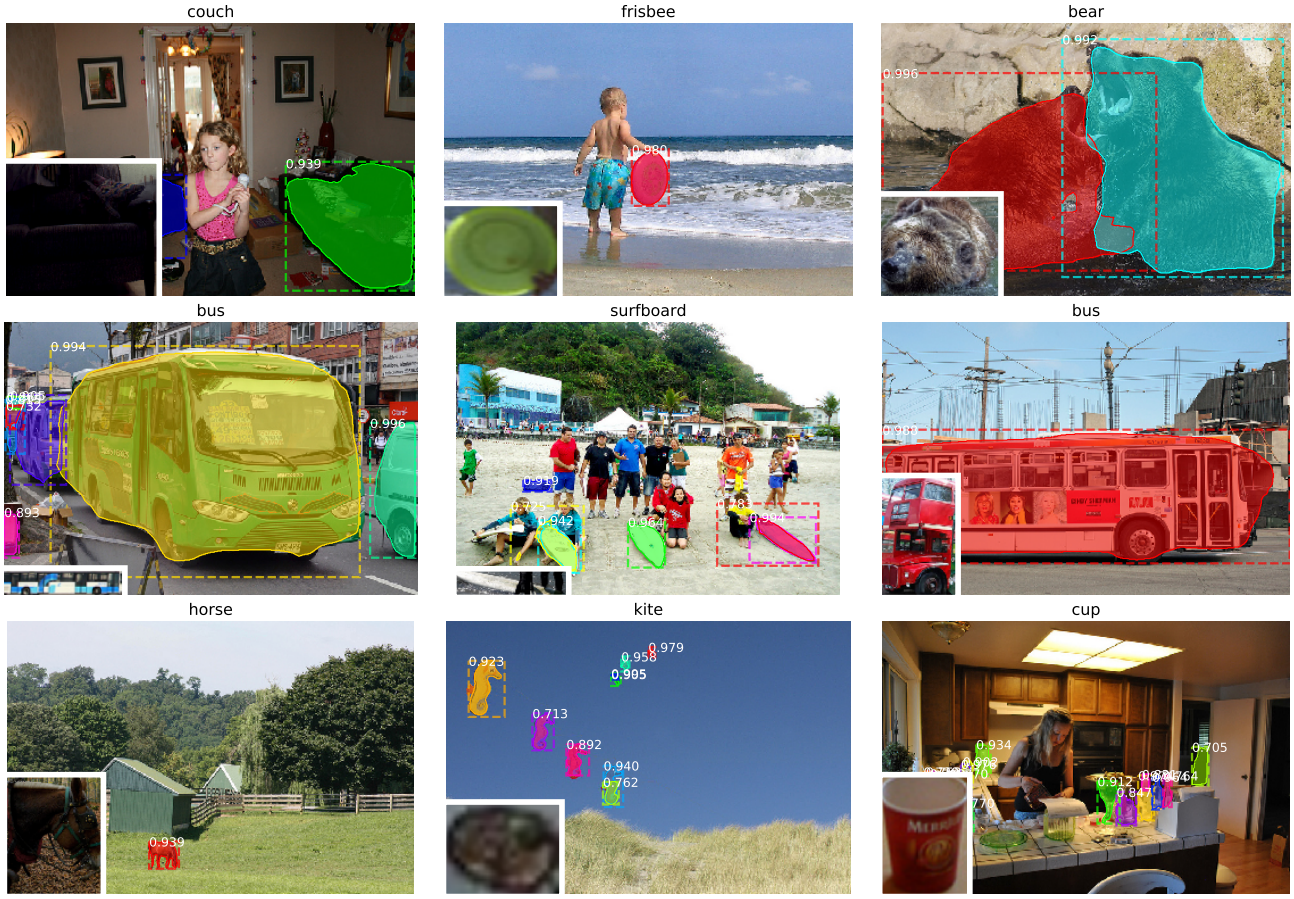


Figure 5. Examples of Siamese Mask R-CNN operating in the one-shot setting, i.e. segmenting object for which no annotations were used using training (split S_2). Reference images are shown in the lower-left corner and the target categories are in the titles (these categories are just for the reader and are not used anywhere in the system).

ture can learn object detection and instance segmentation based on examples, but there is room for improvement, suggesting that the example-based setting is more challenging.

