On the Importance of Visual Context for Data Augmentation in Scene Understanding

Nikita Dvornik, Julien Mairal, Cordelia Schmid

On the Importance of Visual Context for Data Augmentation in Scene Understanding

Nikita Dvornik, Julien Mairal, *Senior Member, IEEE,* and Cordelia Schmid, *Fellow, IEEE*

Abstract—Performing data augmentation for learning deep neural networks is known to be important for training visual recognition systems. By artificially increasing the number of training examples, it helps reducing overfitting and improves generalization. While simple image transformations can already improve predictive performance in most vision tasks, larger gains can be obtained by leveraging task-specific prior knowledge. In this work, we consider object detection, semantic and instance segmentation and augment the training images by blending objects in existing scenes, using instance segmentation annotations. We observe that randomly pasting objects on images hurts the performance, unless the object is placed in the right context. To resolve this issue, we propose an explicit context model by using a convolutional neural network, which predicts whether an image region is suitable for placing a given object or not. In our experiments, we show that our approach is able to improve object detection, semantic and instance segmentation on the PASCAL VOC12 and COCO datasets, with significant gains in a limited annotation scenario, i.e. when only one category is annotated. We also show that the method is not limited to datasets that come with expensive pixel-wise instance annotations and can be used when only bounding boxes are available, by employing weakly-supervised learning for instance masks approximation.

Index Terms—Convolutional Neural Networks, Data Augmentation, Visual Context, Object Detection, Semantic Segmentation.

1 Introduction

Convolutional neural networks (CNNs) are commonly used for scene understanding tasks such as object detection and semantic segmentation. One of the major challenge to use such models is however to gather and annotate enough training data. Various heuristics are typically used to prevent overfitting such as DropOut [1], penalizing the norm of the network parameters (also called weight decay), or early stopping the optimization algorithm. Even though the exact regularization effect of such approaches on learning is not well understood from a theoretical point of view, these heuristics have been found to be useful in practice.

Apart from the regularization methods related to the optimization procedure, reducing overfitting can be achieved with data augmentation. For most vision problems, generic input image transformations such as cropping, rescaling, adding noise, or adjusting colors are usually helpful and may substantially improve generalization. Developing more elaborate augmentation strategies requires then prior knowledge about the task. For example, all categories in the Pascal VOC [2] or ImageNet [3] datasets are invariant to horizontal flips (e.g. a flipped car is still a car). However, flipping would be harmful for hand-written digits from the MNIST dataset [4] (e.g., a flipped "5" is not a digit).

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 [5], [6], [7]. While not achieving perfect photorealism, this strategy with random placements has proven to be surprisingly effective for *object instance detection* [5], which is a fine-grained detection task consisting of retrieving instances of a particular object from an image collection; in contrast, *object detection* and *semantic segmentation* focus on distinguishing between object categories rather than objects themselves and have to account for rich intra-class variability. For these tasks, the random-placement strategy simply does not work, as shown in the experimental section. Placing training objects at unrealistic positions probably forces the detector to become invariant to contextual information and to focus instead on the object's appearance.

Along the same lines, the authors of [6] 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 [7] 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 more general tasks of scene understanding such as object detection, semantic and instance segmentation, which require more generic context modeling than estimating planes and surfaces as done for instance in [6], [7]. To this end, the first contribution of our paper is methodological: we propose a context model based on a convolutional neural network. The model estimates the likelihood of a particular object category 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 authors are with University Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.*
E-mail: firstname.lastname@inria.fr

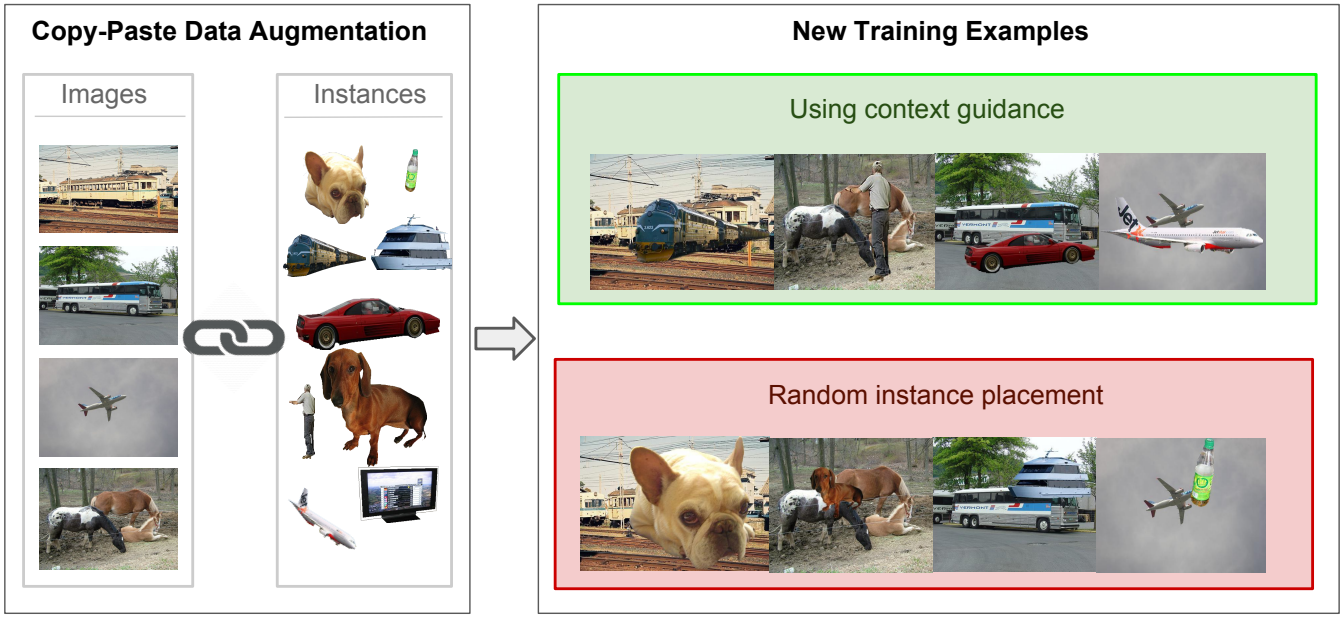

Fig. 1: **Examples of data-augmented training examples produced by our approach.** Images and objects are taken from the VOC'12 dataset that contains segmentation annotations. We compare the output obtained by pasting the objects with our context model vs. those obtained with random placements. Even 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.

