Modeling Visual Context is Key to Augmenting Object Detection Datasets

Nikita Dvornik, Julien Mairal, Cordelia Schmid

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France
firstname.lastname@inria.fr

Abstract. Performing data augmentation for learning deep neural networks is well known to be important for training visual recognition systems. By artificially increasing the number of training examples, it helps reducing overfitting and improves generalization. For object detection, classical approaches for data augmentation consist of generating images obtained by basic geometrical transformations and color changes of original training images. In this work, we go one step further and leverage segmentation annotations to increase the number of object instances present on training data. For this approach to be successful, we show that modeling appropriately the visual context surrounding objects is crucial to place them in the right environment. Otherwise, we show that the previous strategy actually hurts. With our context model, we achieve significant mean average precision improvements when few labeled examples are available on the VOC'12 benchmark.

Keywords: Object Detection, Data Augmentation, Visual Context

1 Introduction

Object detection is one of the most classical computer vision task and is often considered as a basic proxy for scene understanding. Given an input image, an algorithm is expected to produce a set of tight boxes around objects while automatically classifying them. Obviously, modeling correctly object appearances is important, but it is also well-known that visual context provides important cues for recognition, both for computer vision systems and for humans [1].

Objects from the same class tend indeed to be grouped together in similar environments; sometimes they interact with it and do not even make sense in its absence. Whenever visual information is corrupted, ambiguous, or incomplete (*e.g.*, an image contains noise, bad illumination conditions, or an object is occluded or truncated), visual context becomes a crucial source of information. Frequently, certain object categories may for instance most often appear in specific conditions (*e.g.*, planes in the sky, plates on the table), in co-occurrence with objects of other specific classes (*e.g.*, baseball ball and baseball bat), and more generally, any type of clue for object recognition that is not directly related

* Institute of Engineering Univ. Grenoble Alpes

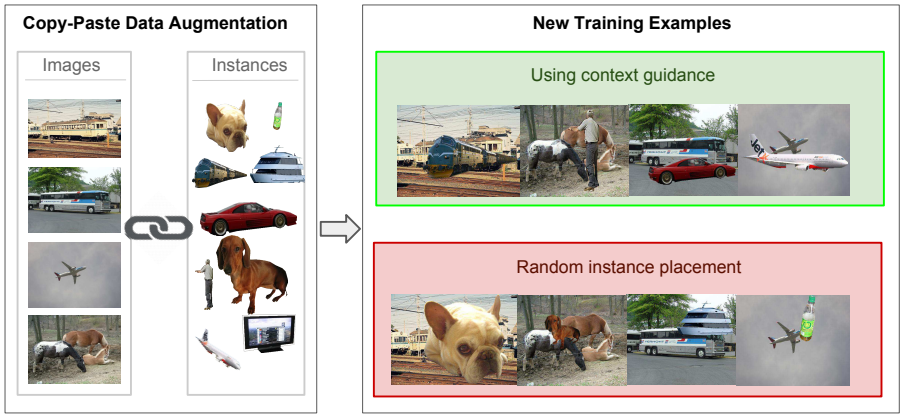

Fig. 1. Examples of data-augmented training examples produced by our approach. Images and objects are taken from the VOC'12 dataset that contains segmentation annotations. We compare the output obtained by pasting the objects with our context model vs. those obtained with random placements. Even though the results are not perfectly photorealistic and display blending artefacts, the visual context surrounding objects is more often correct with the explicit context model.

to the object's appearance is named "context" in the literature. For this reason, a taxonomy of contextual information is proposed in [2] to better understand what type of visual context is useful for object detection.

Before the deep learning/ImageNet revolution, the previous generation of object detectors such as [3–6] modeled the interaction between object locations, categories, and context by manual engineering of local descriptors, feature aggregation methods, and by defining structural relationship between objects. In contrast, recent works based on convolutional neural networks such as [7–10] implicitly model visual context by design since the receptive field of "artificial neurons" grows with the network's depth, eventually covering the full image for the last layers. For this reason, these CNNs-based approaches have shown modest improvements when combined with an explicit context model [11].

Our results are not in contradiction with such previous findings. We show that explicit context modeling is important only for a particular part of object detection pipelines that was not considered in previous work. When training a convolutional neural network, it is indeed important to control overfitting, especially if few labeled training examples are available. Various heuristics are typically used for that purpose such as DropOut [12], penalizing the norm of the network parameters (weight decay), or early stopping. Even though the regularization effect of such approaches is not well understood from a theoretical point of view, these heuristics have been found to be useful in practice.