6.2. One-shot instance segmentation

Next, we report the results of evaluating Siamese Mask R-CNN in the one-shot setting. That is, we train on the background splits without using instances of one-shot evaluation splits (Section 3) as reference images. These results are shown in Table 2. The average detection mAP50 scores for the one-shot splits are around 17%, while the segmentation ones are around 15%, with some variability between splits. These values are significantly lower than those for the background splits, indicating the difficulty of the one-shot setting. The mAP50 scores for the background splits are slightly higher than those in Table 1, because the former contain only 60 categories while the latter were trained on all 80. Taken together, these results suggest that we observe a substantial degree of overfitting on the background classes used during training. This result is in contrast to earlier work on Omniglot [50] that observed good generalization beyond the background set, presumably because Omniglot contains a larger number of categories and the image statistics are simpler.

Object detection

Split	1	2	3	4	Average
Background	38.9	37.1	37.8	36.6	37.6
One-shot	15.3	17.6	17.4	17.0	16.8
Random boxes	2.3	2.0	1.7	2.7	2.2

Instance segmentation

Split	1	2	3	4	Average
Background	36.6	33.7	35.1	33.9	34.8
One-shot	13.2	15.4	16.3	14.7	14.9
Random masks	1.2	1.0	1.0	1.2	1.1

Table 2. Results on MS Coco (in % mAP50). In split i, every fourth class, starting at the i^{th}, is placed into the one-shot set.

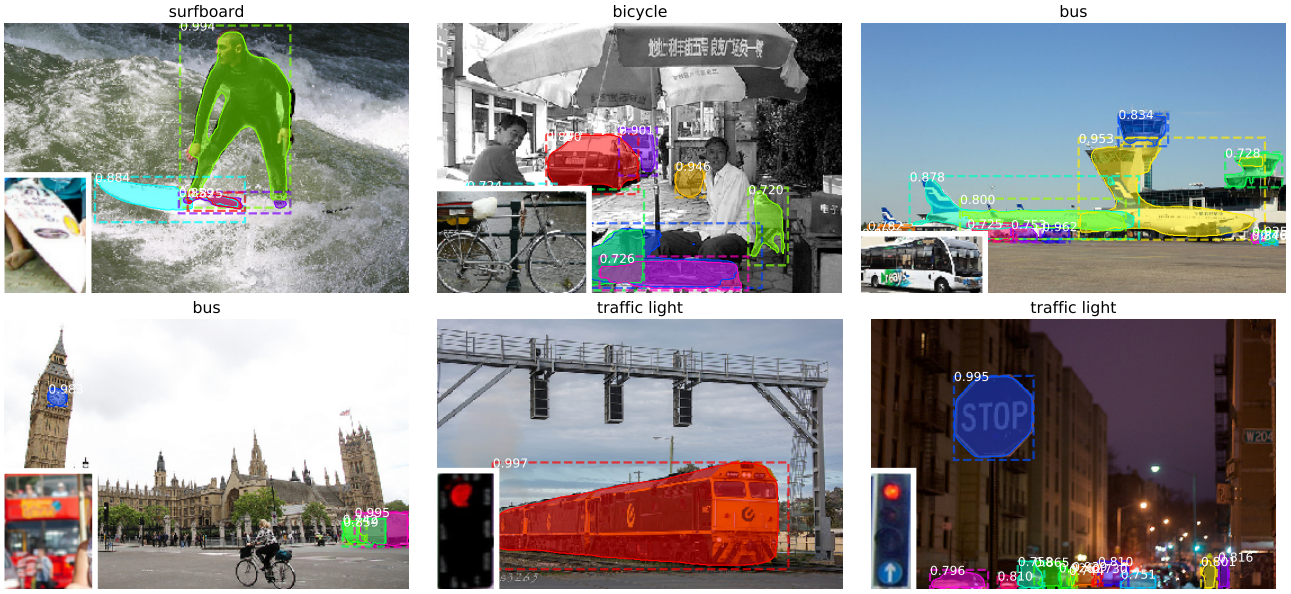


Figure 6. Examples of Siamese Mask R-CNN failure cases. False positives are a common problem for our model and we show examples of categories such as person, car, plane, clock, train and street sign being falsely predicted. These categories are among the most commonly falsely predicted categories (see Fig. 7).