The second contribution is experimental: We show with extensive tests on the COCO [8] and VOC'12 benchmarks using different network architectures that context modeling is in fact a key to obtain good results for detection and segmentation tasks and that substantial improvements over non-data-augmented baselines may be achieved when few labeled examples are available. We also show that having expensive pixel-level annotations of objects is not necessary for our method to work well and demonstrate improvement in detection results when using only bounding-box annotations to extract object masks automatically.

The present work is an extension of our preliminary work published at the conference ECCV in 2018 [9]. The main contributions of this long version are listed below:

- We show that our augmentation technique improves detection performance even when training on large-scale data by considering the COCO dataset for object detection in addition to Pascal VOC.
- Whereas the original data augmentation method was designed for object detection, we generalize it to semantic segmentation and instance segmentation.
- We show how to reduce the need for instance segmentation annotations to perform data augmentation for object detection. We employ weakly-supervised learning in order to automatically generate instance masks.
- We demonstrate the benefits of the proposed augmentation strategy for other object detectors than [10], by evaluating our approach with Faster-RCNN [11] and Mask-RCNN [12].

Our context model and the augmentation pipeline are made available as an open-source software package (follow thoth.inrialpes.fr/research/context_aug).

2 Related Work

In this section, we discuss related work for visual context modeling, data augmentation for object detection and semantic segmentation and methods suitable for automatic object segmentation.

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 [13], [14]. Later, the authors of [15] 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 [16], [17]. Then, deep learning approaches such as convolutional neural networks started to be used [11], [18], [19]; as mentioned previously, their features already contain implicitly contextual information. Yet, the work of [20] 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 [21] 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 [10], [21], [22], showing better performance essentially on small objects. The works of [23], [24] 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 [25], [26].

Modeling visual context for semantic segmentation.

While object detection operates on image's rectangular regions, in semantic segmentation the neighboring pixels with similar values are usually organized together in so-called superpixels [27]. This allows defining contextual relations between such regions. The work of [28] introduces "context clusters" that are discovered and learned from region features. They are later used to define a specific class model for each context cluster. In the work of [29] the authors tile an image with superpixels at different scales and use this representation to build global and local context descriptors. The work of [30] computes texton features [31] for each pixel of an image and defines shape filers on them. This enables the authors to compute local and middle-range concurrence statistics and enrich region features with context information. Modern CNN-based methods on the contrary rarely define an explicit context model and mostly rely on large receptive fields [32]. Moreover, by engineering the network's architecture one can explicitly require local pixel descriptors used for classification to carry global image information too, which enables reasoning with context. To achieve this goal encoder-decoder architectures [10], [33] use deconvolutional operations to propagate coarse semantic image-level information to the final layers while refining details with local information from earlier layers using skip-connections. As an alternative, one can use dilated convolutions [34], [35] that do not down-sample the representation but rather upsample the filters by introducing "wholes" in them. Doing so is computationally efficient and allows to account for global image statistics in pixel classification. Even though visual context is implicitly present in the networks outputs, it is possible to define an explicit context model [35], [36] on top of them. This usually results in moderate improvement in model's accuracy.

Data augmentation for object detection and segmentation. Data augmentation is a major tool to train deep neural networks. It varies from trivial geometrical transformations such as horizontal flipping, cropping with color perturbations, and adding noise to an image [37], to synthesizing new training images [38], [39]. Some recent object detectors [10], [19], [40] benefit from standard data augmentation techniques more than others [11], [18]. The performance of Fast- and Faster-RCNN could be for instance boosted by simply corrupting random parts of an image in order to mimic occlusions [41]. The field of semantic segmentation is enjoying a different trend—augmenting a dataset with synthetic images. They could be generated using extra annotations [42], come from a purely synthetic dataset with dense annotations [43], [44] or a simulated environment [45]. For object detection, recent works such as [46], [47], [48] also 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 [39]. This problem could be alleviated by using transfer-learning techniques such as [49] or by improving photo-realism of synthetic data [50], [51]. To address the same issue, the authors of [6] 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 [7] estimates scene geometry and spatial layout, before synthetically placing objects in the image to create realistic training examples. In [5], 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. In contrast to this method, our approach does not choose pasted locations at random but uses an explicit context model. We found this crucial to improve general object detection.

Automatic Instance Segmentation The task of instance segmentation is challenging and requires considerable amount of annotated data [8] in order to achieve good results. Segmentation annotations are the most labor-demanding since they require pixel-level precision. The need to distinguish between instances of one class makes annotating "crowd scenes" extremely time-consuming. If data for this problem comes without labels, tedious and expensive process of annotation may suggests considering other solutions that do not require full supervision. The work of [52] uses various image statistics and hand-crafted descriptors that do not require learning along with annotated image tags, in order to build a segmentation proposal system. With very little supervision, they learn to descriminate between "good" and "bad" instance masks and as a result are able to automatically discover good quality instance segments within the dataset. As an alternative, one can use weakly-supervised methods to estimate instance masks. The authors of [53] use only category image-level annotations in order to train an object segmentation system. This is done by exploiting class-peak responses obtained using pre-trained classification network and propagating them spatially to cover meaningful image segments. It is beneficial to use instance-level annotations, such as object boxes and corresponding categories, if those are available, in order to improve the system's performance. The work of [54] proposes a rather simple yet efficient framework for doing so. By providing the network with extra information, which is a rectangular region containing an object, a system learns to discover instance masks automatically inside those regions. Alternatively, the system could be trained to provide semantic segmentation masks in a weakly-supervised fashion. Together with bounding boxes, one may use it to approximate instance masks.

3 Approach

In this section, we present a simple experiment to motivate our context-driven data augmentation, and present the full pipeline in details. We start by describing a naive solution to augmenting an object detection dataset, which is to perform copy-paste data augmentation agnostic to context by placing objects at random locations. Next, we explain why it fails for our task and propose a natural solution based on explicit context modeling by a CNN. We show how to apply the context model to perform augmentation for detection and segmentation tasks and how to blend the object into existing scenes. The full pipeline is depicted in Figure. 2.

