Augmentation for small object detection

Mate Kisantal[1] `kisantal.mate@gmail.com`
Zbigniew Wojna[1,2] `zbigniewwojna@gmail.com`
Jakub Murawski[2,3] `kuba.murawski96@gmail.com`
Jacek Naruniec[3] `j.naruniec@ire.pw.edu.pl`
Kyunghyun Cho[4] `kyunghyun.cho@nyu.edu`

[1] Tensorflight, Inc.
[2] University College London
[3] Warsaw University of Technology
[4] New York University

Abstract. In the recent years, object detection has experienced impressive progress. Despite these improvements, there is still a significant gap in the performance between the detection of small and large objects. We analyze the current state-of-the-art model, Mask-RCNN, on a challenging dataset, MS COCO. We show that the overlap between small ground-truth objects and the predicted anchors is much lower than the expected IoU threshold. We conjecture this is due to two factors; (1) only a few images are containing small objects, and (2) small objects do not appear enough even within each image containing them. We thus propose to oversample those images with small objects and augment each of those images by copy-pasting small objects many times. It allows us to trade off the quality of the detector on large objects with that on small objects. We evaluate different pasting augmentation strategies, and ultimately, we achieve 9.7% relative improvement on the instance segmentation and 7.1% on the object detection of small objects, compared to the current state of the art method on MS COCO.

Keywords: object detection, small object detection, instance segmentation, small objects, data augmentation.

1   Introduction

Detecting objects in an image is one of the fundamental tasks of todays computer vision research, as it is often a starting point for many real world applications, including robotics and self-driving cars, satellite and aerial image analysis, and the localization of organs and masses in medical images. This important problem of object detection has recently experienced a lot of progress. The top-1 solution on MS COCO object detection competition,[5] has progressed from the average precision (AP) of 0.373 in 2015 [1] to 0.525 in 2017 (at IoU=.50:.05:.95 which is a primary challenge metric.) Similar progress can be observed in the instance segmentation problem in the context of MS COCO instance segmentation challenge.

---

[5] `http://cocodataset.org/\#detection-leaderboard`

		AP ▾	AP⁵⁰	AP⁷⁵	APˢ	APᴹ	APᴸ
⊕	Megvii (Face++)	0.526	0.730	0.585	0.343	0.556	0.660
⊕	UCenter	0.510	0.705	0.558	0.326	0.539	0.648
⊕	MSRA	0.507	0.717	0.566	0.343	0.529	0.627
⊕	FAIR Mask R-CNN	0.503	0.720	0.558	0.328	0.537	0.627
⊕	Trimps-Soushen+QINIU	0.482	0.681	0.534	0.310	0.512	0.610
⊕	bharat_umd	0.482	0.694	0.536	0.312	0.514	0.606
⊕	DANet	0.459	0.676	0.509	0.283	0.483	0.591
⊕	BUPT-Priv	0.435	0.659	0.475	0.251	0.477	0.566
⊕	DL61	0.424	0.633	0.471	0.246	0.458	0.551
⊕	DeNet	0.424	0.615	0.455	0.222	0.463	0.584

Fig. 1. In the top submissions for MS COCO instance segmentation challenge, the AP detection metric for small objects is 2-3 times lower than that for large objects. The leaderboard snapshot was taken in the summer of 2018.

	Min rectangle area	Max rectangle area
Small object	$0{\times}0$	$32{\times}32$
Medium object	$32{\times}32$	$96{\times}96$
Large object	$96{\times}96$	$\infty \times \infty$

Table 1. The definitions of the small, medium and large objects in MS COCO.

Despite these improvements, existing solutions often underperform with small objects, where small objects are defined as in Table 1 in the case of MS COCO. It is evident from the significant gap in the performance between the detection of small and large objects. See for instance Figure 1, which lists the top ranking submissions for the MS COCO instance segmentation challenge. A similar issue is observed in the instance segmentation task as well. For instance, see the sample predictions from the current state-of-the-art model, Mask-RCNN, in Figure 2, where the model has missed most of the small objects.