6.3. Qualitative analysis

Figure 5 shows examples of successful Siamese Mask R-CNN predictions for one-shot categories (i.e. categories not used during training). These examples allow us to get a feeling for the difficulty of the task: the reference inputs are quite different from the instances in the query image, sometimes they show only part of the reference object and they are never annotated with ground truth segmentation masks. To generate bounding boxes and segmentation masks, the model can use only its general knowledge about objects and their boundaries and the metric learned on the other categories to compute the visual similarity between the reference and the query instances. For instance, the bus on the right or the horse in the bottom left in Figure 5 are incomplete and the network has never been provided with ground truth bounding boxes or instance masks for either horses or buses. Nevertheless, it still finds the correct object in the query image and segments the entire object.

We also show examples of failure cases in Figure 6. The picture that emerges from both successful and failure cases is that the network produces overall very good bounding boxes and segmentation masks, but often fails at targeting it towards the correct category. We elaborate more in the next section on the challenges of the one-shot setting.

6.4. False positives in the one-shot setting

There is a marked drop in model performance between the background and the one-shot evaluation splits, suggesting some degree of overfitting to the background categories

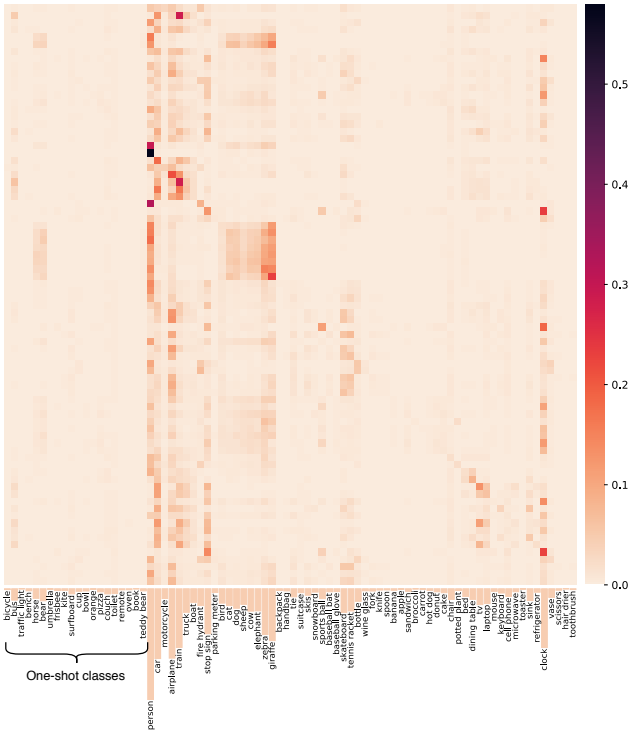


Figure 7. Confusion matrix for the Siamese Mask R-CNN model using split S_2 for one-shot evaluation. The element (i, j) shows the AP50 of using detections for category i and evaluating them as instances of category j. The histogram below the matrix shows the most commonly confused (or falsely predicted) categories.

used during training. If overfitting to background classes was indeed the main issue, we would expect false positives to be biased towards these categories and, in particular, towards those categories that are most frequent in the training set. This seems to be qualitatively the case (Fig. 5). In addition, we quantified this observation by computing a confusion matrix of MS-COCO categories (Fig. 7). The element (i, j) of this matrix corresponds to the AP50 value of detections obtained for reference images of category i, which are evaluated as if the reference images belonged to category j. If there were no false positives, the off-diagonal elements of the matrix would be zero. The sums of values in the columns show instances of categories that are most often falsely detected (the histogram of such sums is shown below the matrix). Among such commonly falsely predicted categories are people, cars, airplanes, clocks, and other categories that are common in the dataset.

6.5. Effect of image clutter

Previous work on synthetic data [50] found that cluttered scenes are especially challenging in the one-shot setting. This effect is also present in the current context. Both detection and segmentation scores are substantially higher when conditioning on images with a small number of total instances (Figure 8), underscoring the importance of extending the model to robustly process cluttered scenes.

7. Discussion

We introduced the task of one-shot instance segmentation and proposed a model based on combining the Mask R-CNN architecture with a metric learning approach to perform this task. There are two main problems in this task: (1) learning a good metric for one-shot detection of novel objects and (2) transferring the knowledge about bounding boxes and instance masks from known to novel object categories. Our results suggest that in the context of MS-COCO, the first part is more difficult than the second part. Overall, bounding boxes and instance masks are of high quality. The relatively weak performance of our current model appears to be caused by its difficulties in classifying if the detected object is of the same category as the reference. Our observation of a substantial amount of overfitting towards the categories used during training supports this hypothesis.

