International Journal of Computational Intelligence Systems
Vol. 14(1), 2021, pp. 1871–1879
DOI: https://doi.org/10.2991/ijcis.d.210622.003; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Detecting Objects from No-Object Regions: A Context-Based Data Augmentation for Object Detection

Jun Zhang[1], Feiteng Han[1,*], Yutong Chun[1,*], Kangwei Liu[2], Wang Chen[3]

[1] *School of Management and Engineering, Capital University of Economics and Business, No. 121, Zhangjia Road, Huaxiang, Fengtai District, Beijing, 100070, China*
[2] *Alibaba DAMO Academy, District 4, Wangjing East Park, Chaoyang District, Beijing, 100102, China*
[3] *School of Information, Renmin University of China, No.59 Zhongguancun Road, Haidian District, Beijing, 100872, China*

ARTICLE INFO

Article History

Received 08 Oct 2020
Accepted 31 May 2021

Keywords

Data augmentation
Object detection
Convolutional neural network
Deep learning

ABSTRACT

Data augmentation is an important technique to improve the performance of deep learning models in many vision tasks such as object detection. Recently, some works proposed the copy-paste method, which augments training dataset by copying foreground objects and pasting them on background images. By designing a learning-based context model to predict realistic placement regions, these approaches have been proved to be more effective than traditional data augmentation methods. However, the performance of the existing context model was limited by three problems: (1) The definitions of positive and negative samples generate too much label noise. (2) The examples with masked regions lose a lot of context information. (3) The sizes (i.e., scale and aspect ratios) of predicted regions are sampled from a prior shape distribution, which leads to a coarse estimation. In this work, we first explore the placement rules that generate realism and effective training examples for detectors. And then, we propose a trainable context model in order to find proper placement regions by classifying and refining dense prior default boxes. We also design a corresponding reasonable generation for training examples by annotating ground truth on free space according to the placement rules. The experimental results on PASCAL VOC show that our approach outperforms the state-of-the-art-related work.

© 2021 *The Authors*. Published by Atlantis Press B.V.
This is an open access article distributed under the CC BY-NC 4.0 license (http://creativecommons.org/licenses/by-nc/4.0/).

1. INTRODUCTION

Object detection is an important task in computer vision, which is the basis of many applications, such as autonomous driving [1–4], face recognization [5–7], human detection [8]. In recent years, the performance of object detectors has been improved considerably due to the great success of deep learning algorithms. However, these deep learning models rely heavily on a large amount of well-annotated training data. While only a few labelled data is available, the performance of deep models will be greatly limited. Therefore, a common idea is to collect a large amount of raw data for manual annotation. However, this process is much more laborious and costly. More importantly, it is difficult to obtain a huge number of annotated training examples in many tasks.

To solve such a problem, data augmentation becomes one of the most effective techniques. Some augmentation methods such as horizontal flipping [9], random erasing [10] have shown an improvement potential in some detection tasks. However, these works only focused on changing the appearance and geometry of the original images, which significantly limits the performance of these traditional augmentation algorithms.

Consequently, copy-paste augmentation approaches [11–15] are presented in recent years, which augment the training data by pasting foreground objects on some background images with a variety of visual scene. In this way, the augmented training data can be obtained with a rich visual background. Among these works, [11,14] adopt a context-free way, which pastes foreground objects on backgrounds with random placement. However, these context-free approaches are only effective on object instance detection task, which leads to a substantial accuracy drops on generic object detection [15,16], because it ignores the relationship between foreground object and surrounding visual context, which is the key factor in object detection [17–19].

Therefore, it is a significant problem to find a proper region on background image for foreground object to be placed. The works [15,16] explore visual context role in object detection augmentation field systematically, and introduce a learning-based context model to find the proper placement regions. Specifically, the context model modeling surrounding visual context information for candidate boxes (i.e., candidate regions) of different shapes, it outputs the probability of each candidate box. Although these context-based copy-paste augmentation algorithms have been proved to outperform the traditional and context-free copy-paste augmentation algorithms, there are still some problems significantly limiting the performance of the context model: (1) The training examples

[*] *Corresponding authors. Email: h_feiteng@163.com; chunyutong@yeah.net*

contain a lot of noise data due to the unreasonable definition of negative samples. Specifically, the definition treats all no-object regions as negative samples. However, the target of the context model is to find the proper placement regions from no-object regions, while some no-object regions are indeed realistic placements regions. (2) The training examples were generated by masking the internal object ground truth, which makes the training examples lose a lot of context information. The loss of context information will make the context model tend to miss some proper regions and propose some useless regions (e.g., predicted regions have a high intersection with object ground truth of background image). (3) The size of the predicted region is coarse due to all sizes are sampled from a prior shape distribution. This makes the final synthetic image unrealistic though the location is proper. For the above problems, some questions should be answered: (i) What are placement rules to generate more realism and effective training examples for detectors? (ii) How to design a context model with effective training examples to learn these rules and automatically propose more effective regions with realistic class-location-size on the free space of background image?

In this paper, we first explore the reasonable placement rules to augment more effective training objects for detectors, which is the basis for designing context model with associated reasonable training examples. It is a challenge to formulate a series of rules where a region should be placed distinctly, because some locations are inherently ambiguous (e.g., the same place may be suitable for many different kinds of things). We simplify this complex problem into two rules which are proved to be simple yet effective by conducting related experiment.