Small object detection is crucial in many downstream tasks. Detecting small or distant objects in the high-resolution scene photographs from the car is necessary to deploy self-driving cars safely. Many objects, such as traffic signs [2, 3] or pedestrians [4], are often barely visible on the high-resolution images. In medical imaging, early detection of masses and tumors is crucial for making an accurate, early diagnosis, when such elements can easily be only a few pixels in size [5, 6]. Automatic industrial inspection can also benefit from small object detection by the localization of small defects that can be visible on the material surfaces [7, 8]. Another application is
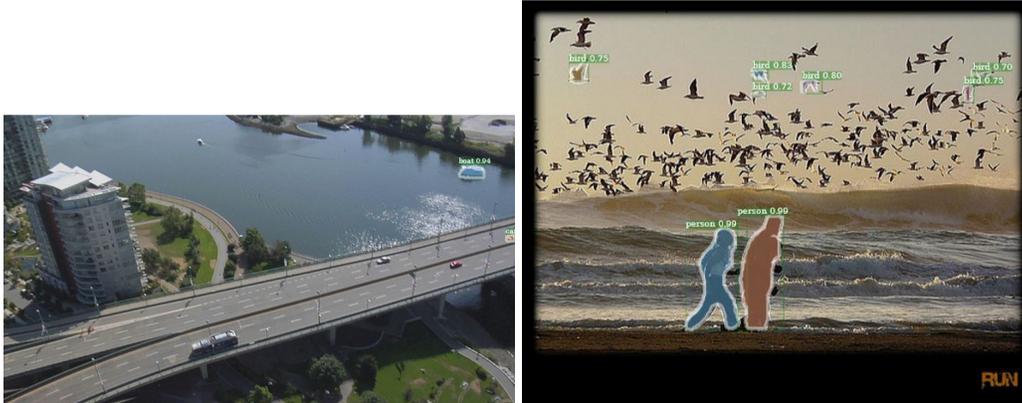
Fig. 2. Sample predictions from the Mask-RCNN, where many small objects were missed by the system, despite their clear visibility within a reasonable context. For instance, only six birds were detected out of hundreds of them.

satellite image analysis, where objects, such as cars, ships, and houses, must be effectively annotated [9, 10]. With an average of 0.5-5m per pixel resolution, these objects are again just a few pixels in size. In other words, small object detection and segmentation requires more attention, as more complex systems are being deployed in the real world. We, therefore, propose a new method to improve small object detection.

We focus on the state-of-the-art object detector, Mask R-CNN [11], on a challenging dataset, MS COCO. We note two properties of this dataset regarding small objects. First, we observe that there are relatively fewer images that contain small objects in the dataset, which potentially biases any detection model to focus more on medium and large objects. Second, the area covered by small objects is much smaller, implying the lack of diversity in the locations of small objects. We conjecture this makes it difficult for the object detection model to generalize to small objects in the test time when they appear in less explored portions of an image.

We tackle the first issue by oversampling those images containing small objects. The second issue is addressed by copy-pasting small objects multiple times in each image containing small objects. When pasting each object, we ensure that pasted objects do not overlap with any existing object. This increases the diversity in the locations of small objects while ensuring that those objects appear in the correct context, as shown in Fig. 3. The increase in the number of small objects in each image further addresses the issue of a small number of positively matched anchors, which we quantitatively analyze in Section 3. Overall, we achieve 9.7% relative improvement for the instance segmentation and 7.1% for object detection for small objects, compared to the current state-of-the-art method, Mask R-CNN, on MS COCO.

Fig. 3. Examples of artificial augmentation by copy pasting the small objects. As we can observe in these examples, pasting at the same image gives a high chance of the right surrounding context of the small object.

2   Related Work

Object Detection Faster region-based convolutional neural network (Faster R-CNN) [1], Region-based fully convolutional network (R-FCN) [12] and Single Shot Detector (SSD) [13] are three major approaches to object detection and they differ by whether and where the region proposal is attached [14]. Faster R-CNN and its variants are designed to help with a variety of object scales, as differential cropping merges all proposals into a single resolution. This, however, happens inside a deep convolutional network, and the resulting cropped boxes may not align perfectly with objects, which may hurt its performance in practice. SSD was recently extended into Deconvolutional Single Shot Detector (DSSD) [15], which upsamples the low-resolution features of SSD by the transposed convolutions in the decoder part [16], to increase the internal spatial resolution. Similarly, Feature Pyramid Network (FPN) [17] extends the Faster R-CNN with decoder type sub-network.

Instance Segmentation Instance segmentation goes beyond object detection and requires predicting the exact mask of each object. Multi-Task Network Cascades (MNC) [18] build a cascade of prediction and mask refinement. Fully convolutional instance-aware semantic segmentation (FCIS) [19] computes a position-sensitive score map shared by every region of interest. [20], which is also a fully convolutional approach, learns pixel embedding. Mask R-CNN [11] extends the FPN model with a branch for predicting masks and introduces new differential cropping for both object detection and instance segmentation.