Our system is not based on the latest and highest-performing object detector, but was rather driven by availability of code for existing approaches; we expect that incorporating better object detection architectures and larger backbones into our one-shot visual search framework will lead to performance improvements analogous to those reported on the fixed-category problem. However, closing the gap between the fixed-category and the one-shot visual search problems would likely require not just better components for our model, but rather conceptual changes to the model itself and to the training data. Such changes might include larger datasets with more object categories than MS-COCO or more sophisticated approaches to one-shot learning from a relatively small number of background categories.

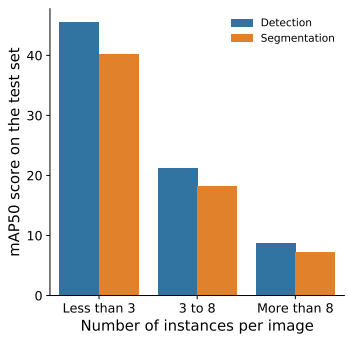


Figure 8. One-shot mAP50 scores for the split S_2 for test images with different numbers of instances per image.

There are a couple of drawbacks to our current approach, and resolving them is likely to lead to improvements in performance. For instance, during training we currently treat all instances of the one-shot categories as background, which probably encourages the model to suppress their detection even if they match the reference well. In addition, the reference instances are sometimes hard to recognize even for humans, because they are cropped to their bounding box and lack image context, which can be an important cue for recognition. Finally, the system currently relies exclusively on comparing each object proposal to the reference image and performing a match/non-match discrimination. However, one may instead want to do an N+1-way classification, assigning each instance to one of the N already known categories or a novel, N+1$^{\text{st}}$ one, and only in the latter case rely on a similarity metric and a binary match/non-match classification.

In summary, one-shot instance segmentation is a hard problem on a diverse real-world dataset like MS-COCO. It requires combining ideas from few-shot/metric learning, object detection and segmentation, and we believe it is a perfect test bed for developing truly general vision systems.

Acknowledgements

This work was supported by the German Research Foundation (DFG) through Collaborative Research Center (CRC 1233) "Robust Vision" and DFG grant EC 479/1-1 (to A.S.E.), by the German Federal Ministry of Education and Research through the Tbingen AI Center (FKZ 01IS18039A), by the International Max Planck Research School for Intelligent Systems (C.M. and I.U.), and by the Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Govern-

mental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

References

[1] Coco detection leaderboard. http://cocodataset.org/, Aug. 2018.

[2] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[3] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient Interactive Annotation of Segmentation Datasets With Polygon-RNN++. In *CVPR*, page 10, 2018.

[4] P. Arbelez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014.

[5] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, pages 2858–2866. IEEE, 2017.

[6] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran. Zero-Shot Object Detection. *arXiv:1804.04340 [cs]*, Apr. 2018. arXiv: 1804.04340.

[7] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. Whats the Point: Semantic Segmentation with Point Supervision. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *ECCV*, Lecture Notes in Computer Science, pages 549–565. Springer International Publishing, 2016.

[8] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Sckinger, and R. Shah. Signature Verification Using A "Siamese" Time Delay Neural Network. *IJPRAI*, 7(4):669–688, 1993.

[9] S. R. Bul, L. Porzi, and P. Kontschieder. In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. In *CVPR*, 2018. arXiv: 1712.02616.

[10] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taix, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*. IEEE, 2017.

[11] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 DAVIS Challenge on Video Object Segmentation. *arXiv:1803.00557 [cs]*, Mar. 2018. arXiv: 1803.00557.

[12] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating Object Instances with a Polygon-RNN. In *CVPR*, pages 4485–4493. IEEE, July 2017.

[13] H. Chen, Y. Wang, G. Wang, and Y. Qiao. LSTD: A Low-Shot Transfer Detector for Object Detection. *arXiv:1803.01529 [cs]*, Mar. 2018. arXiv: 1803.01529.

[14] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam. MaskLab: Instance Segmentation by Refining Object Detection With Semantic and Direction Features. In *CVPR*, page 10, 2017.

[15] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI*, PP(99):1–1, 2018.

[16] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pages 539–546. IEEE, 2005.

[17] J. Dai, K. He, and J. Sun. BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. In *ICCV*, Mar. 2015. arXiv: 1503.01640.