Based on the rules, we propose a learning-based context model, which consists of two modules: Context Region Proposal Module (CRPM) and Class-Location-Size Aware Module (CLSAM). Firstly, CRPM takes a whole background image as input and output context region proposals with object placement proposals. Secondly, these context region proposals are fed into CLSAM which output the categories and refine offsets of the final object placement proposals.

In addition, we introduce a reasonable definition of negative and positive examples for generating effective training examples. Based on previous placement rules, we annotate placements ground truth on the free space of training images with a dense way. This will ensure that the positive examples sampled from free space with annotations while the negative examples sampled from the rest unreasonable free space.

The contributions of this paper are summarized as follows:

1. We discover the placement rules for copy-paste detection augmentation algorithm, which are simple and effective.
2. We introduce a context model to learn the rules and automatically predict more placement regions with realistic class-location-size on the free space of training image.
3. We compare our method with state-of-the-art related work on Faster R-CNN [9]. The results show that our approach has a better performance. We further conduct a series of experiments based on a variety of classic detectors, including one stage and two stage, to evaluate the generalization of our method. The experimental results indicate that our method achieves a persistent improvement on the different detectors.

2. RELATED WORK

In this section, we will review related works for data augmentation for object detection. All of these works are classified into two categories: non-copy-paste data augmentation and copy-paste data augmentation. The copy-paste data augmentations consists of context-free and context-based.

2.1. Non-Copy-Paste Detection Augmentation

Traditional detection augmentations try to improve performance by diverting the appearance and geometry structure of foreground object. Horizontal flipping [9,20] may be one of the most popular techniques to augment training data. In addition, the work of [20] adopts a lot of strategies such as random-brightness, random-lighting-noise, random-crop, random-expand to change the color, brightness, and scale of training data. For the same purpose, the authors of [10] propose a random erasing method, which randomly choose a rectangle region in the image and replace its pixels with random values or the ImageNet mean pixel value. Instead only cropping, the paper [21] randomly crops four images and patches them to create a new training image. This approach is suitable for not only detection but also classification task. However, in some specific task, such as logo detection(e.g., *vivo*, *adidas*, *oppo*), the change of spatial structure of objects will lead to ambiguity.

The authors of [22] try to employ the domain randomization strategy to synthesize images, and then handle the variability in real world data. With additional fine-tuning on real data, the network yields better performance than using real data alone. The work of [23] introduces an instance-switching (IS) strategy, which synthesize new training examples by switching instances of same class from other images. Besides, a novel class-balance method is proposed to balance the object instance. In the work of [24], a learning-based method is proposed to choose the best combination of data augmentation policies, which target at improving generalization performance for detectors.

2.2. Copy-Paste Detection Augmentation

In order to relieve the limitation of traditional augmentation, the emerging copy-paste methods were studied in recent years. The copy-paste methods copy the foreground object extracted from raw training data ground truth, and paste it on the background images.

Context-free copy-paste. Where do we place the foreground object on background image? The works [11,14] adopt a context-free strategy, which place the foreground object in a random way and ignore the surroundings visual context around foreground object. As for generic object instance detection, the authors of [11] utilize a segmentation model based on convolutional neural network (CNN) to obtain the foreground object instances and then paste it on background images randomly as training images. Besides, some blending strategy were utilized for addressing pixel artifacts problem.

However, this approach has no effect in generic object detection. In logo detection task, the work of [14] augments the training data with colorful foreground by doing logo foreground transformation (such as scaling, shearing, rotation, and coloring) before blending. However, the synthetic logo images are always used to pre-train deep models, and then fine-tune it with real-labeled training data.

Context-based copy-paste. In contrast, context-based algorithms [12,15] try to find a context-related region which make the synthetic image more photorealism after blend the foreground object on it. In Georgakis *et al.* [12], the authors propose a context-aware method which consider geometry and spatial structure of background subregions before place text foreground. Through this way, a substantial improvement of text localization have been gained. [15] may be the first work which discuss the importance of visual context on augmenting object detection training examples in a comprehensive way. This work focuses on the generic object detection task, and introduces a context model based on CNN to predict scores for candidate boxes from background image. Finally, a series of experiments conducted on public dataset show that it can obtain an improvement when few labeled data can be available. The work of [25] develops a task-aware method to augment training data. The method introduces a learning-based synthesizer network that is optimized to generate effective training instances by evaluating a target network. These two networks are trained in an adversarial way. In addition, the synthesizer is forced to synthesize photorealism images by a discriminator trained on real images.

3. THE PROPOSED METHOD

In this section, we will describe the entire pipeline of our method in details. The augmentation algorithm we proposed can be seen as an extension of the copy-paste augmentation algorithms. Instead of randomly placing foreground objects on background images, we first study the placement rules that generate photorealism and effective synthetic images. After that, we design a context model based on CNN to learn these placement rules, and automatically predict the effective placement regions. In addition, we introduce a reasonable definition of negative and positive samples to generate effective training examples for the context model. We further describe the objective function and optimization parameters of the context model. Therefore, we first introduce the exploration process of placement rules in Section 3.1. And then, the details of context model will be explained in Section 3.2.