Besides these heuristics related to the learning procedure, another way to control overfitting consists of artificially increasing the size of training data by using

prior knowledge on the task. For instance, all object classes from the VOC'12 dataset [13] are invariant to horizontal flips (*e.g.*, a flipped car is still a car) and to many less-trivial transformations. A more ambitious data augmentation technique consists of leveraging segmentation annotations, either obtained manually, or from an automatic segmentation system, and create new images with objects placed at various positions in existing scenes [14–16]. While not achieving perfect photorealism, this strategy with random placements has proven to be surprisingly effective for *object instance detection* [14], which is a fine-grained detection task consisting of retrieving instances of a particular object from an image collection; in contrast, *object detection* focuses on detecting object instances from a particular category. Unfortunately, the random-placement strategy does not extend to the object detection task, as shown in the experimental section. By placing training objects at unrealistic positions, implicitely modeling context becomes difficult and the detection accuracy drops substantially.

Along the same lines, the authors of [15] have proposed to augment datasets for text recognition by adding text on images in a realistic fashion. There, placing text with the right geometrical context proves to be critical. Significant improvements in accuracy are obtained by first estimating the geometry of the scene, before placing text on an estimated plane. Also related, the work of [16] is using successfully such a data augmentation technique for object detection in indoor scene environments. Modeling context has been found to be critical as well and has been achieved by also estimating plane geometry and objects are typically placed on detected tables or counters, which often occur in indoor scenes.

In this paper, we consider the general object detection problem, which requires more generic context modeling than estimating plane and surfaces as done for instance in [15, 16]. To this end, the first contribution of our paper is methodological: we propose a context model based on a convolutional neural network, which will be made available as an open-source software package. The model estimates the likelihood of a particular category of object to be present inside a box given its neighborhood, and then automatically finds suitable locations on images to place new objects and perform data augmentation. A brief illustration of the output produced by this approach is presented in Figure 1. The second contribution is experimental: We show with extensive tests on the VOC'12 benchmark that context modeling is in fact a key to obtain good results for object detection and that substantial improvements over non-data-augmented baselines may be achieved when few labeled examples are available.

2 Related Work

In this section, we briefly discuss related work for visual context modeling and data augmentation for object detection.

Modeling visual context for object detection. Relatively early, visual context has been modeled by computing statistical correlation between low-level features of the global scene and descriptors representing an object [17, 18]. Later, the

authors of [4] introduced a simple context re-scoring approach operating on appearance-based detections. To encode more structure, graphical models were then widely used in order to jointly model appearance, geometry, and contextual relations [19, 20]. Then, deep learning approaches such as convolutional neural networks started to be used [7–9]; as mentioned previously, their features already contain implicitly contextual information. Yet, the work of [21] explicitly incorporates higher-level context clues and combines a conditional random field model with detections obtained by Faster-RCNN. With a similar goal, recurrent neural networks are used in [22] to model spatial locations of discovered objects. Another complementary direction in context modeling with convolutional neural networks use a deconvolution pipeline that increases the field of view of neurons and fuse features at different scales [22–24], showing better performance essentially on small objects. The works of [2, 25] analyze different types of contextual relationships, identifying the most useful ones for detection, as well as various ways to leverage them. However, despite these efforts, an improvement due to purely contextual information has always been relatively modest [11, 26].

Data augmentation for object detection. Data augmentation is a major tool to train deep neural networks. If varies from trivial geometrical transformations such as horizontal flipping, cropping with color perturbations, and adding noise to an image [27], to synthesizing new training images [28, 29]. Some recent object detectors [9, 10, 23] benefit from standard data augmentation techniques more than others [7, 8]. The performance of Fast- and Faster-RCNN could be for instance increased by simply corrupting random parts of an image in order to mimic occlusions [30]. Regarding image synthesis, recent works such as [31–33] build and train their models on purely synthetic rendered 2d and 3d scenes. However, a major difficulty for models trained on synthetic images is to guarantee that they will generalize well to real data since the synthesis process introduces significant changes of image statistics [29]. To address this issue, the authors of [15] adopt a different direction by pasting real segmented object into natural images, which reduces the presence of rendering artefacts. For object instance detection, the work [16] estimates scene geometry and spatial layout, before placing objects in the image to create realistic training examples. In [14], the authors propose an even simpler solution to the same problem by pasting images in random positions but modeling well occluded and truncated objects, and making the training step robust to boundary artifacts at pasted locations.

3 Modeling Visual Context for Data Augmentation

Our approach for data augmentation mainly consists of two parts: we first model visual context by using bounding box annotations, where the surrounding of a box is used as an input to a convolutional neural network to predict the presence or absence of an object within the box. Then, the trained context model is used to generate a set of possible new locations for objects. The full pipeline is presented in Fig. 2. In this section, we describe these two steps in details, but before that, we present and discuss a preliminary experiment that has motivated our work.