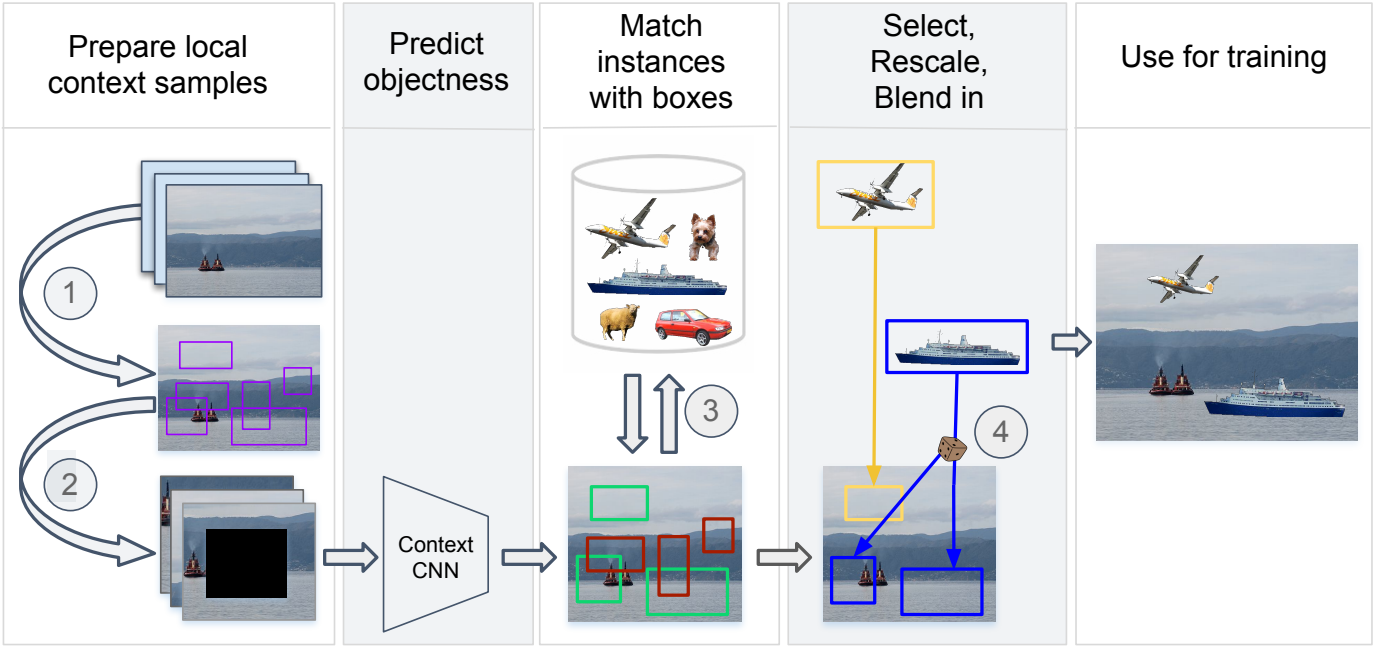

Fig. 2: **Illustration of our data augmentation approach.** We select an image for augmentation and 1) generate 200 candidate boxes that cover the image. Then, 2) for each box we find a neighborhood that contains the box entirely, crop this neighborhood and mask all pixels falling inside the bounding 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 Copy-paste Data Augmentation with Random Placement is not Effective for Object Detection

In [5], data augmentation is performed by positioning segmented objects at random locations 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 blending strategies to smooth object boundaries such as Poisson blending [55], 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, for each of the categories, preventing the network to overfit them. According to [5], this strategy can bring 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 augmentation strategy described above does not improve the results on the general object detection task and may even hurt the performance as we show in the experimental section. To justify the initial claim, we follow [5] as close as possible and conduct the following experiment on the PASCAL VOC12 dataset [2]. Using provided instance segmentation masks we extract objects from images and store them in a so-called instance-database. They are used to augment existing images in the training dataset by placing the instances at random locations. In order to reduce blending artifacts we use one of the following strategies: smoothing the edges using Gaussian or linear blur, applying Poisson blending [55] in the segmented region, blurring the whole image by simulating a slight camera motion or leaving the pasted object untouched. As distractors, we used objects from the COCO dataset [8] belonging to categories not present in PASCAL VOC [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 Explicit Context Modeling by CNN

The core idea behind the proposed method is that it is possible to some extent to guess the category of an object just by looking at its visual surroundings. That is precisely what we are modeling by a convolutional neural network, which takes contextual neighborhood of an object as input and is trained to predict the object's class. Here, we describe the training data and the learning procedure in more details.

Contextual data generation. In order to train the contextual model we use a dataset that comes with bounding box and

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

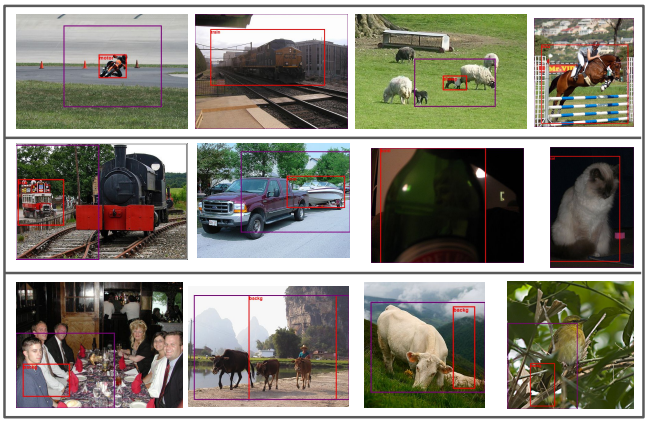

Fig. 3: **Contextual images - examples of inputs to the context model**. A subimage bounded by a magenta box is used as an input to the context model after masking-out the object information inside a red box. The top row lists examples of positive samples encoding real objects surrounded by regular and 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.

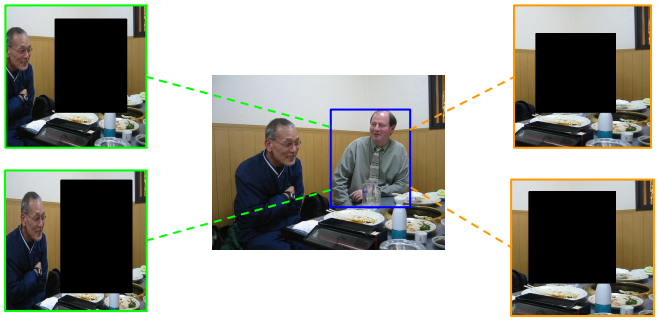

Fig. 4: **Different contextual images obtained from a single bounding box.** A single ground-truth bounding box (in blue) is able to generate a set of different context images (in green and orange) by varying the size of the initial box and the context neighborhood. While the orange contextual images may be recognized as a chair, the green ones make it more clear that the person was masked out. This motivates the need to evaluate several context images for one box during the context estimation phase.

object class annotations. Each ground-truth bounding box in the dataset is able to generate positive "contextual images" that are used as input to the system. As depicted in the Figure 3, a "contextual image" is a sub-image of an original training image, fully enclosing the selected bounding box, whose content is masked out. Such a contextual image only carries information about visual neighborhood that defines middle-range context and no explicit information about the deleted object. One box is able to generate multiple different context images, as illustrated in Figure 4. Background "contextual images" are generated from bounding boxes that do not contain an object and are formally defined in [9]. To prevent distinguishing between positive and background images only by looking at the box shape and to force true visual context modeling, we estimate the shape distribution of positive boxes and sample the background ones from it. Precisely, we estimate the joint distribution of scale s and aspect ratio a with a two-dimensional histogram, as described in [9], and we draw a pair (s, a) from this distribution in order to construct a background box. Since in natural images there is more background boxes than the ones actually containing an object, we address the imbalance by sampling more background boxes, following sampling strategies in [9], [11].