3.1. The Exploration of Placement Rules

The context-free copy-paste augmentation algorithms [11,14] have shown an improvement on object instance detection task. However, this random-placement strategy was proved to be ineffective on generic object detection task by [15,16]. Therefore, the authors proposed a learning-based context model based on CNN, which modeled visual context information for candidate boxes and predicted the likelihood of a particular category. However, their definition about training examples treated all no-object regions as negative which ignored a fact that the target of copy-paste approach was to paste foreground objects on the free space of backgrounds. In contrast, we first study placement rules on free space for the copy-paste algorithm, which aim at generating photorealism and effective training examples for detectors. It is the basis of designing context model and automatically predicting more effective placement regions for foreground objects. Formulating the objective rules is a big challenge. To some extent, this is a problem of ambiguity. For instance, the same region may be suitable for many different kinds of objects, which is also contextual reasonable.

Instead of formulating this complex problem, we simplify it into two simple yet effective placement rules: (1) shifting. (i.e., shifting the ground truth to the surround free space with a realistic size); (2) relationship correlation. (i.e., making a strong correlation with surrounding different objects, such as the bottle should be on the table, people set on the chair, etc.). For the second rule, if a region is suitable for placing many different kinds of objects, we will randomly select one of them. Our rules are simple yet effective which have been proven based on experiments conducted on PASCAL VOC2012 dataset [26]. Specifically, we select a subset of VOC12(1449 images) as the basic data, and annotate some bounding boxes on free space with a dense way according to the above rules (some example pictures can be seen in Figure 1, and the statistical details of the annotated data are shown in Table 1). And then, we copy the foreground objects from segmentation ground truth and paste them on these bounding boxes with a blending strategy. Finally, the synthetic images with augmented foreground objects are used to train the detectors. From another perspective, although it need the handcraft annotations, it can be seen as one of laborious but effective augmentation algorithms when the raw images is difficult to obtain . More details about the experiments will be shown in Section 4.3.

3.2. The Context Model

The context model aims at predicting contextual reasonable placements on free space for foreground objects. From another perspective, it can be thought of as a "detection" framework that aims at detecting invisible objects on no-objects regions. Inspired by Faster R-CNN [9], we develop an end-to-end context model, which can be seen as a two-stage "invisible-object" detection framework consists of two modules: CRPM and CLSAM.

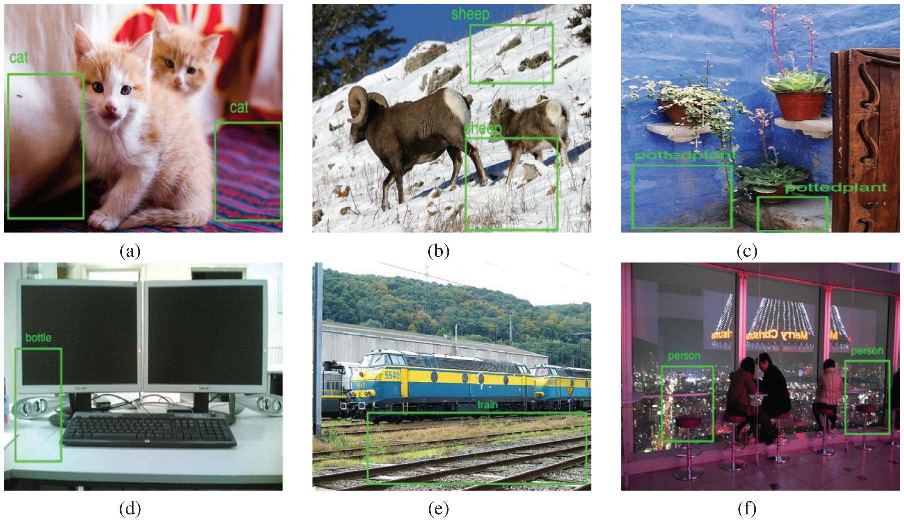
The details of our context model is shown in Figure 2. The CRPM predicts the context region proposals. The top left image in Figure 2 shows the generation of context region ground truth. The context region losses are related to context region ground truth instead of placement (object) ground truth. Based on the context region proposals, we can obtain the object default boxes (i.e., candidate placement regions). Then, the CLSAM output the classification scores and offsets. These offsets are about candidate placement regions and the final placement regions. Therefore, the regression loss is generated by candidate placement regions and placement (object) ground truth.

3.2.1. The context region proposal module

The CRPM has a same architecture with region proposal network (RPN) proposed in Faster R-CNN [9], which is composed of two

Table 1 Statistics of our annotated dataset.

Aero	Bike	Bird	Boat	Bott.	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Pers.	Plant	Sheep	Sofa	Train	TV	Total
174	42	230	115	229	23	134	194	66	68	26	171	48	34	445	179	101	19	51	40	**2389**



(a) (b) (c) (d) (e) (f)

Figure 1 Example of our placement dataset.