Small objects Detecting small objects may be addressed by increasing the input image resolution [21, 13] or by fusing high-resolution features with high-dimensional features from the low-resolution image [22–25]. This approach of using the higher
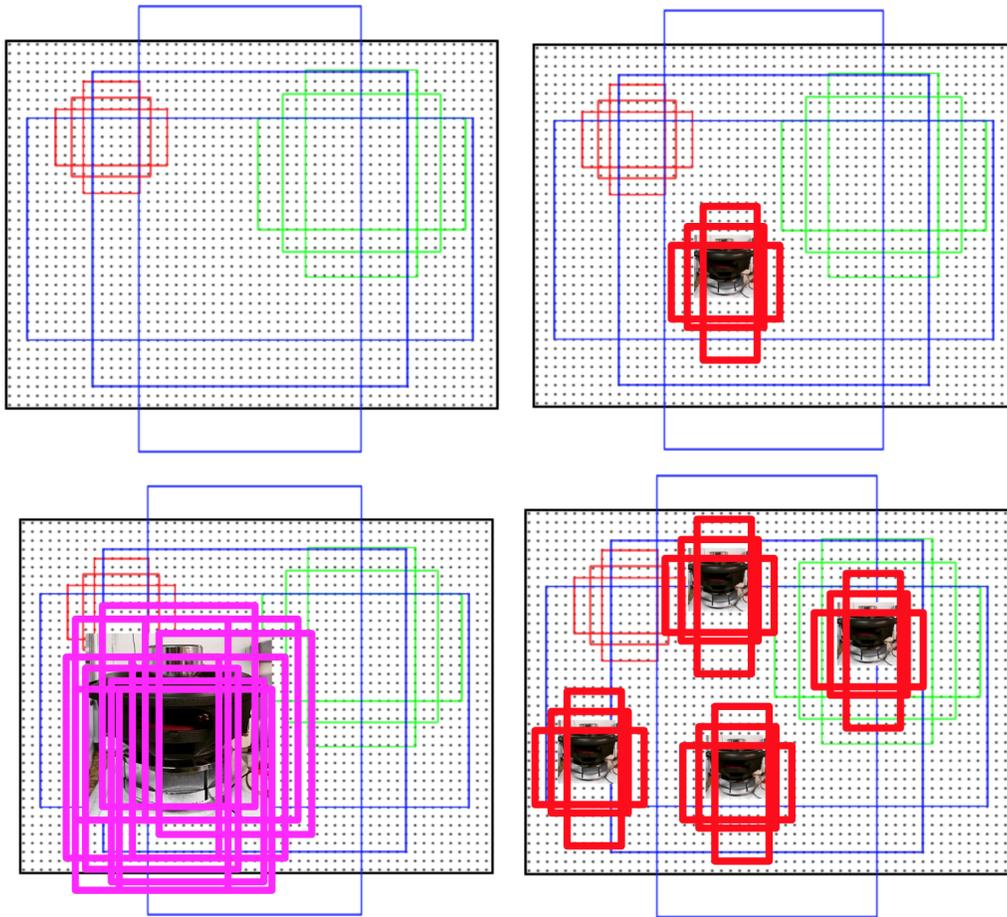
Fig. 4. Schematic illustration of anchors of different scales matching the ground truth objects. Small objects have much less anchors matched. To overcome the problem, we propose to artificially augment the images by copy-pasting small objects, so that there is more anchors positively matched with small objects during training.

resolution, however, increases computational overhead and does not address the imbalance between small and large objects. [26] instead uses a Generative Adversarial Network (GAN) to build features in a convolutional network that are indistinguishable between small and large objects in the context of a traffic sign and pedestrian detection. [27] uses different anchor scales based on different resolution layers in a region proposal network. [28] shifts image features by the correct fraction of the anchor size to cover gaps between them. [29–32] add the context when cropping a small object proposal.

3 Identifying issues with detecting small objects

In this section, we first overview the MS COCO dataset and the object detection model used in our experiments. We then discuss the issues of the MS COCO dataset and the anchor matching process used in training, that contributes to the difficulty of small object detection.

3.1 MS COCO

We experiment with the MS COCO Detection dataset [33]. The MS COCO 2017 Detection dataset contains 118,287 images for training, 5,000 images for validation, and 40,670 test images. 860,001 and 36,781 objects from 80 categories are annotated with ground-truth bounding boxes and instance masks.