[18] J. Dai, K. He, and J. Sun. Convolutional Feature Masking for Joint Object and Stuff Segmentation. In *CVPR*, pages 3992–4000, June 2015. arXiv: 1412.1283.

[19] J. Dai, K. He, and J. Sun. Instance-aware Semantic Segmentation via Multi-task Network Cascades. In *CVPR*, 2016. arXiv: 1512.04412.

[20] N. Dong and E. P. Xing. Few-Shot Semantic Segmentation with Prototype Learning. In *BMVC*, page 13, 2018.

[21] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng. Few-Example Object Detection with Model Communication. *TPAMI*, pages 1–1, 2018.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[23] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic Instance Segmentation via Deep Metric Learning. *arXiv:1703.10277 [cs]*, Mar. 2017. arXiv: 1703.10277.

[24] C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, pages 1126–1135, July 2017.

[25] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, pages 580–587, June 2014.

[27] B. Hariharan, P. Arbelez, R. Girshick, and J. Malik. Simultaneous Detection and Segmentation. In *ECCV*, July 2014. arXiv: 1407.1808.

[28] B. Hariharan, P. Arbelez, R. Girshick, and J. Malik. Hypercolumns for Object Segmentation and Fine-grained Localization. In *CVPR*, 2015. arXiv: 1411.5752.

[29] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, Oct. 2017.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[31] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. *ECCV*, pages 340–353, 2012.

[32] S. Hong, H. Noh, and B. Han. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. In *NIPS*, June 2015. arXiv: 1506.04924.

[33] R. Hu, P. Dollr, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *CVPR*, 2018.

[34] G. Keren, M. Schmitt, T. Kehrenberg, and B. Schuller. Weakly Supervised One-Shot Detection with Attention Siamese Networks. *arXiv:1801.03329 [cs, stat]*, Jan. 2018. arXiv: 1801.03329.

[35] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In *CVPR*, 2017. arXiv: 1603.07485.

[36] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition - oneshot1.pdf. *ICML*, 2015.

[37] S. Kong and C. Fowlkes. Recurrent Pixel Embedding for Instance Grouping. In *CVPR*, 2018. arXiv: 1712.08273.

[38] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, Dec. 2015.

[39] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *CVPR*, pages 3659–3667, 2016.

[40] Z. Li, F. Zhou, F. Chen, and H. Li. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv:1707.09835 [cs]*, July 2017. arXiv: 1707.09835.

[41] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation. In *CVPR*, Apr. 2016. arXiv: 1604.05144.

[42] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, pages 936–944, July 2017.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal Loss for Dense Object Detection. *ICCV*, Aug. 2017. arXiv: 1708.02002.

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, Lecture Notes in Computer Science, pages 740–755. Springer, Cham, Sept. 2014.

[45] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017.

[46] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path Aggregation Network for Instance Segmentation. *arXiv:1803.01534 [cs]*, Mar. 2018. arXiv: 1803.01534.

[47] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, Lecture Notes in Computer Science, pages 21–37. Springer, Cham, Oct. 2016.

[48] X. Liu, Z. Deng, and Y. Yang. Recent progress in semantic image segmentation. *Artificial Intelligence Review*, pages 1–18, June 2018.

[49] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[50] C. Michaelis, M. Bethge, and A. Ecker. One-Shot Segmentation in Clutter. In *ICML*, pages 3546–3555, July 2018.

[51] T. Munkhdalai and A. Trischler. Metalearning with Hebbian Fast Weights. *arXiv:1807.05076 [cs, stat]*, July 2018. arXiv: 1807.05076.

[52] T. Munkhdalai and H. Yu. Meta Networks. In *International Conference on Machine Learning*, pages 2554–2563, July 2017.

[53] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler. Rapid Adaptation with Conditionally Shifted Neurons. In *International Conference on Machine Learning*, pages 3664–3673, July 2018.

[54] A. Nichol, J. Achiam, and J. Schulman. On First-Order Meta-Learning Algorithms. *arXiv:1803.02999 [cs]*, Mar. 2018. arXiv: 1803.02999.

[55] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[56] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015.

[57] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation. In *ICCV*, Feb. 2015. arXiv: 1502.02734.

[58] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Multi-Class Multiple Instance Learning. In *ICLR*, 2015. arXiv: 1412.7144.

[59] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. *arXiv preprint arXiv:1711.07240*, 7, 2017.

[60] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network. *arXiv:1703.02719 [cs]*, Mar. 2017. arXiv: 1703.02719.