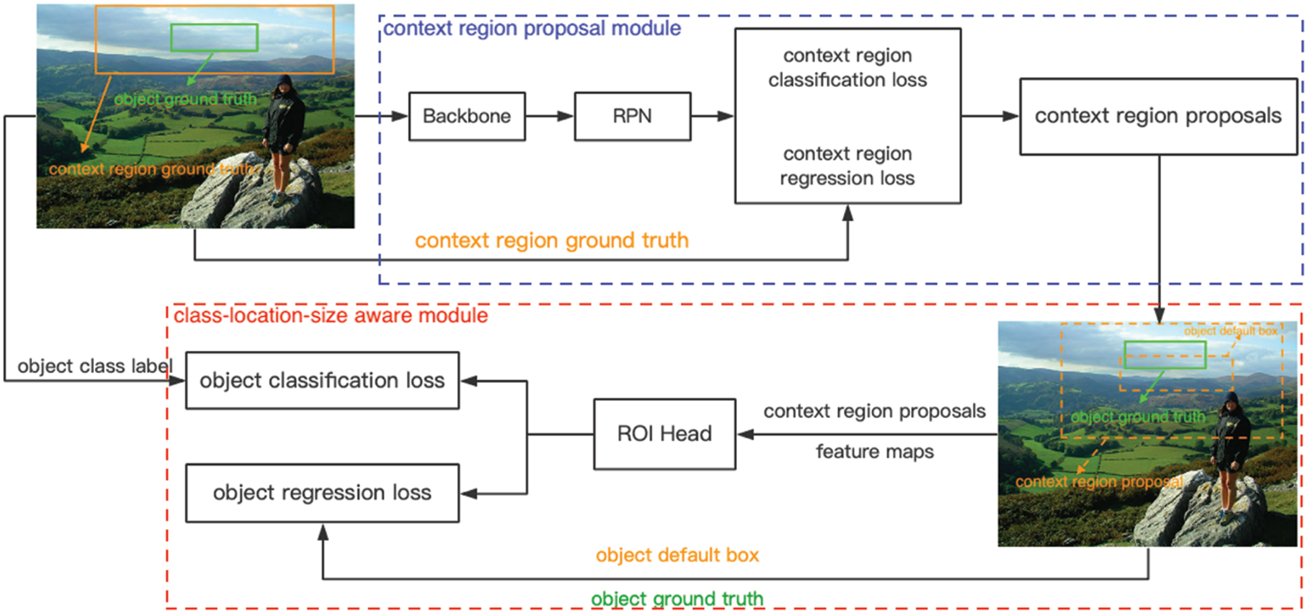
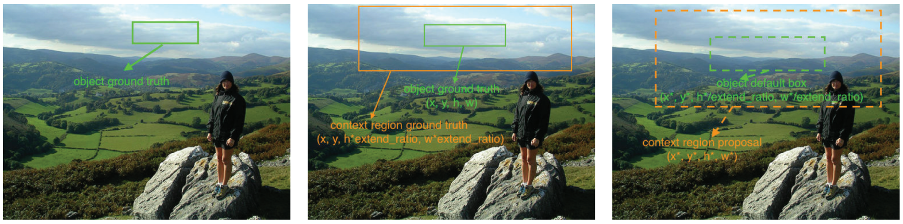


Figure 2 The detailed architecture of our approach.

branches: classification and regression. By predicting objectness confidence and offsets for anchor boxes with a variety of scales and ratios, they output a set of region proposals. These region proposals will be fed into classifier to predict the final class probabilities and offsets. However, the key difference between CRPM and RPN is the property of region proposals. Specifically, the region proposals predicted by RPN represent the true object regions that only contain the raw object and do not contain extra surrounding context information. In contrast, the region proposals outputted by our CRPM, named context region proposals, contain not only candidate placement regions (i.e., invisible-object region proposals) but also extra surrounding context information. In addition, the context region proposal have a fixed-ratio (a hyper parameter, e.g., *extend_ratio*) scaling relation with candidate placement regions. So we create a connection with shape between context region proposals and the candidate placement regions. In other words, we implicitly model the size information of placement regions into the context region proposals by the anchor mechanism. For instance, if the coordinate of context region proposal is (x, y, c_h, c_w), the candidate placement regions should be (x, y, $c\_h/extend\_ratio$, $c\_w/extend\_ratio$) (Figure 3(c)). The properties of the context region proposal have two advantages: (1) the candidate placement regions that can be thought to be prior default boxes make the CLSAM easier to learn refinedment offsets; (2) the surrounding visual context information


object ground truth
object ground truth
(x, y, h, w)
context region ground truth
(x, y, h*extend_ratio, w*extend_ratio)
object default box
(x*, y*, h*/extend_ratio, w*/extend_ratio)
context region proposal
(x*, y*, h*, w*)

(a).Placement ground truth (b).Context region ground truth (c).Default box

Figure 3 Training sample generation of context model: (a) Object placement ground truth annotated according to the placement rules. (b) Context region ground truth generated by extending the object placement ground truth. (c) Default placement box obtained by scaling the context region proposal.

will be the basis for CLASM to predict the final class probabilities of the candidate placement boxes.

3.2.2. The class-location-size aware module

The CLSAM takes the output of CRPM as input and outputs the class-location-size of final object placement regions. Similar to the roi-head in Faster R-CNN [9], it also has two branches. The first branch aims to predict class probabilities while the second branch is responsible for predicting offsets. However, the objective of their predicted offsets is very different. In roi-head, the predicted offsets are used to transform the object region proposals predicted by the RPN. In contrast, the regression branch of our CLSAM aims at predicting refinement shifts for the default boxes inside the context region proposals. As mentioned above, the default boxes are derived from context region proposals, and have a prior relation with context region proposals. Note that our default boxes are different from the previous default boxes proposed in SSD [20], or anchor boxes in Faster R-CNN [9], because its scale and aspect ratios are not fixed but determined by the context region proposals. Instead of directly predicting offsets for context region proposals, our method makes the learning process more stable and the offsets more refine.