In the MS COCO detection challenge, the primary evaluation metric is the average precision (AP). In general, AP is defined as the average of ratios of true positives to all positives for all recall values. Because an object needs to be both located and correctly classified, correct classification is only counted as a true positive detection if the predicted mask or bounding box has an intersection-over-union (IoU) higher than 0.5. The AP scores are averaged across the 80 categories and ten IoU thresholds, evenly distributed between 0.5 and 0.95. The metrics also include AP measured across different object scales. In this work, our primary interest is the AP on small objects.

3.2 Mask R-CNN

For our experiments, we use the Mask R-CNN implementation from [34] with a ResNet-50 backbone and adapt the linear scaling rule proposed in [35] for setting learning hyperparameters. We use a shorter training schedule than the baselines in [34]. We train our models for 36k iterations distributed over four GPUs, using a base learning rate of 0.01. For optimization, we use stochastic gradient descent with the momentum set to 0.9 and weight decay, with the coefficient set to 0.0001. The learning rate is scaled down with a factor of 0.1 twice during training, after 24k and 32k iterations. All the other parameters are kept as in the baseline Mask R-CNN+FPN+ResNet-50 configuration from [34].

The region proposal stage of the network is particularly important in our investigation. We are using a feature pyramid network (FPN) for generating object proposals [17]. It predicts object proposals relative to fifteen anchor boxes from five scales $(32^2, 64^2, 128^2, 256^2, 512^2)$ and three aspect ratios $(1, 0.5, 2)$. An anchor receives a positive label if it has an IoU higher than 0.7 against any ground-truth box, or if it has the highest IoU against a ground-truth bounding box.

	Object Count	Images	Total Object Area	Matched Anchors	Average matching anchors	Average max IoU
small	41.43%	51.82%	1.23%	29.96%	1.00	0.29
medium	34.32%	70.07%	10.18%	25.54%	1.03	0.57
large	24.24%	82.28%	88.59%	44.49%	2.54	0.66

Table 2. The MS COCO dataset objects statistics with respect to matched anchors in Mask-RCNN based on RPN.

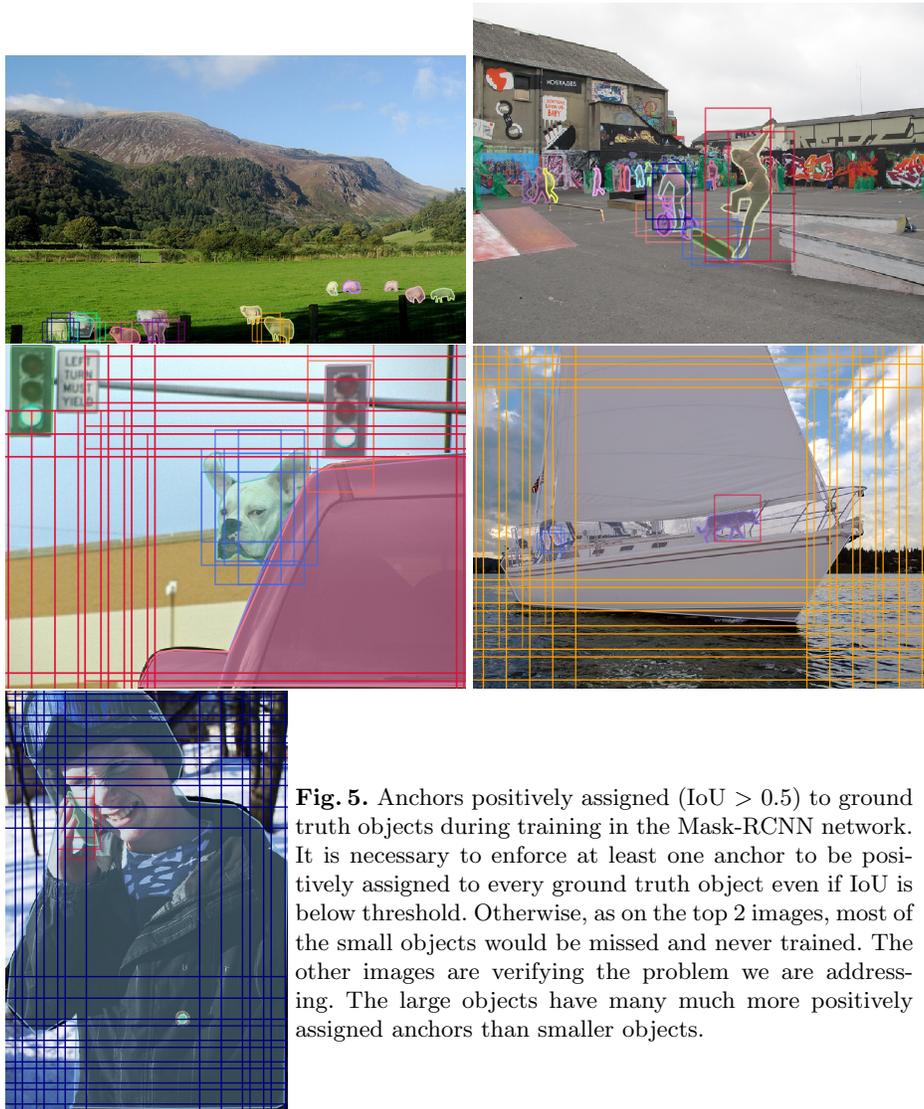**Fig. 5.** Anchors positively assigned (IoU > 0.5) to ground truth objects during training in the Mask-RCNN network. It is necessary to enforce at least one anchor to be positively assigned to every ground truth object even if IoU is below threshold. Otherwise, as on the top 2 images, most of the small objects would be missed and never trained. The other images are verifying the problem we are addressing. The large objects have many much more positively assigned anchors than smaller objects.