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. These contextual images are resized to 300×300 pixels, and the output of the network is a label in $\{0, 1, ..., C\}$, where C is the number of object categories. The 0-th class represents background and corresponds to a negative "context image". For such a multi-class image classification problem, we use the classical ResNet50 network [56] pre-trained on ImageNet, and change the last layer to be a softmax with $C + 1$ activations (see experimental section for details).

3.3 Context-driven Data Augmentation

Once the context model is trained, we use it to provide locations where to paste objects. In this section, we elaborate on the context network inference and describe the precise procedure used for blending new objects into existing scenes.

Selection of candidate locations for object placement. A location for pasting an object is represented as a bounding box. For a single image, we sample 200 boxes at random from the shape distribution used in 3.2 and later select the successful placement candidates among them. These boxes are used to build corresponding contextual images, that we feed to the context model as input. As output, the model provides a set of scores in range between 0 and 1, representing the presence likelihood of each object category in a given bounding box, by considering its visual surrounding. The top scoring boxes are added to the final candidate set. Since the model takes into account not only the visual surroundings but a box's geometry too, we need to consider all possible boxes inside an image to maximize the recall. However this is too costly and using 200 candidates was found to provide good enough bounding boxes among the top scoring ones.
After analyzing the context model's output we made the following observation: if an object of category c is present in an image it is a confident signal for the model to place another object of this class nearby. The model ignores this signal only if no box of appropriate shape was sampled in the object's neighborhood. This often happens when only 200 candidate locations are sampled; however, evaluating more locations would introduce a computational overhead. To fix this issue, we propose a simple heuristic, which consists of drawing boxes in the neighborhood of this object

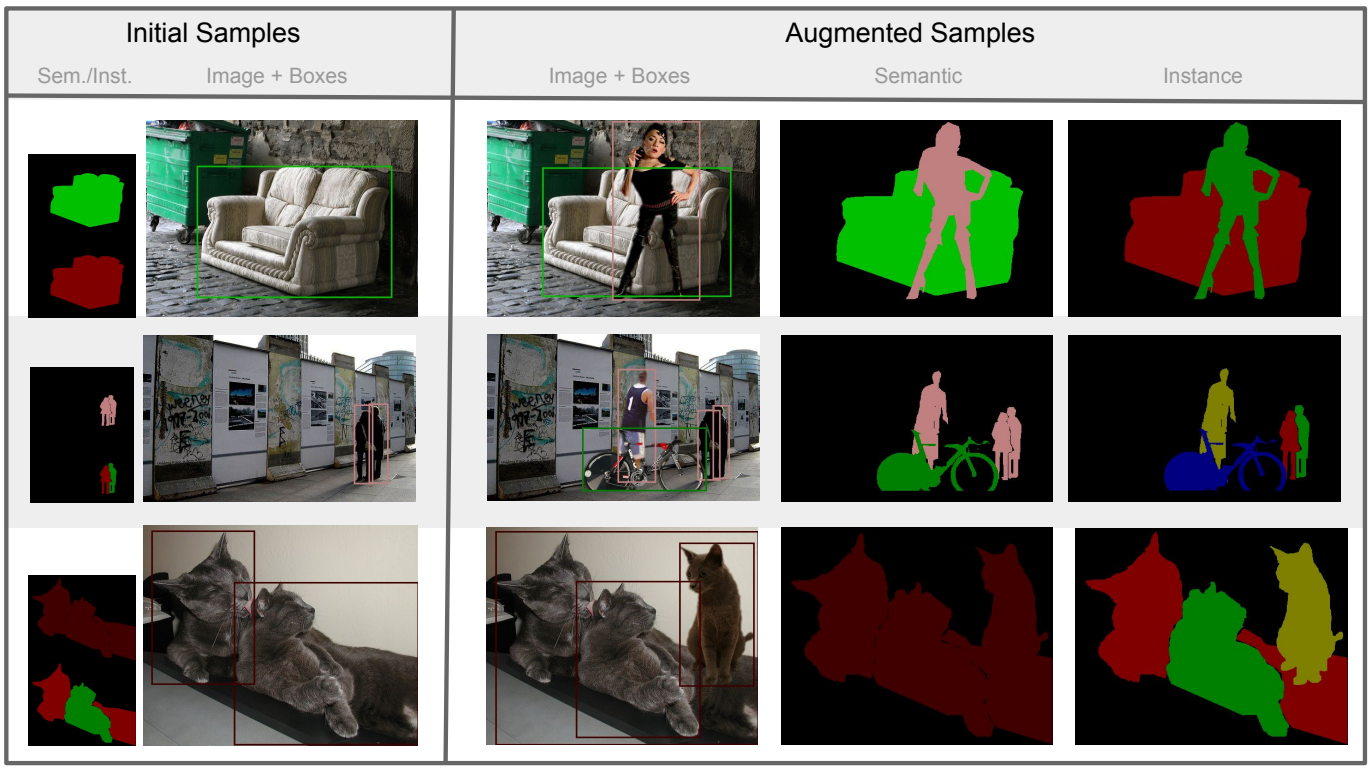

Fig. 5: **Data augmentation for different types of annotations.** The first column contains samples from the training dataset with corresponding semantic/instance segmentation and bounding box annotations. Columns 2-4 present the result of applying context-driven augmentation to the initial sample with corresponding annotations.

and adding them to the final candidate set. The added boxes have the same geometry (up to slight distortions) as the neighboring object's box.

Candidate scoring process. As noted before, we use the context model to score the boxes by using its softmax output. Since the process of generating a contextual image is not deterministic, predictions on two contextual images corresponding to the same box may differ substantially, as illustrated in Figure 4. We alleviate this effect by sampling 3 contextual images for one location and average the predicted scores. After the estimation stage we retain the boxes where an object category has score greater than 0.7; These boxes together with the candidates added at the previous step form the final candidate set that will be used for object placement.