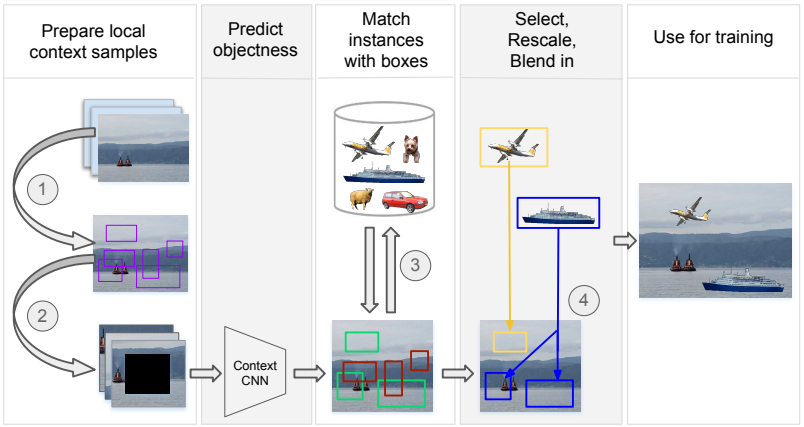

Fig. 2. Illustration of our data augmentation approach. We select an image for augmentation and 1) generate 200 candidate boxes that cover the image. Then, 2) for each box we find a neighborhood that contains the box entirely, crop this neighborhood and mask all pixels falling inside the bounding box; this "neighborhood" with masked pixels is then fed to the context neural network module and 3) object instances are matched to boxes that have high confidence scores for the presence of an object category. 4) We select at most two instances that are rescaled and blended into the selected bounding boxes. The resulting image is then used for training the object detector.

3.1 Preliminary Experiment with Random Positioning

In [14], data augmentation is performed by placing segmented objects at random positions in new scenes. As mentioned previously, the strategy was shown to be effective for object instance detection, as soon as an appropriate procedure is used for preventing the object detector to overfit blending artefacts—that is, the main difficulty is to prevent the detector to "detect artefacts" instead of detecting objects of interest. This is achieved by using various strategies to smooth object boundaries such as Poisson blending [34], and by adding "distractors" objects that do not belong to any of the dataset categories, but which are also synthetically pasted on random backgrounds. With distractors, artefacts occur both in positive and negative examples, preventing the network trained for object detection to overfit them. According to [14], this strategy brings substantial improvements for the object instance detection/retrieval task, where modeling the fine-grain appearance of an object instance seems to be more important than modeling visual context as in the general category object detection task.

Unfortunately, the above context-free strategy does not extend trivially to the object detection task we consider. Our preliminary experiment conducted on the VOC'12 dataset actually shows that it may even hurt the accuracy of the detector, which has motivated us to propose instead an explicit context model. Specifically, we conducted an experiment by following the original strategy of [14]

as closely as possible. We use the subset of the VOC'12 train set that has ground-truth segmentation annotations to cut object instances from images and then place them on other images from the training set. As in [14], we experimented with various blending strategies (Gaussian or linear blur, Poisson blending, or using no blending at all) to smooth the boundary artifacts. Following [14], we also considered "distractors", which are then labeled as background. Distractors were simply obtained by copy-pasting segmented objects from the COCO dataset [35] from categories that do not appear in VOC'12.[1]

For any combination of blending strategy, by using distractors or not, the naive data augmentation approach with random placement did not improve upon the baseline without data augmentation for the classical object detection task. A possible explanation may be that for instance object detection, the detector does not need to learn intra-class variability of object/scene representations and seems to concentrate only on appearance modeling of specific instances, which is not the case for category-level object detection. This experiment was the key motivation for proposing a context model, which we now present.

3.2 Modeling Visual Context with Convolutional Neural Networks

Since the context-free data augmentation failed, we propose to learn where to automatically place objects by using a convolutional neural network. Here, we present the data generation, model training, and object placement procedures.

Contextual data generation. We consider training data with bounding box and category annotations. For each bounding box B associated to a training image I, we create a set of training contexts, which are defined as subimages of I fully enclosing the bounding box B whose content is masked out, as illustrated in Figure 3. Several contexts can be created from a single annotated bounding box B by varying the size of the subimage around B and its aspect ratio. In addition, "background" contexts are also created by considering random bounding boxes whose intersection over union with any ground truth doesn't exceed a threshold of 0.3, and whose content is also masked out. The shape of such boxes is defined by aspect ratio a and relative scale s. We draw a pair of parameters from the joint distribution induced by bounding boxes containing positive objects, i.e. a 30×30 bins normalized histogram. Since in general, there is more background samples than the ones actually containing objects, we sample "background" contexts 3 times more often following sampling strategies in [7, 9].