[61] P. O. Pinheiro and R. Collobert. From Image-level to Pixel-level Labeling with Convolutional Networks. In *CVPR*, volume CVPR, 2015. arXiv: 1411.6228.

[62] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to Segment Object Candidates. In *NIPS*, June 2015. arXiv: 1506.06204.

[63] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr. Learning to Refine Object Segments. In *ECCV*, Mar. 2016. arXiv: 1603.08695.

[64] S. Rahman, S. Khan, and F. Porikli. Zero-Shot Object Detection: Learning to Simultaneously Recognize and Localize Novel Concepts. *arXiv:1803.06049 [cs]*, Mar. 2018. arXiv: 1803.06049.

[65] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine. Conditional Networks for Few-Shot Semantic Segmentation. *arXiv:1806.07373 [cs, stat]*, Feb. 2018.

[66] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, pages 779–788, June 2016.

[67] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *CVPR*, pages 6517–6525, July 2017.

[68] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv:1804.02767 [cs]*, Apr. 2018. arXiv: 1804.02767.

[69] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-Learning for Semi-Supervised Few-Shot Classification. In *ICLR*, Mar. 2018. arXiv: 1803.00676.

[70] M. Ren and R. S. Zemel. End-to-End Instance Segmentation with Recurrent Attention. In *CVPR*, pages 293–301, July 2017.

[71] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[72] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *ECCV*, pages 312–329. Springer, 2016.

[73] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, Lecture Notes in Computer Science, pages 234–241. Springer, Cham, Oct. 2015.

[74] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-Learning with Latent Embedding Optimization. *arXiv:1807.05960 [cs, stat]*, July 2018. arXiv: 1807.05960.

[75] A. F. Sanders and M. Donk. Visual search. In *Handbook of perception and action*, volume 3, pages 43–77. Elsevier, 1996.

[76] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, R. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *arXiv:1806.04734 [cs]*, June 2018. arXiv: 1806.04734.

[77] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. RepMet: Representative-based metric learning for classification and one-shot object detection. *arXiv:1806.04728 [cs]*, June 2018. arXiv: 1806.04728.

[78] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-Shot Learning for Semantic Segmentation. *BMVC*, 2017.

[79] J. Snell, K. Swersky, and R. Zemel. Prototypical Networks for Few-shot Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NIPS*, pages 4080–4090. Curran Associates, Inc., 2017.

[80] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *CVPR*, page 10, 2018.

[81] E. Triantafillou, R. Zemel, and R. Urtasun. Few-Shot Learning Through an Information Retrieval Lens. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NIPS*, pages 2252–2262. Curran Associates, Inc., 2017.

[82] J. R. R. Uijlings, K. E. A. v. d. Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, Sept. 2013.

[83] I. Ustyuzhaninov, C. Michaelis, W. Brendel, and M. Bethge. One-shot Texture Segmentation. *arXiv:1807.02654 [cs]*, July 2018. arXiv: 1807.02654.

[84] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and others. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[85] P. Wang, L. Liu, C. Shen, Z. Huang, A. v. d. Hengel, and H. T. Shen. Multi-attention Network for One Shot Learning. In *CVPR*, pages 6212–6220. IEEE, July 2017.

[86] Y. Wang, X.-M. Wu, Q. Li, J. Gu, W. Xiang, L. Zhang, and V. O. K. Li. Large Margin Few-Shot Learning. *arXiv:1807.02872 [cs, stat]*, July 2018. arXiv: 1807.02872.

[87] J. M. Wolfe, G. A. Alvarez, R. Rosenholtz, Y. I. Kuzmova, and A. M. Sherman. Visual search for arbitrary objects in real scenes. *Attention, perception & psychophysics*, 73(6):1650–1671, Aug. 2011.

[88] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, pages 3781–3790, June 2015.

[89] H. Yang and G. J. Zelinsky. Visual search is guided to categorically-defined targets. *Vision Research*, 49(16):2095–2103, Aug. 2009.

[90] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[91] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, pages 2921–2929, 2016.

[92] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao. Weakly Supervised Instance Segmentation Using Class Peak Response. In *CVPR*, page 10, 2018.

Appendix

1. Implementation details

1.1. Backbone

We use the standard architecture of ResNet-50 [30] without any modifications.