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 [5]. 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. Figure 6 presents the blending techniques mentioned above. We also do not consider Poisson blending in our approach, which was considerably slowing down the data generation procedure. Unlike [5] 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 [5]. As a consequence, we do not need to exploit external data to perform data augmentation.

Updating image annotation. Once an image is augmented by blending in a new object, we need to modify the annotation accordingly. In this work, we consider data augmention for both object detection and semantic segmentation, as illustrated in Figure 5. Once a new object is placed in the scene, we generate a bounding box for object detection by drawing the tightest box around that object. In case where an initial object is too occluded by the blended one, i.e. the IoU between their boxes is higher than 0.8, we delete the bounding box of the original object from the annotations. For semantic segmentation, we start by considering augmentation on instance masks (Figure 5, column 4) and then convert them to semantic masks (Figure 5, column 3). If a new instance occludes more than 80% of an object already present in the scene, we discard annotations for all pixels belonging to the latter instance. To obtain semantic segmentation masks from instance segmentations, each instance pixel is labeled with the corresponding objects class.

4 Experiments

In this section, we use the proposed context model to augment object detection and segmentation datasets. We start by presenting experimental and implementation details in Sections 4.1 and 4.2 respectively. In Section 4.3 we present a preliminary experiment that motivates the proposed solution. In Sections 4.4.1 and 4.4.2 we study the

Fig. 6: **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 [55].

effect of context-driven data augmentation when augmenting an object detection dataset. For this purpose we consider the Pascal VOC12 dataset that has instance segmentation annotations and we demonstrate the applicability of our method to different families of object detectors. We study the scalability of our approach in Section 4.5 by using the COCO dataset for object detection and instance segmentation. We show benefits of our method in Section 4.6 by augmenting the VOC12 for semantic segmentation. In Section 4.7, we use weakly-supervised learning for estimating object masks and evaluate our approach on the Pascal VOC12 dataset using only bounding box annotations. Finally, Section 4.8 studies how the amount of data available for training the context model influences the final detection performance.

4.1 Dataset, Tools, and Metrics

Datasets. In our experiments, we use the Pascal VOC'12 [2] and COCO [8] datasets. In the VOC'12 dataset, we only consider a subset that contains segmentation annotations. The training set contains $1\,464$ images and is dubbed `VOC12train-seg` later in the paper. Following standard practice, we use the test set of VOC'07 to evaluate the detection performance, which contains $4\,952$ images with the same 20 object categories as VOC'12. We call this image set `VOC07-test`. When evaluating segmentation performance, we use the validation set of the VOC'12 annotated with segmentation masks `VOC12val-seg` that contains $1\,449$ images.
The COCO dataset [8] is used for large-scale object detection experiments. It includes 80 object categories for detection and instance segmentation. For both tasks, there are 118K images for training that we denote as `COCO-train2017` and 5K for validation and testing denoted as `COCO-val2017`.

Models. To test our data-augmentation strategy we chose a single model capable of performing both object detection and semantic segmentation. BlitzNet [10] is an encoder-decoder architecture, which is able to solve either of the tasks, or both simultaneously if trained with box and segmentation annotations together. The open-source implementation is available online. If used to solve the detection task, BlitzNet achieves close to the state-of-the-art results (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 [10]); this network is similar to the DSSD detector of [22] that was also used in the Focal Loss paper [57]. When used as a segmentor, BlitzNet resembles the classical U-Net architecture [58] and also achieves results comparable to the state-of-the-art on VOC'12-test set (75.5% mIoU). The advantage of such class of models is that it is relatively fast (it may work in real time) and supports training with big batches of images without further modification. To make the evaluation extensive, we also consider a different region-based class of detectors. For that purpose we employ an open-source implementation of Faster-RCNN [59] which uses ResNet50 [56] architecture as a feature extractor. Finally, when tackling object detection and instance segmentation on COCO, we use Mask-RCNN [12] that solves both tasks simultaneously. For each region proposal the network outputs estimated class probabilities, regressed box offsets and a predicted instance mask. We run the official implementation of [60] that uses ResNet50 as a backbone, followed by an FPN [61] module. This setup corresponds to the current state of the art in object detection and instance segmentation.

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 object detection and instance segmentation for one object class is the average precision (AP). Mean Average Precision (mAP) is used to report the overall performance on the dataset. Mean Intersection Over Union (mIoU) is used to measure performance on semantic segmentation.

4.2 Implementation Details

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 [56] with ImageNet initialization to train a contextual model in all our experiments. Since we have access only to the training set at any stage of the pipeline we define two strategies for training the context model. When the amount of positive samples is scarce, 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 validation set. The moment when the loss curves start diverging noticeably is used as a stopping point. We call this training setting "small-data regime". When the size of the training set is moderate and we are in "normal-data regime", we split it in two parts ensuring that for each class, there is a similar number of positive examples in both splits. The context model is trained on one split and applied to another one. We train the model with ADAM optimizer [62] starting with learning rate 10^{-4} and decreasing it by the factor of 10 once during the learning phase. The number of steps depends on a dataset. We sample 3 times more background contextual images, as noted in Section 3.2. Visual examples of augmented images produced when using the context model are presented in Figure 7. Overall, training the context model is about 4-5 times faster than training the detector.

Training detection and segmentation models. In this work, the BlitzNet model takes images of size 300×300 as an input and produces a task-specific output. When used as a detector, the output is a set of candidate object boxes

with classification scores and in case of segmentation it is an estimated semantic map of size 75×75; like our context model, it uses ResNet50 [56] pre-trained on ImageNet as a backbone. The models are trained by following [10], with the ADAM optimizer [62] starting from learning rate 10^{-4} and decreasing it later during training by a factor 10 (see Sections 4.4 and 4.6 for 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 [10]. For the Faster-RCNN detector training, we consider the classical model of [11] with ResNet50 backbone and closely follow the instructions of [59]. On the Pascal VOC12 dataset, training images are rescaled to have both sides between 600 and 1000 pixels before being passed to the network. The model is trained with the Momentum optimizer for 9 epochs in total. The starting learning rate is set to 10^{-2} and divided by 10 after first 8 epochs of training. When using Mask-RCNN [12], the images are rescaled to have a maximum size of 1333 pixel on one side or a minimum one of 800 pixels. Following the original implementation [60], training and evaluation is performed on 2 GPUs, where images are grouped in batches of size 16. We set the starting learning rate to $2 \cdot 10^{-2}$ which is decreased by a factor of 10 twice later during training. For both Faster-RCNN and Mask-RCNN standard data augmentation includes only horizontal flipping.