3.2.3. The positive and negative generation

Although our context model has a same network architecture with Faster R-CNN [9], the definition of positive/negative examples and default boxes mechanism is different, which play a decisive role in the invisible-object detection.

In the work of [16], the positive examples were generated by cropping a sub-image which contained a masked ground truth from the raw image, and the negative examples were created by randomly sampling a masked region with its neighborhood from free space. As mentioned before, this strategy generates too much label noise and loses a lot of context information, even though it saves a lot of labor cost of labelling data. In contrast, we take the images with object bounding boxes annotated on free space in the Section 3.1 as the basic training data of our context model. Based on these bounding boxes, we design the strategies for CRPM and CLSAM respectively to generate corresponding training examples.

For CRPM, as shown in Figure 3(b), we create the context ground truth by extending the object ground truth on free space with the same aspect ratio. This forces the CRPM to learn the context region proposals which not only contain the candidate placement boxes inside but also have a fixed scaling ratio with candidate placement boxes. Afterward, we use the intersection-over-union (IoU) between anchor boxes and context ground truth to determine the positive and negative anchor examples.

For CLSAM, the positive examples must be simultaneously accord with two conditions: (1) the IoU between context region proposals and context region ground truth should exceed the corresponding thresh (i.e., 0.5); (2) the IoU between object default boxes and object ground truth should also exceed the corresponding thresh (0.2). The above conditions ensure that the sampled positive examples with the proper context and candidate object placement regions. For the negative examples, we only consider the IoU between context region proposals and context ground truth. If the IoU smaller than 0.5, the context region proposal will be considered to be a negative example.

Without any masked region, the positive examples are generated from the bounding boxes on free space, and the negative examples are generated from the remaining and unreasonable free space because of the dense ground truth on free space.

3.2.4. Objective function

The objective function of our context model consists of loss for CRPM and CLSAM, which is given by

$$L_{obj} = L_{CRPM} + L_{CLSAM} \tag{1}$$

where the L_{CRPM} and L_{CLSAM} are the loss for CRPM and CLSAM respectively. L_{CRPM} and L_{CLSAM} have a same mathematical expression, including classification loss and regression loss, is defined as

$$L = \frac{1}{N_{cls}} \sum_i l_{cls}(s_i, s_i *) + \lambda \frac{1}{N_{reg}} \sum_i p_i l_{reg}(t_i - t_i *) \tag{2}$$

where l_{cls} is the log loss and l_{reg} is the smooth $L1$ function. s_i and $s_i *$ are the predicted score and the ground truth label respectively. λ is a hyper parameter used to balancing the regression and classification loss. We adopt the default λ value used in Faster R-CNN. The p_i is 0 if the training sample is negative, and is 1 if the training sample is positive, which make the regression loss ignored when the training sample is negative. The t_i and $t_i *$ are the predicted offsets and offsets labels. The offsets encoding method is same as the Faster R-CNN. However, L_{CRPM} and L_{CLSAM} are very different.

In L_{CRPM}, the N_{cls} and N_{reg} are the number of context region proposals and anchor locations in a mini-batch respectively. s_i and $s_i *$ are the predicted context region proposal score and the ground truth label respectively. The predicted offsets is about the anchor box and context region ground truth.

In L_{CLSAM}, the N_{cls} and N_{reg} denote the number of total candidate placement regions and the positive candidate placement regions. The s_i is the predicted final placement score and $s_i *$ is the ground truth label. The offsets predicted by CLSAM are about the candidate placement region and placement ground truth.

3.2.5. Optimization parameters

The parameters need to be optimized involve convolutional kernels in CRPM and fully-connected layers in CLSAM. In CRPM, we take the VGG16 as the backbone network. The convolutional kernels of RPN are 512 × 3 × 3, 18 × 1 × 1, 36 × 1 × 1. In CLSAM, we set the roi-pooling size as 7 × 7. We take two 4096-dimension fully-connected layers to extract the final feature vector, and then use the 21-dimension (including background) and 84-dimension fully-connected layers to output the final classification scores and localization offsets, respectively. Therefore, the total number of parameter to be optimized in our context model is 137,069,376.

4. EXPERIMENTS

We will introduce the detail of our experiments in this section. Firstly, we state the basic dataset and evaluation in Section 4.1. Secondly, the implement details of all experiments is shown in Section 4.2. Thirdly, the Section 4.3 describe the process of our study about placement rules. Finally, the comparison experiments will be introduced in Section 4.4.

4.1. Datasets and Evaluation

Basic Datasets: All of our experiments related to a subset of PASCAL VOC (include VOC2007 and VOC2012) [26], which is a classic generic object detection public dataset containing 20 categories. The proposed placement rules and training data of context model involve a subset of VOC12 validation set, named *VOC12val-seg* (1449 images), which have segmentation annotations. Similar to [16], we use the subset of VOC12, named *VOC12train-seg* (1464 images), as training data of detection baseline. We select the *VOC07-test*, which is a test set of VOC2007 and contains 4952 images, to evaluate object detectors trained on different dataset *VOC07-test*.