1.2. Feature matching

- We use layers[2] `res2c_relu` (256 features), `res3d_relu` (512), `res4f_relu` (1024) and `res5c_relu` (2048) of the backbone as a feature representation of the inputs. For brevity, we refer to these layers as C_2, C_3, C_4 and C_5.
- FPN generates multi-scale representations P_i, $i = \{2, 3, 4, 5, 6\}$ consisting of 256 features (for all i) as follows. P_5 is a result of applying a 1×1 conv layer to C_5 (to get 256 features). P_i ($i = \{2, 3, 4\}$) is a sum of a 1×1 conv layer applied to C_i and up-sampled (by a factor of two on each side) P_{i+1}. P_6 is a down-sampled P_5 (by a factor of two on each side).
- The final similarity scores between the input scene and the reference at scale i are computed by obtaining P_i^{scene} and P_i^{ref} as described above, applying global average pooling to P_i^{ref}, and computing pixel-wise differences $D_i = \text{abs}(P_i^{\text{scene}} - \text{pool}(P_i^{\text{ref}}))$.
- The final feature representations containing information about similarities between the scene and the reference are computed by concatenating P_i^{scene} and D_i, and applying a 1×1 conv layer, outputting 384 features.

1.3. Region Proposal Network (RPN)

- We use 3 anchor aspect ratios (0.5, 1, 2) at each pixel location for the 5 scales (32, 64, 128, 256, 512) $i = \{2, \ldots, 6\}$ defined above, resulting in $3 \times (32^2 + \ldots + 512^2) \approx$ 1M proposals in total.
- The architecture is a $3 \times 3 \times 512$ conv layer, followed by the 1×1 conv outputting k times number of anchors per location (three in our case) features (corresponding to proposal logits for $k = 2$ or to bounding box deltas for $k = 4$).

1.4. Classification and bounding box regression head

The classification head produces same/different classifications for each proposal and performs bounding box regression.

- Inputs: the computed bounding boxes (outputs of the RPN) are cropped from P_i, reshaped to 7×7, and concatenated for $i = \{2, \ldots, 5\}$. Only 6000 top scoring anchors are processed for efficiency.
- Architecture: two fc-layers (1024 units with ReLU) followed by a logistic regression into 2 classes (same as reference or not).
- Bounding box regression is part of the classification branch, but uses a different output layer. This output layer produces fine adjustments (deltas) of the bounding box coordinates (instead of class probabilities).
- Non-maximum suppression (NMS; threshold 0.7) is applied to the predicted bounding boxes.

1.5. Segmentation head

- Inputs: the computed bounding boxes are cropped from P_i, reshaped to 14×14, and concatenated for $i = \{2, \ldots, 5\}$.
- Architecture: four 3×3 conv layers (with ReLU and BN) followed by a transposed conv layer with 2×2 kernels and stride of 2, and a final 1×1 conv layer outputting two feature maps consisting of logits for foreground/background at each spatial location.

[2]Using the notation from here: `http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006`

S_1		S_2		S_3		S_4	
1	Person	2	Bicycle	3	Car	4	Motorcycle
5	Airplane	6	Bus	7	Train	8	Truck
9	Boat	10	Traffic light	11	Fire Hydrant	12	Stop sign
13	Parking meter	14	Bench	15	Bird	16	Cat
17	Dog	18	Horse	19	Sheep	20	Cow
21	Elephant	22	Bear	23	Zebra	24	Giraffe
25	Backpack	26	Umbrella	27	Handbag	28	Tie
29	Suitcase	30	Frisbee	31	Skis	32	Snowboard
33	Sports ball	34	Kite	35	Baseball bat	36	Baseball glove
37	Skateboard	38	Surfboard	39	Tennis rocket	40	Bottle
41	Wine glass	42	Cup	43	Fork	44	Knife
45	Spoon	46	Bowl	47	Banana	48	Apple
49	Sandwich	50	Orange	51	Broccoli	52	Carrot
53	Hot dog	54	Pizza	55	Donut	56	Cake
57	Chair	58	Couch	59	Potted plant	60	Bed
61	Dining table	62	Toilet	63	TV	64	Laptop
65	Mouse	66	Remote	67	Keyboard	68	Cell phone
69	Microwave	70	Oven	71	Toaster	72	Sink
73	Refrigerator	74	Book	75	Clock	76	Vase
77	Scissors	78	Teddy bear	79	Hair drier	80	Toothbrush

Table 3. One-shot class splits (S_1 – S_4, Section 3) of MS-COCO.