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` or `COCO-train` sets 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. The scaling factor is kept close to 1 not to introduce scaling artefacts. When blending the objects into the new background, we follow [5] 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. By introducing new instances in a scene we may also introduce heavy occlusions of existing objects. The strategy for resolving this issue depends on the task and is clarified in Sections 4.4 and 4.6.

4.3 Why is Random Placement not Working?

As we discovered in the Section 3.1, random copy-paste data augmentation does not bring improvement when used to augment object detection datasets. There are multiple possible reasons for observing this behavior, such as violation of context constraints imposed by the dataset, objects looking "out of the scene" due to different illumination conditions or simply artifacts introduced due to blending techniques. To investigate this phenomenon, we conduct a study, that aims to better understand (i) the importance of visual context for object detection, (ii) the role of illumination conditions and (iii) the impact of blending artefacts. For simplicity, we choose the first 5 categories of VOC'12, namely *aeroplane, bike, bird, boat, bottle*, and train independent detectors per category.

Method	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
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

TABLE 1: Ablation study on the first five categories of VOC'12. All models are learned independently. We compare classical data augmentation techniques (Base-DA), approaches obtained by copy-pasting objects, either randomly (Random-DA) or by preserving context (Enlarge+Reblend-DA). The line "Removing context" corresponds to the first experiment described in Section 4.3; Enlarge-Reblend corresponds to the second experiment.

Baseline when no object is in context. To confirm the negative influence of random placing, we consider one-category detection, where only objects of one selected class are annotated with bounding boxes and everything else is considered as background. Images that do not contain objects of the selected category become background images. After training 5 independent detectors as a baseline, we construct a similar experiment by learning on the same number of instances, but considering 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; (iii) all objects may suffer from rendering artefacts. As shown in Table 1, 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 in the third row of Table 1. Here, we propose a simple experiment showing that neither (iii) blending artefacts nor (ii) illumination difference are 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. To mimic the illumination change we apply a slight color transformation to the segmented object. 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 forth row of Table 1, this approach improves over the baseline, which suggests that the lack of visual context is probably the key explaining the result observed before. The experiment also confirms that the presence of difference in illumination and blending artefacts is not critical for the object detection task.

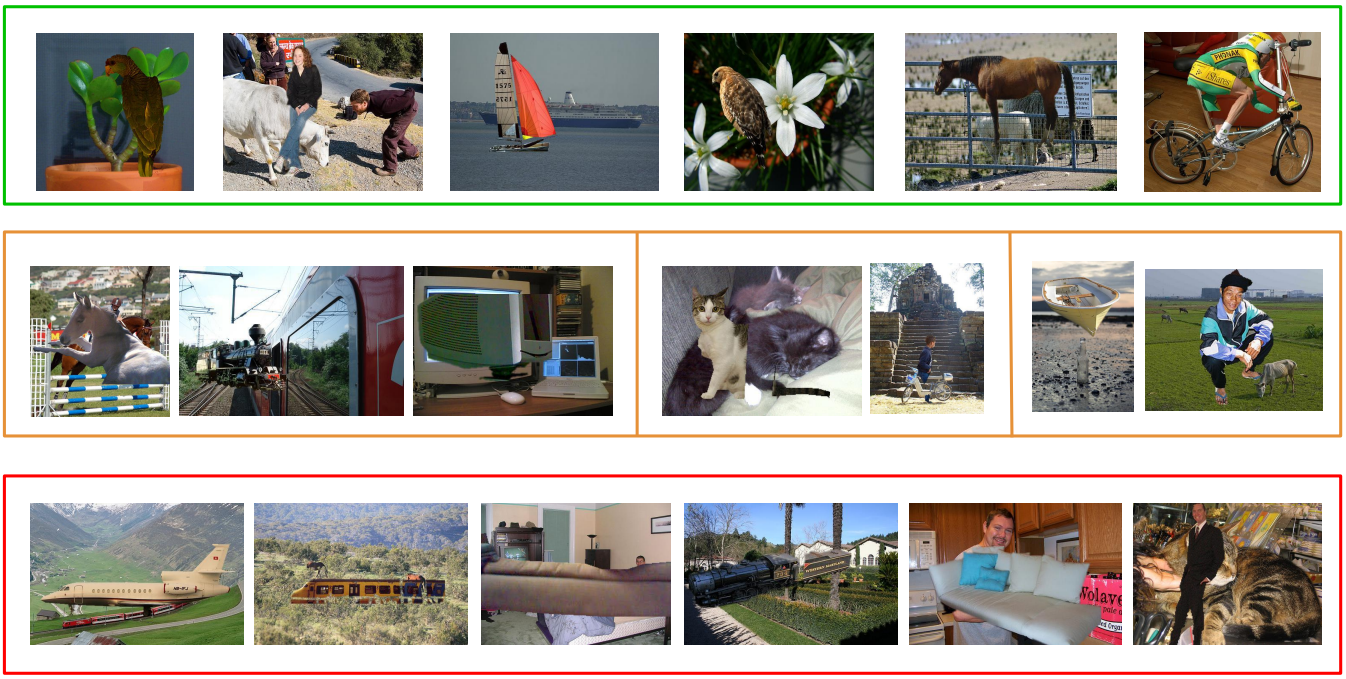

Fig. 7: **Examples of instance placement with context model guidance.** The figure presents samples obtained by placing a matched examples into the box predicted by the context model. The top row shows generated images that are visually almost indistinguishable from the real ones. The middle row presents 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.