3.3 Small object detection by Mask R-CNN on MS COCO

In MS COCO, 41.43% of all the objects appearing in the training set are small, while only 34.4% and 24.2% are medium and large objects, respectively. On the other hand, only about half of the training images contain any small objects, while 70.07% and 82.28% of training images contain medium and large objects, respectively. see *Object Count* and *Images* in the Table 2. This confirms the first issue of small object detection. The problem is related to the diversity of the objects and their contexts. While for bigger objects, we have more images, meaning more diversity, smaller objects seem to be grouped in the same images and scenes.

The second issue is immediately apparent by considering the *Total Object Area* for each size category. A mere 1.23% of the annotated pixels belong to small objects. Medium-sized objects take up already more than eight times more area, 10.18% of the total annotated pixels, while the majority of pixels, 82.28% are labeled as parts of the large objects. Any detector trained on this dataset does not see enough cases of small objects, both across images and across pixels.

As described earlier in this section, each predicted anchor from the region proposal network receives a positive label if it has the highest IoU with a ground-truth bounding box or if it has an IoU higher than 0.7 for any ground truth box. This procedure highly favors large objects, as a large object spanning multiple sliding-window locations often has a high IoU with many anchor boxes, while a small object may only be matched with a single anchor box with a low IoU. As listed in Table 2, only 29.96% of positively matched anchors are paired with small objects, while 44.49% of positively matched anchors with large objects. From the other perspective, it implies that there are 2.54 matched anchors per large object, while only one matched anchor per small object. Furthermore, as the *Average Max IoU* metric reveals, even the best matching anchor box of a small object has a low IoU value typically. The average max IoU for small objects is only 0.29, while medium and large objects have their best matching anchors at around two times higher IoU, 0.57 and 0.66, respectively. We illustrate this phenomenon in fig. 5 by visualizing a few examples. These observations suggest that small objects contribute much less to computing the region proposal loss, which biases the entire network to favor large and medium objects.

4 Oversampling and Augmentation

We are improving the performance of object detectors on small objects by explicitly addressing the small object related issues of the MS COCO dataset that we outlined in the previous section. In particular, we over-sample images containing small objects and perform small object augmentation to encourage a model to focus more on small objects. Although we evaluate the proposed approach using Mask

R-CNN, it can be used with any other object detection network or framework, as both oversampling and augmentation are done as data preprocessing.

Oversampling We address the issue of relatively fewer images containing small objects by oversampling those images during training [36]. It is an effortless and straight-forward way to alleviate this problem of the MS COCO dataset and improve performance on small object detection. In the experiments, we vary the oversampling rate. We investigate the effect of oversampling not only on small object detection but also on detecting medium and large objects.

Augmentation On top of oversampling, we also introduce dataset augmentation focused on small objects. Instance segmentation masks provided in the MS COCO dataset allow us to make a copy of any object from its original location. The copy is then pasted to different positions. By increasing the number of small objects in each image, the number of matched anchors increases. This, in turn, improves the contribution of small objects to computing the loss function of the RPN during training.