Evaluation: We use Faster R-CNN [9], a classic two-stage detector, to compare our method with the state-of-the-art-related work [16]. In addition, we evaluate our approach based on YOLOv3 and Faster R-CNN with Feature Pyramid Networks (FPNs) [27] to validate the generalization of the proposed augmentation algorithm. As for evaluation metric, we choose the mean average precision (mAP) with IoU(0.5).

4.2. Implementation Details

Context model: In CRPM, we use the VGG16 [28] pre-trained on ImageNet [29] as the backbone network. When generate context region ground truth, we set the extend ratio 3.0. If the object placement ground truth is close to the image boundary, the context region ground truth will be out of boundary. In this case, we shift the context region ground truth inside the image instead of clipping. The reason is that we should guarantee the context region have enough contextual information. In inference phase, we set the confidence as 0.7 in all of experiments.

Training process of context model: The size of training image is 600 * 1000. Different from [16], the input of our context model is a whole image, not sub-image. We train our context-based model with batch size 1 for 14 epochs. A stochastic gradient descent (SGD) optimizer with 0.9 momentum is used. The start learning rate is 0.0001, which is decreased by a factor 5 for the later every 5 epoch.

Training process of object detectors: To be fair, we adopt a same open-source Faster R-CNN version [30] with [16]. The size of input image is 600 * 1000. We use the VGG16 pretrained on ImageNet as the backbone to extract feature maps. The only adopted traditional augmentation algorithm is flipping. In training, we also use the SGD with 0.9 momentum as optimizer, and 0.01 as the starting learning rate divided by 10 after 8 epochs. The detector is trained for 13 epochs in total.

For YOLOv3 [31], we first resize the input image to 416 * 416. The Darknet53 is adopted to extract feature maps. We train the detector within 13000 iterations by the SGD optimizer with 0.9 momentum. The initial learning rate is set 0.001, which is decreased to 0.0001 after 10000 iterations. In addition to YOLOv3, we apply the FPN component to the Faster R-CNN with a Resnet-101 [32] backbone to evaluate our method. We employ a same optimizer and initial learning rate with YOLOv3, containing 15 epochs in total. After 10 epochs, the learning rate is divided by 10.

Blending module: The final section of the copy-paste augmentation algorithm is that pasting foreground objects on the background images with blending algorithms and updating annotations according to the predicted placement regions. According to predicted category, we select one of the corresponding foreground objects, which is most similar to predicted region in shape. Specifically, the similar shape should meet two conditions: (1) the aspect ratio difference is less than 0.1; (2) the area difference is less than 0.2. Afterward, we resize the raw foreground object to the size of predicted region. The two conditions make the foreground object photorealism after resize. If the size of IoU formed by predicted region and raw object ground truth is more than 0.8, we will discard it and keep the origin annotation. Such an occlusion case is rare. For blending algorithm, we follow the work of [16], which means one of the following blending algorithms is randomly selected: (1) doing nothing; (2) generating blur on the whole background image; (3) adding Gaussianblur on the object boundaries.

4.3. Experiment of Placement Rules

In this section, we design an experiment to explore the placement rules for the copy-paste augmentation algorithms. As described in Section 3.1, we simplify the complicated placement rules into two items: (1) shifting two objects in same category together (e.g., the scene that the same kind of objects gathered together in a dense way); (2) making a strong correlation with surrounding objects

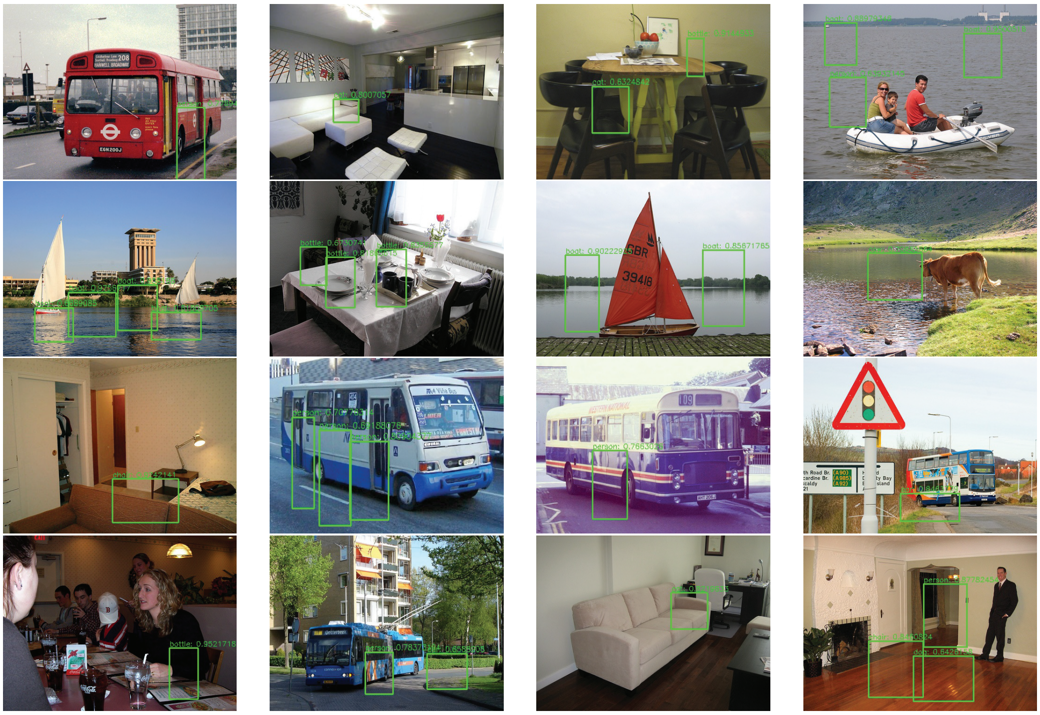


Figure 4 Visual examples of placement regions predicted by our method.