Model training. Given the set of all contexts, gathered from all training data, we train a convolutional neural network to predict the presence of each object in the masked bounding box. The input to the network are the "contextual images" obtained during the data generation step, and which contain a masked bounding box inside. These contextual images are resized to 300×300 pixels,

[1] Note that external data from COCO was used only in this preliminary experiment and not in the experiments reported later in Section 4.

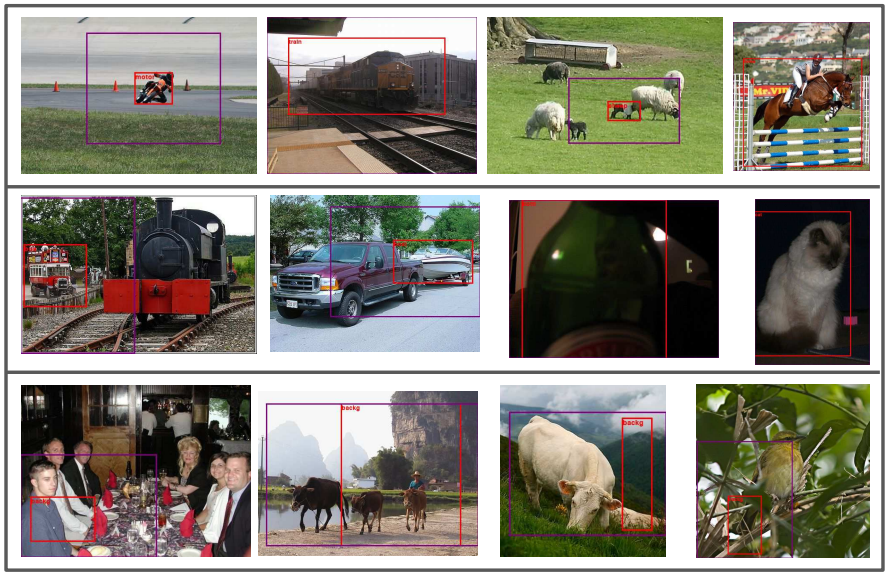

Fig. 3. Contextual images - examples of inputs to the context model. A subimage bounded by a magenta box is used as an input to the context model after masking-out the object information inside a red box. The top row lists examples of positive samples encoding real objects surrounded by regular and predictable context. Positive training examples with ambiguous or uninformative context are given in the second row. The bottom row depicts negative examples enclosing background. This figure shows that contextual images could be ambiguous to classify correctly and the task of predicting the category given only the context is challenging.

Fig. 4. Different kinds of blending used in experiments. From left to right: linear smoothing of boundaries, Gaussian smoothing, no processing, motion blur of the whole image, Poisson blending [34].

and the output of the network is a label in a set $\{1, 2, ..., K + 1\}$, where K is the number of object categories and the $(K+1)$-th class represents background. For such a multi-class image classification problem here, we use the classical ResNet50 network [36] pre-trained on ImageNet, and change the last layer to be a softmax with $K + 1$ activations (see experimental section for details).

Selection of object locations at test time. Once the context model is trained by using training data annotated with bounding boxes, we use it to select locations to perform data augmentation on a given image. As input, the trained classifier receives "contextual images" with a bounding box masked out (as in Section 3.2). The model is able to provide a set of "probabilities" representing the presence of each object category in a given bounding box, by considering its visual surrounding. Since evaluating all potential bounding boxes from an image is too costly, we randomly draw 200 candidate bounding boxes and retain the ones where an object category has a score greater than 0.8; empirically, the number 200 was found to provide good enough bounding boxes among the top scoring ones, while resulting in a reasonably fast data augmentation procedure.

Blending objects in their environment. Whenever a bounding box is selected by the previous procedure, we need to blend an object at the corresponding location. This step follows closely the findings of [14]. We consider different types of blending techniques (Gaussian or linear blur, simple copy-pasting with no post-processing, or generating blur on the whole image to imitate motion), and randomly choose one of them in order to introduce a larger diversity of blending artefacts. We also do not consider Poisson blending in our approach, which was considerably slowing down the data generation procedure. Unlike [14] and unlike our preliminary experiment described in Section 3.1, we do not use distractors, which were found to be less important for our task than in [14]. As a consequence, we do not need to exploit external data to perform data augmentation. Qualitative results are illustrated on Figure 4.