Before pasting the object to a new location, we apply random transformations on it. We scale the objects by changing the object size $\pm 20\%$ and rotate it $\pm 15°$. We only consider non-occluded objects, as pasting disjoint segmentation masks with unseen parts in-between often results in less realistic images. We ensure that the newly pasted object does not overlap with any existing object and is at least five pixels away from the image boundaries.

In Fig. 4, we graphically illustrate the proposed augmentation strategy and how it increases the number of matched anchors during training, leading to a better detector of small objects.

5 Experimental Setup

5.1 Oversampling

In the first set of experiments, we investigate the effect of oversampling images containing small objects. We vary the oversampling ratio between two, three, and four. Instead of actual stochastic oversampling, we create multiple copies of images with small objects offline for efficiency.

5.2 Augmentation

In the second set of experiments, we investigate the effects of using augmentation on small object detection and segmentation. We copy and paste all small objects in each image once. We also oversample images with small objects to study the interaction between the oversampling and augmentation strategies.

	Segmentation AP				Detection AP			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.28	0.167	0.329	0.393	0.303
oversampling 2×	0.120	0.299	0.409	0.279	0.173	0.328	0.387	0.304
oversampling 3×	**0.123**	0.300	0.404	0.279	**0.177**	0.329	0.382	**0.305**
oversampling 4×	0.120	0.299	0.398	0.276	0.174	0.329	0.374	0.302

Table 3. Experiments with different oversampling ratios. We observe that oversampling helps regardless of the ratio for detecting small objects. The ratio allows us to make a trade-off between small and large objects.

We test three settings. In the first setting, we replace each image with small objects by the one with copy-pasted small objects. In the second setting, we duplicate these augmented images to mimic oversampling. In the final setup, we keep both the original and augmented images, which is equivalent to oversampling the images with small objects by the factor of two, while augmenting the duplicated copies with more small objects.

5.3 Copy-Pasting Strategies

There are different ways to copy-pasting small objects. We consider three separate strategies. First, we pick one small object in an image and copy-paste it multiple times in random locations. Second, we choose numerous small objects and copy-paste each of these exactly once in an arbitrary position. Lastly, we copy-paste all small objects in each image multiple times in random places. In all the cases, we use the third setting of augmentation above; that is, we keep both the original image and its augmented copy.

5.4 Pasting Algorithms

When pasting a copy of a small object, there are two things to consider. First, we must decide whether a pasted object would overlap with any other object. Although we choose not to introduce any overlap, we experimentally verify whether it is a good strategy. Second, it is a design choice whether to perform an additional procedure to smooth the edge of a pasted object. We experiment whether Gaussian blurring of the boundary with varying filter sizes could help compared to no further processing.

6 Result and Analysis

6.1 Oversampling

By sampling the small object images more often during training (see Table 3), AP on both small object segmentation and detection can be improved. The most

	Segmentation AP				Detection AP			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.28	0.167	0.329	0.393	0.303
aug	0.108	0.299	0.422	0.278	0.161	0.328	0.4	0.302
aug+oversample 2x	0.117	0.300	0.406	0.277	0.168	0.333	0.387	0.302
original+aug	**0.124**	0.301	0.41	**0.28**	**0.179**	0.329	0.386	**0.304**

Table 4. Augmentation experiments. The best performance, in terms of both small objects and overall, is achieved when the original images with small objects and their copy with copy-pasted small objects are used for training.

gain is observed with $3\times$ oversampling, which increases AP for small objects by 1% (corresponding to a relative improvement of 8.85%). While performance on the medium object scale is less affected, large object detection and segmentation performance consistently suffer from oversampling, implying that the ratio must be chosen based on the relative importance between small and large objects.

6.2 Augmentation

In Table 4, we present the results using different combinations of the proposed augmentation and oversampling strategy. When we replace each image with small objects by its copy that contains more small objects (the second row), the performance degraded notably. When we oversampled these augmented images by the factor of two, the segmentation and detection performance on the small objects regained its loss, although the overall performance was still worse than the baseline. When we evaluated this model on an augmented validation set, instead of the original one, we, however, saw a 38% increase in the small object augmentation performance (0.161), suggesting that the trained model effectively overfit to "pasted" small objects but not necessarily to the original small objects. We believe this is due to the artifacts from pasting, such as imperfect object masks and brightness differences from the background, that are relatively easy for a neural network to spot. The best results were achieved by combining oversampling and doing augmentation with a probability of $p = 0.5$ (*original+aug*) with the ratio of original to augmented small objects is 2:1. This setting yielded better results than oversampling alone, confirming the effectiveness of the proposed strategy of pasting small objects.