Table 2 The improvements of our placement rules.

Method (Faster R-CNN)	Aero	Bike	Bird	Boat	Bott.	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Pers.	Plant	Sheep	Sofa	Train	TV	mAP
Baseline	54.0	62.6	52.7	32.0	35.9	67.3	67.1	75.3	28.4	55.9	34.0	64.0	66.2	61.9	60.6	26.7	56.7	45.1	61.9	64.4	**53.6**
Our rules	58.2	66.4	57.1	40.2	40.5	66.9	69.7	78.7	35.3	60.2	38.3	68.6	67.9	63.5	62.6	30.0	60.1	47.4	63.2	64.0	**56.9**

of different category (e.g., the bottle should be on the table, people set on the chair, *etc.*). We apply the two simple placement rules to *VOC12val-seg* for generating the synthetic images, named *VOC12val-seg-syn*. Specifically, we first discard the raw images whose free space area is too small to place bounding boxes. Secondly, we annotate bounding boxes on the free space of the rest images and paste the foreground object onto these bounding boxes as the training examples of detectors. To evaluate the effectiveness of placement rules, we train the Faster R-CNN with VGG16 on *VOC12val-seg* and *VOC12val-seg-syn* respectively. The results are shown in Table 2. Based on the experiment results, we can observe that our placement rules improve the mAP by 3.3%, which demonstrate the effective of our rules. Therefore, we adopt the annotations on free space as ground truth to train our context model, and drive the context model to learn these rules.

4.4. Comparison with Other Methods

Comparison with state of the art: To demonstrate the outperform performance of our method,we compare the proposed method with a existing state-of-the-art approach which is named Context-DA [16]. Besides, we set the Faster R-CNN trained on VOC12train-seg as a baseline. The detail of the comparison results is shown in Table 3. Compared with baseline, our approach improve the mAP by 5.3% and still outperform the Context-DA by 3.4%. Note that the results of baseline and Context-DA are come from the Context-DA original paper [16].

Improvements on other detectors: In order to evaluate the generalization of our augmentation algorithm, we conduct some other experiments based on different baselines. Specifically, we still take the *VOC12train-seg* as the training dataset of the baseline. Besides, we add two new detectors (i.e.,YOLOv3 + Darknet53 and Faster R-CNN+FPN+Resnet-101). The experiment results indicate that our augmentation algorithm improves YOLOv3 and Faster R-CNN with FPN by 2.4% and 1.9% respectively. The results are shown in Table 4. Some visual example of placement regions predicted by our method are shown in Figure 4.

5. DISCUSSIONS AND FUTURE WORK

In this work, we firstly propose a placement rules on free space for the copy-paste augmentation algorithm. These simple yet effective rules can be seen as a solution to relieve the problem that few raw images can be obtained. Based on these rules, we further design a novel context model to learn these rules and automatically predict the proper placement regions for foreground objects. With the context model, we can automatically augment effective training object

Table 3 Comparison with state of the art.

Method (Faster R-CNN)	Aero	Bike	Bird	Boat	Bott.	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Pers.	Plant	Sheep	Sofa	Train	TV	mAP
Baseline	50.9	63.9	45.7	38.6	36.0	64.2	65.6	71.0	31.8	61.3	40.3	54.1	57.5	63.2	60.0	22.2	55.5	44.3	58.7	60.9	**52.3**
Context-DA [16]	56.1	66.2	48.2	41.1	40.7	61.8	66.1	66.7	34.7	61.5	37.3	62.6	64.8	63.4	61.8	25.6	57.9	45.7	63.8	59.0	**54.2**
Ours	58.6	70.2	51.3	45.3	42.3	68.7	68.8	71.2	32.3	68.0	46.6	61.2	69.5	66.3	65.8	33.3	62.6	46.8	62.6	59.5	**57.6**

Table 4 Improvements on Faster R-CNN+FPN and YOLOv3.