4 Experiments

In this section, we present experiments demonstrating the importance of context modeling for data augmentation. We evaluate our approach on the subset of the VOC'12 dataset that contains segmentation annotations, and study the impact of data augmentation when changing the amount of training data. In Section 4.1, we present data, tools, and evaluation metrics. In Section 4.2, we present implementation details that are common to all experiments, in order to make our results reproducible (the source code to conduct our experiments will also be made publicly available in an open-source software package). First, we present experiments for object detectors trained on single categories in Section 4.3—that is, detectors are trained individually for each object category, and an experiment for the standard multiple-category setting is presented in Section 4.4. Finally, we present an ablation study in Section 4.5 to understand the effect of various factors (blending and placement strategies, amount of labeled data).

4.1 Dataset, Tools, and Metrics

Dataset. In all our experiments, we use a subset of the Pascal VOC'12 training dataset [13] that contains segmentation annotations to train all our models (context-model and object detector). We call this training set `VOC12train-seg`, which contains 1 464 images. Following standard practice, we use the test set of VOC'07 to evaluate the models, which contains 4 952 images with the same 20 object categories as VOC'12. We call this image set `VOC07-test`.

Object detector. To test our data-augmentation strategy we chose one of the state-of-the art object detectors with open-source implementation, BlitzNet [23] that achieves 79.1% mAP on `VOC07-test` when trained on the union of the full training and validation parts of VOC'07 and VOC'12, namely `VOC07-train+val` and `VOC12train+val` (see [23]); this network is similar to the DSSD detector of [24] that was also used in the Focal Loss paper [37]. The advantage of such class of detectors is that it is relatively fast (it may work in real time) and supports training with big batches of images without further modification.

Evaluation metric. In VOC'07, a bounding box is considered to be correct if its Intersection over Union (IoU) with a ground truth box is higher than 0.5. The metric for evaluating the quality of detection for one object class is the average precision (AP), and the mean average precision (mAP) for the dataset.

4.2 Implementation Details

Selecting and blending objects. Since we widely use object instances extracted from the training images in all our experiments, we create a database of objects cut out from the `VOC12train-seg` set to quickly access them during training. For a given candidate box, an instance is considered as matching if after scaling it by a factor in $[0.5, 1.5]$ the re-scaled instance's bounding box fits inside the candidate's one and takes at least 80% of its area. When blending them into the new background, we follow [14] and use randomly one of the following methods: adding Gaussian or linear blur on the object boundaries, generating blur on the whole image by imitating motion, or just paste an image with no blending. To not introduce scaling artifacts, we keep the scaling factor close to 1.

Training the context model. After preparing the "contextual images" as described in 3.2, we re-scale them to the standard size 300×300 and stack them in batches of size 32. We use ResNet50 [36] with ImageNet initialization to train a contextual model in all our experiments. Since we have access only to the training set at any moment we train and apply the model on the same data. To prevent overfitting, we use early stopping. In order to determine when to stop the training procedure, we monitor both training error on our training set and validation error on the VOC'12 validation set `VOC12-val`. The moment when the loss curves start diverging noticeably is used as a stopping point. To this

end, when building context model for one class vs. background, we train a network for 1.5K iterations, then decrease the learning rate by a factor 10 and train for 500 additional iterations. When learning a joint contextual model for all 20 categories, we first run the training procedure for 4K iterations and then for 2K more iterations after decreasing the learning rate. We sample 3 times more background contextual images, as noted in Section 3.2. Visual examples of images produced by the context model are presented in Figure 5. Overall, training the context model is about 5 times faster than training the detector.

Training the object detector. In this work, the detector takes images of size 300×300 as an input and produces a set of candidate object boxes with classification scores; like our context model, it uses ResNet50 [36] pre-trained on ImageNet as a backbone. The detector is trained by following [23], with the ADAM optimizer [38] starting from learning rate 10^{-4} and decreasing it later during training by a factor 10 (see Sections 4.3 and 4.4 for the number of epochs used in each experiment). In addition to our data augmentation approach obtained by copy-pasting objects, all experiments also include classical data augmentation steps obtained by random-cropping, flips, and color transformations, following [23].

4.3 Single-Category Object Detection

In this section, we conduct an experiment to better understand the effect of the proposed data augmentation approach, dubbed "Context-DA" in the different tables, when compared to a baseline with random object placement "Random-DA", and when compared to standard data augmentation techniques called "Base-DA". The study is conducted in a single-category setting, where detectors are trained independently for each object category, resulting in a relatively small number of positive training examples per class. This allows us to evaluate the importance of context when few labeled samples are available and see if conclusions drawn for a category easily generalize to other ones.