6.3 Copy-Pasting strategies

Copy-pasting of a single object In Table 5, we see that copy-pasting a single object results in a better model on small objects; however, at the cost of a slight performance drop on large images. These results are also better than two times over-

	Segmentation AP				Detection			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.280	0.167	0.329	0.393	0.303
1× pasted	**0.122**	0.303	0.405	**0.281**	**0.174**	0.333	0.384	**0.307**
2× pasted	0.122	0.300	0.408	0.279	0.175	0.330	0.382	0.304
3× pasted	0.121	0.301	0.403	0.280	0.175	0.330	0.386	0.305
4× pasted	0.120	0.298	0.403	0.277	0.174	0.328	0.383	0.304
5× pasted	0.116	0.298	0.405	0.277	0.171	0.328	0.386	0.304

Table 5. Copy-pasting of a single object. We observe that it is often best to copy-paste a single object only a few times (1× or 2×,) especially to achieve the high overall performance.

	Segmentation AP				Detection			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.280	0.167	0.329	0.393	0.303
1× pasted	0.122	0.303	0.405	0.281	0.174	0.333	0.384	0.307
2× pasted	0.121	0.300	0.406	0.279	0.175	0.329	0.385	0.305
3× pasted	**0.124**	0.303	0.411	**0.281**	**0.178**	0.331	0.387	**0.306**
4× pasted	0.120	0.299	0.403	0.279	0.179	0.329	0.382	0.305
5× pasted	0.119	0.301	0.412	0.279	0.176	0.330	0.389	0.305

Table 6. Copy-pasting of multiple objects. Compared to copy-pasting of a single object, it is better to make more copies (3×).

sampling in itself. The performance, however, peaks already at one or two pastes. Adding the same object more times does not yield any performance improvement.

Copy-pasting of multiple objects As it can be seen in Table 6, it is better to copy-paste multiple small objects per image than to copy-paste only a single object. In this case, we see the benefits of pasting up to three times per object.

Copy-pasting of all small objects Finally, Table 7 lists the results where all the small objects in each image are copy-pasted. We found the best results concerning both the segmentation and detection at augmenting with all the objects once. We suspect two possible causes behind this. First, By having multiple copies of all small objects the ratio of original to pasted small objects rapidly decreases. Second, the number of objects in each image multiplies, and this causes a more considerable mismatch between training and test images.

	Segmentation AP				Detection			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.280	0.167	0.329	0.393	0.303
1× pasted	**0.124**	0.301	0.410	**0.280**	**0.179**	0.329	0.386	**0.304**
2× pasted	0.119	0.299	0.410	0.279	0.179	0.328	0.388	0.304
3× pasted	0.113	0.299	0.405	0.276	0.167	0.330	0.383	0.302

Table 7. Copy-pasting of all small objects. It is best to make only one copy of every small object, and this strategy does not outperform the strategy of copy-pasting multiple (but not all) objects many times.

	Segmentation AP				Detection			
	small	medium	large	all	small	medium	large	all
original+aug	**0.124**	0.301	0.41	**0.28**	**0.179**	0.329	0.386	**0.304**
overlapping	0.118	0.301	0.411	0.279	0.173	0.331	0.389	0.305
blur 3	0.117	0.299	0.408	0.278	0.17	0.329	0.389	0.304
blur 5	0.119	0.3	0.408	0.279	0.173	0.328	0.383	0.303

Table 8. Results with different pasting algorithms. We observe that it is crucial not to introduce any overlap when copy-pasting a small object, and that it is not advisable to Gaussian-blur the edge of a pasted object.

6.4   Pasting Algorithms

As shown in the Table 8, pasting randomly into images without considering what other objects already occupy areas leads to inferior performance on small images. It justifies our design choice to avoid any overlap between a pasted object and existing objects. Further, Gaussian blurring of the edge of a pasted object did not show any improvement, suggesting that it is better to paste an object as it is, unless with a more sophisticated strategy of fusing in the object.

7   Conclusion

We investigated the problem of small object detection. We showed that one of the factors behind the reduced average precision for small objects is the lack of representation of small objects in training data. This is especially true with the existing state-of-the-art object detector, which requires the presence of enough objects for predicted anchors to match during training. We proposed two strategies for augmenting the original MS COCO database to overcome the issue. First, we show the performance on small objects can easily improve by oversampling images containing small objects during training. Second, we propose an augmentation algorithm based on copy-pasting small objects. Our experiments proved a 9.7% relative improvement for the instance segmentation and 7.1% for object detection for small

objects compared to the current state of the art, obtained by Mask R-CNN, on MS COCO. The proposed set of augmentation methods offers the trade-off between the quality of predictions for small and large objects, as verified by the experiments.