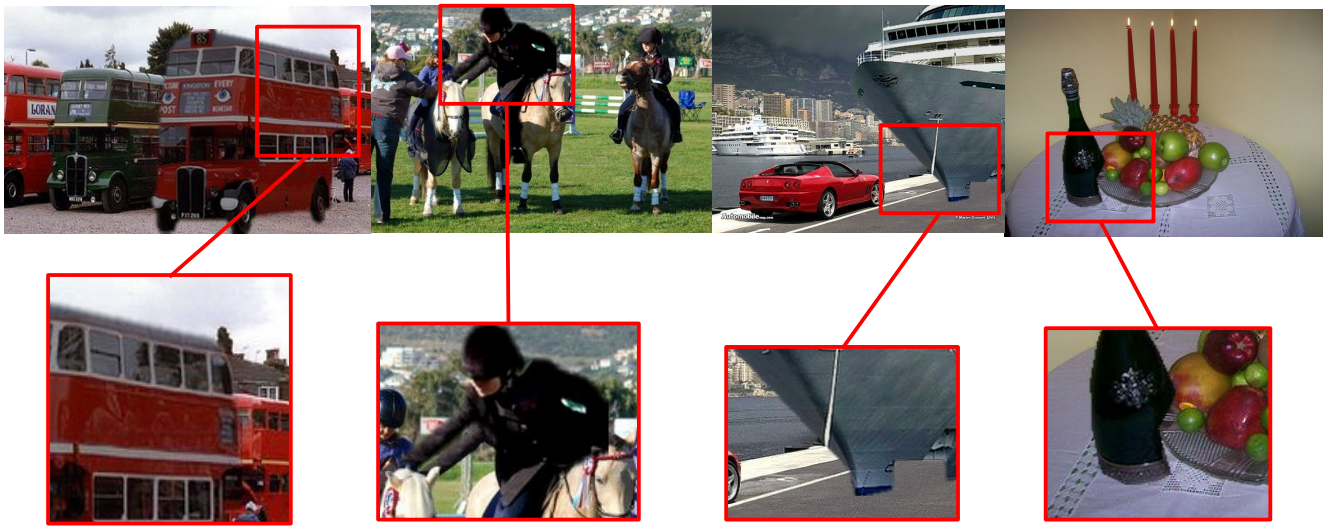

Fig. 8: **Illustration of artifacts arising from enlargement augmentation.** In the enlargement data augmentation, an instance is cut out of the image, up-scaled by a small factor and placed back at the same location. This approach leads to blending artefacts. Modified images are given in the top row. Zoomed parts of the images centered on blending artifacts are presented in the bottom line.

Visual examples of such artefacts are presented in Figure 8.

4.4 Object Detection Augmentation on VOC PASCAL

In this subsection, we are conducting experiments on object detection by augmenting the PASCAL VOC'12 dataset. In order to measure the impact of the proposed technique in a "small data regime", we pick the single-category detection scenario and also consider a more standard multi-category setting. We test single-shot region-based families of detectors—with BlitzNet and Faster-RCNN respectively—and observe improved performance in both cases.

4.4.1 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 $[0.5, 1.5]$ 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 context model was trained in "small-data regime" for 2K iterations and the learning rate was dropped once after 1.5K steps. The results for this experiment are presented in Table 2.

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 improves upon the random placement one, on average by 7%, and upon the baseline that uses only classical data augmentation, on average by 6%. Interestingly, we identify categories for which visual context is crucial (aeroplane, bird, boat, bus, cat, cow, dog, plant), for which context-driven data augmentation brings more than 7% improvement and some categories that display no significant gain or losses (chair, table, persons, tv), where the difference with the baseline is less noticeable (around 1-3%).

4.4.2 Multiple-Categories Object Detection

In this section, we conduct the same experiment as in Section 4.4.1, 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 2). When training the context model, we follow the "normal-data strategy" described in Section 4.2 and train the model for 8K iterations, decreasing the learning rate after 6K steps. The results are presented in Table 3 and show a modest average improvement of 2.1% for a single shot and 1.4% for a region-based detector on average over the corresponding baselines, which is relatively consistent across categories. This confirms that data augmentation is crucial when few labeled examples are available.

4.5 Object Detection and Instance Segmentation Augmentation on COCO

In order to test our augmentation technique at large scale, we use in this section the COCO dataset [8] whose training set size is by 2 orders of magnitude larger than `voc12train-seg`, and consider both object detection and instance segmentation tasks.

4.5.1 Object Detection with BlitzNet

By design, the experiment is identical to the one presented in Section 4.4.2. However, for the COCO dataset we need to train a new context model. This is done by training for 350K iterations (decay at 250K) as described in Section 4.2. The non data-augmented baseline was trained according to [10]; when using our augmentation pipeline, we train the detector for 700K iterations and decrease the learning rate by a factor of 10 after 500K and 600K iterations. Table 5 shows that we are able to achieve a modest improvement of 0.7%, and that data augmentation still works and does not degrade the performance regardless the large amount of data available for training initially.

4.5.2 Detection and Segmentation with Mask-RCNN

For this experiment, we use Mask-RCNN [12] that jointly solves object detection and instance segmentation. When training the baseline model, we closely follow original guidelines[2] and train the model with 2x schedule (for 180K iterations) to maximize the baseline's performance. Training the model with 1x schedule (for 90K iterations) results in underfitting, while training with x4 schedule (for 360K iterations), results in overfitting. In order to improve the performance of Mask-RCNN for both tasks, we train the model with x4 schedule and use the context-driven data augmentation. In order to reduce pasting artifacts negatively affecting Mask-RCNN, we decrease the augmentation probability during the training. More precisely, augmentation probability is set to 0.5 in the beginning of the training and then linearly decreased to 0 by the end of the training procedure. Training with constant augmentation probability did not improve the performance over the x2 baseline. On the other hand, gradually reducing augmentation probability results in less aggressive regularization and brings more

2. https://github.com/facebookresearch/Detectron/blob/master/MODEL_ZOO.md

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
Enlarge-DA	60.1	63.4	51.6	48.0	34.8	68.8	72.1	70.4	41.1	63.7	62.3	56.3	70.1	67.8	65.3	37.9	58.1	61.2	75.5	65.9	59.7
Context-DA	68.9	73.1	62.5	57.6	38.9	72.5	74.8	77.2	42.9	69.7	59.5	63.9	76.1	70.2	69.2	43.9	58.3	59.7	77.2	64.8	64.0
Impr. Cont.	**10.1**	**8.7**	**13.7**	**9.2**	5.0	6.0	5.1	**9.2**	2.5	**10.7**	1.5	**7.5**	4.0	6.0	2.5	**7.3**	3.8	6.7	4.2	1.2	5.8

TABLE 2: 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. "Enlarge-DA" augmentation blends up-scaled instances back in their initial location, which is given in row 3. The forth 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 7%) are in bold. All numbers are averaged over 3 independent experiments.

model	CDA	aero	bike	bird	boat	bott.	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	avg.
BlitzNet300		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
	✓	69.9	73.8	63.9	62.6	35.3	78.3	73.5	80.6	42.8	73.8	62.7	74.5	81.1	73.2	68.9	38.1	67.8	64.3	79.3	66.1	**66.5**
F-RCNN		65.8	70.9	66.5	54.6	45.9	72.7	72.9	80.3	36.8	70.3	48.0	78.9	70.7	70.6	66.3	33.1	64.7	59.8	71.8	61.1	63.1
	✓	67.4	67.7	64.9	58.0	50.4	71.6	74.9	80.4	36.8	70.2	56.4	75.7	73.7	71.6	71.5	39.4	68.6	63.5	67.7	60.1	**64.5**