The baseline with random object placements on random backgrounds is conducted in a similar fashion as our context-driven approach, by following the strategy described in the previous section. For each category, we treat all images with no object from this category as background images, and consider a collection of cut instances as discussed in Section 4.1. During training, we augment a negative (background) image with probability 0.5 by pasting up to two instances on it, either at randomly selected locations (Random-DA), or using our context model in the selected bounding boxes with top scores (Context-DA). The instances are re-scaled by a random factor in $[0.5, 2]$ and blended into an image using a randomly selected blending method mentioned in Section 4.1. For all models, we train the object detection network for 6K iterations and decrease the learning rate after 2K and 4K iterations by a factor 10 each time. The results for this experiment are presented in Table 1.

The conclusions are the following: random placement indeed hurts the performance on average. Only the category bird seems to benefit significantly from it, perhaps because birds tend to appear in various contexts in this dataset and

some categories significantly suffer from random placement such as boat, table, and sheep. Importantly, the visual context model always improve upon the random placement one, on average by 5%, and upon the baseline that uses only classical data augmentation, on average by 4%. Interestingly, we identify categories for which visual context is crucial (aeroplane, bird, boat, bus, cat, cow, horse), for which context-driven data augmentation brings more than 5% improvement and some categories that display no significant gain or losses (chair, table, persons, train), where the difference with the baseline is less than 1%.

Table 1. Comparison of detection accuracy on `VOC07-test` for the single-category experiment. The models are trained independently for each category, by using the 1 464 images from `VOC12train-seg`. The first row represents the baseline experiment that uses standard data augmentation techniques. The second row uses in addition copy-pasting of objects with random placements. The third row presents the results achieved by our context-driven approach and the last row presents the improvement it brings over the baseline. The numbers represent average precision per class in %. Large improvements over the baseline (greater than 5%) are in bold.

method	aero	bike	bird	boat	bott.	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	avg.
Base-DA	58.8	64.3	48.8	47.8	33.9	66.5	69.7	68.0	40.4	59.0	61.0	56.2	72.1	64.2	66.7	36.6	54.5	53.0	73.4	63.6	58.0
Random-DA	60.2	66.5	55.1	41.9	29.7	66.5	70.0	70.1	37.4	57.4	45.3	56.7	68.3	66.1	67.0	37.0	49.9	55.8	72.1	62.6	56.9
Context-DA	67.0	68.6	60.0	53.3	38.8	73.3	72.4	74.3	39.7	64.3	61.4	60.3	77.6	69.0	67.3	38.6	56.2	56.9	74.4	66.8	62.0
Impr. Cont.	**8.2**	4.3	**11.2**	**5.5**	4.9	**6.8**	2.7	**6.3**	-0.7	**5.3**	0.4	4.1	**5.5**	4.8	0.6	2.0	1.7	3.9	1.0	3.2	4.0

Table 2. Comparison of detection accuracy on `VOC07-test` for the multiple-category experiment. The model is trained on all categories at the same time, by using the 1 464 images from `VOC12train-seg`. The first row represents the baseline experiment that uses standard data augmentation techniques. The second row uses also our context-driven data augmentation. The numbers represent average precision per class in %.

method	aero	bike	bird	boat	bott.	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	avg.
Base-DA	63.6	73.3	63.2	57.0	31.5	76.0	71.5	79.9	40.0	71.6	61.4	74.6	80.9	70.4	67.9	36.5	64.9	63.0	79.3	64.7	64.6
Context-DA	66.8	75.3	65.9	57.2	33.1	75.0	72.4	79.6	40.6	73.9	63.7	77.1	81.4	71.8	68.1	37.9	67.6	64.7	81.2	65.5	65.9

4.4 Multiple-Categories Object Detection

In this section, we conduct the same experiment as in 4.3, but we train a single multiple-category object detector instead of independent ones per category. Network parameters are trained with more labeled data (on average 20 times more than for models learned in Table 4.3). The results are presented in Table 2 and show a modest improvement of 1.3% on average over the baseline, which is relatively consistent across categories, with 18 categories out of 20 that benefit from the context-driven data augmentation. This confirms that data augmentation is crucial when few labeled examples are available.

4.5 Ablation Study

Finally, we conduct an ablation study to better understand (i) the importance of visual context for object detection, (ii) the impact of blending artefacts, and (iii) the importance of data augmentation when using very few labeled examples. For simplicity, we choose the first 5 categories of VOC'12, namely *aeroplane, bike, bird, boat, bottle*, and train independent detectors per category as in Section 4.3, which corresponds to a setting where few samples are available for training.