References

1. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
2. Deshmukh, V.R., Patnaik, G., Patil, M.: Real-time traffic sign recognition system based on colour image segmentation. International Journal of Computer Applications **83** (2013)
3. Sermanet, P., LeCun, Y.: Traffic sign recognition with multi-scale convolutional networks. In: International Joint Conference on Neural Networks, IEEE (2011) 2809–2813
4. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2056–2063
5. Bottema, M.J., Slavotinek, J.P.: Detection and classification of lobular and dcis (small cell) microcalcifications in digital mammograms. Pattern Recognition Letters **21** (2000) 1209–1214
6. Nagarajan, M.B., Huber, M.B., Schlossbauer, T., Leinsinger, G., Krol, A., Wismüller, A.: Classification of small lesions in dynamic breast mri: eliminating the need for precise lesion segmentation through spatio-temporal analysis of contrast enhancement. Machine vision and applications **24** (2013) 1371–1381
7. Abouelela, A., Abbas, H.M., Eldeeb, H., Wahdan, A.A., Nassar, S.M.: Automated vision system for localizing structural defects in textile fabrics. Pattern recognition letters **26** (2005) 1435–1443
8. Ng, H.F.: Automatic thresholding for defect detection. Pattern recognition letters **27** (2006) 1644–1649
9. Modegi, T.: Small object recognition techniques based on structured template matching for high-resolution satellite images. In: SICE Annual Conference, 2008, IEEE (2008) 2168–2173
10. Kampffmeyer, M., Salberg, A.B., Jenssen, R.: Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2016) 1–9
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2017) 2961–2969
12. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. (2016) 379–387
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Proceedings of the European Conference on Computer Vision, Springer (2016) 21–37
14. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 7310–7311
15. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
16. Wojna, Z., Ferrari, V., Guadarrama, S., Silberman, N., Chen, L.C., Fathi, A., Uijlings, J.: The devil is in the decoder: Classification, regression and gans. International Journal of Computer Vision (2019) 1–13
17. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2117–2125

18. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3150–3158
19. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2359–2367
20. Fathi, A., Wojna, Z., Rathod, V., Wang, P., Song, H.O., Guadarrama, S., Murphy, K.P.: Semantic instance segmentation via deep metric learning. arXiv preprint arXiv:1703.10277 (2017)
21. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. (2015) 424–432
22. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2129–2137
23. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2874–2883
24. Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., Wu, J.: Feature-fused ssd: fast detection for small objects. In: Ninth International Conference on Graphic and Image Processing (ICGIP 2017). Volume 10615., International Society for Optics and Photonics (2018) 106151E
25. Menikdiwela, M., Nguyen, C., Li, H., Shaw, M.: Cnn-based small object detection and visualization with feature activation mapping. In: 2017 International Conference on Image and Vision Computing New Zealand, IVCNZ 2017, Christchurch, New Zealand, December 4-6, 2017. (2017) 1–5
26. Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017)
27. Eggert, C., Zecha, D., Brehm, S., Lienhart, R.: Improving small object proposals for company logo detection. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ACM (2017) 167–174
28. Fang, L., Zhao, X., Zhang, S.: Small-objectness sensitive detection based on shifted single shot detector. Multimedia Tools and Applications (2018) 1–19
29. Chen, C., Liu, M.Y., Tuzel, O., Xiao, J.: R-cnn for small object detection. In: Asian conference on computer vision, Springer (2016) 214–230
30. Ren, Y., Zhu, C., Xiao, S.: Small object detection in optical remote sensing images via modified faster r-cnn. Applied Sciences **8** (2018) 813
31. Cheng, P., Liu, W., Zhang, Y., Ma, H.: Loco: Local context based faster r-cnn for small traffic sign detection. In: International Conference on Multimedia Modeling, Springer (2018) 329–341
32. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE (2017) 1522–1530
33. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European Conference on Computer Vision, Springer (2014) 740–755
34. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. `https://github.com/facebookresearch/detectron` (2018)
35. Goyal, P., Dollár, P., Girshick, R.B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. CoRR **abs/1706.02677** (2017)
36. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106** (2018) 249–259