Method (Faster R-CNN+FPN)	Aero	Bike	Bird	Boat	Bott.	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Pers.	Plant	Sheep	Sofa	Train	TV	mAP
Baseline	67.5	77.6	71.8	56.2	53.8	78.0	77.5	85.5	46.0	75.8	59.0	80.4	78.6	74.6	74.0	40.3	70.7	62.4	75.9	69.2	**68.7**
Ours	70.3	78.4	76.3	58.1	56.0	77.6	79.1	88.2	46.3	76.5	58.4	83.1	78.9	76.3	79.8	43.6	75.4	63.5	75.2	71.7	**70.6**
Method (YOLOv3)	**Aero**	**Bike**	**Bird**	**Boat**	**Bott.**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Pers.**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**mAP**
Baseline	67.5	71.9	61.6	51.9	37.6	73.1	73.0	80.4	34.5	67.8	54.9	72.0	74.3	65.4	67.6	33.5	65.2	57.0	75.1	62.4	**62.3**
Ours	68.3	74.3	68.4	51.1	42.1	75.2	72.0	82.8	36.2	72.7	57.7	73.0	78.7	68.9	69.9	34.6	70.4	59.6	76.4	62.1	**64.7**

instances with annotations on images without any labor and limitation on the number of images. However, a limitation of our context model is that it need some annotations on free space. Therefore, we will explore the comprehensive prior relationships about different categories in the future, and then associate with the detector to achieve the target of augmenting more effective training examples for detectors without any extra labeling annotations. In addition, we will evaluate the generalization of our method on some other tasks, such as segmentation and special object detection.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

Conceptualization, Jun Zhang and Feiteng Han; Methodology, Feiteng Han and Kangwei Liu; Writing–review and editing, Jun Zhang, Feiteng Han and Yutong Chun; Project administration, Yutong Chun; Resources, Wang Chen. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

This research was funded by National Social Science Fund of China grant number 20BGL001.

REFERENCES

[1] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, K. Weinberger, Pseudolidar from visual depth estimation: bridging the gap in 3d object detection for autonomous driving, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[2] Z. Zhe, D. Liang, S. Zhang, X. Huang, S. Hu, Traffic-sign detection and classification in the wild, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.

[3] B. Li, W. Ouyang, L. Sheng, X. Zeng, X. Wang, Gs3d: an efficient 3d object detection framework for autonomous driving, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[4] P. Li, X. Chen, S. Shen, Stereo R-CNN based 3d object detection for autonomous driving, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[5] J. Deng, J. Guo, X. Niannan, S. Zafeiriou, Arcface: additive angular margin loss for deep face recognition, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[6] M. Najibi, B. Singh, L.S. Davis, Farpn: floating region proposals for face detection, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[7] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, F. Huang, DSFD: dual shot face detector, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019.

[8] G. Brazil, X. Liu, Pedestrian detection with autoregressive network phases, in Proceeding of IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019.

[9] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017), 1137–1149.

[10] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New York, USA, 2020.

[11] D. Dwibedi, I. Misra, M. Hebert, Cut, paste and learn: surprisingly easy synthesis for instance detection, in The IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017.

[12] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[13] G. Georgakis, A. Mousavian, A.C. Berg, J. Kosecka, Synthesizing training data for object detection in indoor scenes, CoRR, abs/1702.07836, 2017.

[14] H. Su, X. Zhu, S. Gong, Deep learning logo detection with data expansion by synthesising context, in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, Santa Rosa, CA, USA, 2017, pp. 530–539.

[15] N. Dvornik, J. Mairal, C. Schmid, Modeling visual contesxt is key to augmenting object detection datasets, in IEEE Europe Conference on Computer Vision (ECCV), Munich, Germany, 2018.

[16] N. Dvornik, J. Mairal, C. Schmid, On the importance of visual context for data augmentation in scene understanding, CoRR, abs/1809.02492, 2018.

[17] A. Oliva, A. Torralba, The role of context in object recognition, Trends Cogn. Sci. 11 (2007), 520–527.

[18] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, M. Hebert, An empirical study of context in object detection, in Proceedings/CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 1271–1278.

[19] R. Mottaghi, X. Chen, X. Liu, N.G. Cho, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Lecture Notes in Computer Science, Springer, Cham, Netherlands, 2016.

[21] R. Takahashi, T. Matsubara, K. Uehara, Data augmentation using random image cropping and patching for deep cnns, in IEEE Trans. Circuits Syst. Video Technol. 30 (2018), 2917–2931.

[22] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, S. Birchfield, Training deep networks with synthetic data: bridging the reality gap by domain randomization, in 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, IEEE Computer Society, Salt Lake City, UT, USA, 2018, pp. 969–977.

[23] H. Wang, Q. Wang, F. Yang, W. Zhang, W. Zuo, Data augmentation for object detection via progressive and selective instance-switching, CoRR, abs/1906.00358, 2019.

[24] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection, CoRR, abs/1906.11172, 2019.

[25] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J.M. Rehg, V. Chari, Learning to generate synthetic data via compositing, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019), Computer Vision Foundation/IEEE, Long Beach, CA, USA, 2019, pp. 461–470.

[26] M. Everingham, S.M. Ali Eslami, L. Van Gool, C.K.I. Williams, J.M. Winn, A. Zisserman, The pascal visual object classes challenge: a retrospective, Int. J. Comput. Vision. 111 (2015), 98–136.

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 2117–2125.

[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in 3rd International Conference on Learning Representations (ICLR 2015), Conference Track Proceedings, San Diego, CA, USA, 2015.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, Imagenet: a large-scale hierarchical image database, in 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), IEEE Computer Society, Miami, FL, USA, 2009, pp. 248–255.

[30] J. Yang, J. Lu, D. Batra, D. Parikh, A faster pytorch implementation of faster R-CNN, 2017. https://github.com/jwyang/faster-rcnn.pytorch.

[31] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv, 2018.

[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognitionin Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 770–778.