Baseline when no object is in context. Our experiments show that augmenting naively datasets with randomly placed objects slightly hurts the performance. To confirm this finding, we consider a similar experiment, by learning on the same number of instances as in Section 4.3, but we consider as positive examples only objects that have been synthetically placed in a random context. This is achieved by removing from the training data all the images that have an object from the category we want to model, and replacing it by an instance of this object placed on a background image. The main motivation for such study is to consider the extreme case where (i) no object is placed in the right context; (ii) all objects may suffer from rendering artefacts. As shown in Table 3, the average precision degrades significantly by about 14% compared to the baseline. As a conclusion, either visual context is indeed crucial for learning, or blending artefacts is also a critical issue. The purpose of the next experiment is to clarify this ambiguity.

Impact of blending when the context is right. In the previous experiment, we have shown that the lack of visual context and the presence of blending artefacts may explain the performance drop observed on the fourth row of Table 3. Here, we propose a simple experiment showing that blending artefacts are not critical when objects are placed in the right context: the experiment consists of extracting each object instance from the dataset, up-scale it by a random factor slightly greater than one (in the interval $[1.2, 1.5]$), and blend it back at the same location, such that it covers the original instance. As a result, the new dataset benefits slightly from data augmentation (thanks to object enlargement), but it also suffers from blending artefacts for *all object instances*. As shown on the fifth row of Table 3, this approach improves over the baseline, though not as much as the full context-driven data augmentation, which suggests that the lack of visual context was the key explaining the result observed before. The experiment also confirms that the presence of blending artefacts is not critical for the object detection task. Visual examples of such artefacts are presented in Figure 6.

Performance with very few labeled data. Finally, the last four rows of Table 3 present our results when reducing the amount of labeled data, in a setting where this amount is already small when using all training data. The improvement provided by our approach is significant and consistent (about 6% when using only 50% and 25% of the training data). Even though one may naturally expect larger improvements when a very small number of training examples are available, it should be noted that in such very small regimes, the quality of the context model

may degrade as well (e.g., the dataset contains only 87 images of birds, meaning that with 25%, we use only 22 images with positive instances).

Table 3. Ablation study on the first five categories of VOC'12. All models are learned independently as in Table 1. We compare classical data augmentation techniques (Base-DA), approaches obtained by copy-pasting objects, either randomly (Random-DA) or according to a context model (Context-DA). The line "Removing context" corresponds to the first experiment described in Section 4.5; Enlarge-Reblend corresponds to the second experiment, and the last four rows compare the performance of Base-DA and Context-DA when varying the amount of training data from 50% to 25%.

Data portion	aero	bike	bird	boat	bottle	average
Base-DA	58.8	64.3	48.8	47.8	33.9	48.7
Random-DA	60.2	66.5	55.1	41.9	29.7	48.3
Context-DA	67.0	68.6	60.0	53.3	38.8	57.5
Removing context	44.0	46.8	42.0	20.9	15.5	33.9
Enlarge + Reblend-DA	60.1	63.4	51.6	48.0	34.8	51.6
Base-DA 50 %	55.6	60.1	47.6	40.1	21.0	42.2
Context-DA 50 %	62.2	65.9	55.2	46.9	27.2	48.8
Base-DA 25 %	51.3	54.0	33.8	28.2	14.0	32.5
Context-DA 25 %	57.8	59.5	40.6	34.3	19.0	38.3

5 Discussions and Future Work

In this paper, we introduce a data augmentation technique dedicated to object detection, which exploits segmentation annotations. From a methodological point of view, we show that this approach is effective and goes beyond traditional augmentation approaches. One of the keys to obtain significant improvements in terms of accuracy was to introduce an appropriate context model which allows us to automatically find realistic locations for objects, which can then be pasted and blended at in the new scenes. While the role of explicit context modeling has been unclear so far for object detection, we show that it is in fact crucial when performing data augmentation and learn with few labeled data, which is one of the major issue that deep learning models are facing today.

We believe that these promising results pave the way to numerous extensions. In future work, we will for instance study the application of our approach to other scene understanding tasks, e.g., semantic or instance segmentation, and investigate how to adapt it to larger datasets. Since our approach relies on pre-segmented objects, which are subsequently used for data augmentation, we are also planning to exploit automatic segmentation tools such as [39] in order to use our method when only bounding box annotations are available.

Acknowledgments. This work was supported by a grant from ANR (MACARON project ANR-14-CE23-0003-01), by the ERC grant 714381 (SOLARIS project), the ERC advanced grant ALLEGRO and gifts from Amazon and Intel.

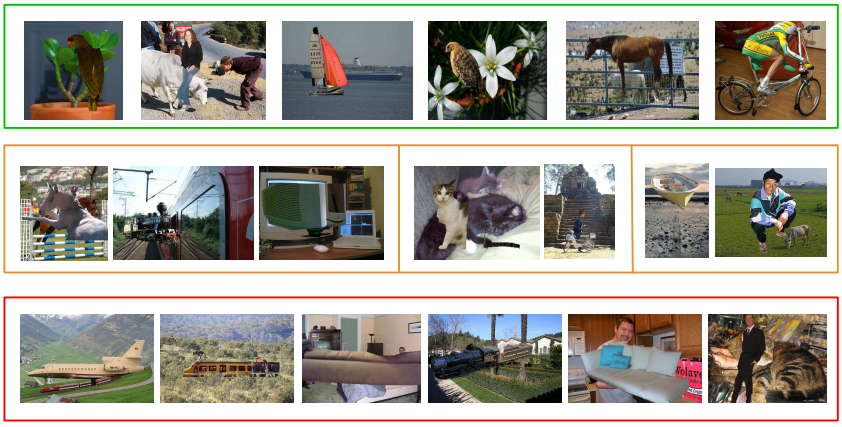

Fig. 5. Examples of instance placement with context model guidance. The figure presents samples obtained by placing a matched examples into the box predicted by the context model. The top row shows generated images that are visually almost indistinguishable from the real ones. The middle row presents samples of good quality although with some visual artifacts. For the two leftmost examples, the context module proposed an appropriate object class, but the pasted instances do not look visually appealing. Sometimes, the scene does not look natural because of the segmentation artifacts as in the two middle images. The two rightmost examples show examples where the category seems to be in the right environment, but not perfectly placed. The bottom row presents some failure cases.

Fig. 6. Illustration of artifacts arising from enlargement augmentation. In the enlargement data augmentation, an instance is cut out of the image, up-scaled by a small factor and placed back at the same location. This approach leads to blending artefacts. Modified images are given in the top row. Zoomed parts of the images centered on blending artifacts are presented in the bottom line.

References

1. Oliva, A., Torralba, A.: The role of context in object recognition. Trends in cognitive sciences (12) (2007) 520–527
2. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009)
3. Murphy, K., Torralba, A., Eaton, D., Freeman, W.: Object detection and localization using local and global features. In: Toward Category-Level Object Recognition. (2006) 382–400
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI) **32**(9) (2010) 1627–1645
5. Park, D., Ramanan, D., Fowlkes, C.: Multiresolution models for object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). (2010)
6. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Proceedings of the European Conference on Computer Vision (ECCV). (2008)
7. 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 (NIPS). (2015)
8. Girshick, R.: Fast R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV). (2015)
9. 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 (ECCV). (2016)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
11. Yu, R., Chen, X., Morariu, V.I., Davis, L.S.: The role of context selection in object detection. In: British Machine Vision Conference (BMVC). (2016)
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1) (2014) 1929–1958
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision (IJCV) **88**(2) (2010) 303–338
14. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017)
15. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
16. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing training data for object detection in indoor scenes. arXiv preprint arXiv:1702.07836 (2017)
17. Torralba, A., Sinha, P.: Statistical context priming for object detection. In: Proceedings of the International Conference on Computer Vision (ICCV). (2001)
18. Torralba, A.: Contextual priming for object detection. International Journal of Computer Vision (IJCV) **53**(2) (2003) 169–191
19. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2010)

20. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: Proceedings of the International Conference on Computer Vision (ICCV). (2009)
21. Chu, W., Cai, D.: Deep feature based contextual model for object detection. Neurocomputing **275** (2018) 1035–1042
22. Bell, S., Zitnick, C.L., 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 (CVPR). (2016)
23. Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C.: Blitznet: A real-time deep network for scene understanding. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017)
24. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
25. Barnea, E., Ben-Shahar, O.: On the utility of context (or the lack thereof) for object detection. arXiv preprint arXiv:1711.05471 (2017)
26. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2010)
27. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Proceedings of the International Conference on Computer Vision (ICCV) (2015)
28. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using gan for improved liver lesion classification. arXiv preprint arXiv:1801.02385 (2018)
29. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models. In: Proceedings of the International Conference on Computer Vision (ICCV). (2015)
30. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. arXiv preprint arXiv:1708.04896 (2017)
31. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. ACM Transactions on Graphics (TOG) **30**(6) (2011) 157
32. Movshovitz-Attias, Y., Kanade, T., Sheikh, Y.: How useful is photo-realistic rendering for visual learning? In: Proceedings of the European Conference on Computer Vision (ECCV). (2016)
33. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: Proceedings of the International Conference on Computer Vision (ICCV). (2015)
34. Prez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics (SIGGRAPH'03) **22**(3) (2003) 313–318
35. 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 (ECCV). (2014)
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
37. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017)
38. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). (2015)

39. Liao, Z., Farhadi, A., Wang, Y., Endres, I., Forsyth, D.: Building a dictionary of image fragments. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012)