TABLE 3: 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 column specifies the detector used in the experiment, the second column notes if Context-driven Data Augmentation (CDA) was used. 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	79.0	43.7	65.8	57.9	53.8	83.8	77.9	76.7	19.2	56.6	46.6	67.6	59.0	73.1	77.9	46.8	69.4	37.8	73.7	70.3	63.3
Random-DA	78.1	47.1	75.4	57.8	57.2	83.5	76.2	76.6	20.5	57.0	43.1	69.2	57.5	71.5	78.2	40.0	63.3	42.0	74.5	64.1	63.1
Enlarge-DA	77.2	45.4	67.9	57.9	61.0	84.1	78.8	76.3	20.3	58.4	46.9	67.5	60.5	73.9	78.1	45.2	71.1	38.8	73.6	71.1	64.1
Context-DA	81.7	46.4	73.4	60.7	59.4	85.3	78.8	79.1	20.6	60.0	48.0	68.1	62.2	75.3	78.8	47.6	71.6	39.9	73.6	70.3	65.4
Impr. Cont.	**2.7**	**2.7**	**7.6**	**2.8**	**4.6**	1.5	1.1	2.3	1.4	**3.4**	1.4	0.5	**3.2**	2.3	2.2	0.9	0.8	2.1	-0.1	0	2.1

TABLE 4: Comparison of segmentation accuracy on `VOC12val-seg`. The model is trained on all 20 categories by using the $1\,464$ images from `VOC12train-seg`. Base-DA represents the baseline experiment that uses standard data augmentation techniques. Context-DA uses also our context-driven data augmentation. Random-DA is its context-agnostic analogue. Enlarge-DA corresponds to randomly enlarging an instance and blending it back. The last row presents absolute improvement over the baseline. The numbers represent IoU per class in %. Categories enjoying an improvement higher than 2.5% are in bold. All numbers are averaged over 3 independent experiments.

benefits when training on a large dataset, such as COCO. As Table 5 shows, following this augmentation strategy results in a 0.5% an 0.3% mAP improvement for detection and segmentation respectively, when comparing to the most accurate baseline Mask-RCNN, trained with x2 schedule. Augmenting the training data with random placement strategy hurts the performance substantially, which highlights the importance of context for data augmentation.

4.6 Semantic Segmentation Augmentation

In this section, we demonstrate the benefits of the proposed data augmentation technique for the task of semantic segmentation by using the VOC'12 dataset. First, we set up the baseline by training the BlitzNet300 [10] architecture for semantic segmentation. Standard augmentation techniques such as flipping, cropping, color transformations and adding random noise were applied during the training, as described in the original paper. We use `voc12train-seg` subset for learning the model parameters. Following the training procedure described in Section 4.2, we train the model for 12K iterations starting from the learning rate of 10^{-4} and decreasing it twice by the factor of 10, after 7K and 10K steps respectively. Next, we perform data augmentation of the training set with the proposed context-driven strategy and train the same model for 15K iterations, dropping the learning rate at 8K and 12K steps. In order to blend new objects in and to augment the ground truth we follow routines described in Section 3.3. We also carry out an experiment where new instances are placed at random locations, which represents a context-agnostic counterpart of our method. We summarize the results of all 3 experiments in Table 4. As we can see from the table, performing copy-paste augmentation at random locations for semantic segmentation slightly degrades the model's performance by 0.2%. However when objects are placed in the right context, we achieve a boost of 2.1% in mean intersection over union. These results resem-

Model	DA	@0.5:0.95	@0.5	@0.75	S	M	L
Object Detection							
BlitzNet300		27.3	46.0	28.1	10.7	26.8	46.0
BlitzNet300	Rnd	26.8	45.0	27.6	9.3	26.0	45.7
BlitzNet300	Cont	**28.0**	**46.7**	**28.9**	10.7	**27.8**	**47.0**
Mask-RCNN		38.6	59.7	42.0	22.1	41.5	50.6
Mask-RCNN	Rnd	36.9	57.3	39.7	20.5	39.6	48.0
Mask-RCNN	Cont	**39.1**	**60.3**	**42.3**	**22.4**	**42.2**	**51.2**
Instance Segmentation							
Mask-RCNN		34.5	56.5	36.3	15.7	37.1	52.1
Mask-RCNN	Rnd	33.6	55.2	35.8	14.8	35.5	50.0
Mask-RCNN	Cont	**34.8**	**57.0**	**36.5**	**15.9**	**37.6**	**52.5**

TABLE 5: Comparison of object detection and instance segmentation accuracy on `COCO-val2017` for the multiple-category experiment. The model is trained on all categories at the same time, by using the 118 783 images from `COCO-train2017`. The first column specifies a model used to solve a task, the second column notes if Context-driven (Cont) or random-placement (Rnd) Data Augmentation was used. For different IoU thresholds @0.5:0.95, @0.5 and @0.75) and for different object size (S, M, L), the numbers represent mAP in %. Best results are in bold.

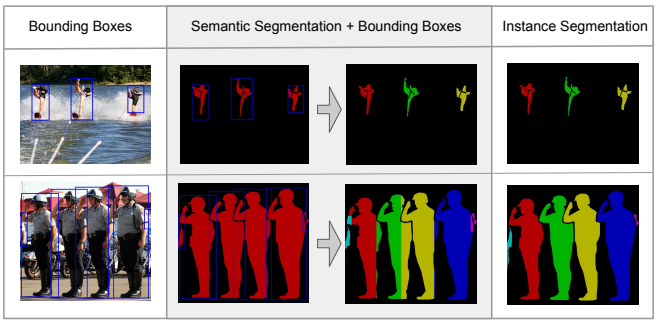

Fig. 9: **Possible types of instance-level annotation.** The left column presents an image annotated with object boxes. Column 2 shows semantic segmentation annotations with object boxes on top and approximate instance segmentations derived from it. The last column presents the original instance segmentation annotations.

ble the case of object detection a lot and therefore highlight the role of context in scene understanding. We further analyze the categories that benefit from our data augmentation technique more than the others. If improvement for a class AP over the baseline is higher than 2.5%, Table 4 marks the result in bold. Again, we can notice correlation with the detection results from Section 4.4.1 which demonstrates the importance of context for the categories that benefit from our augmentation strategy in both cases.

4.7 Reducing the need for pixel-wise object annotation

Our data augmentation technique requires instance-level segmentations, which are not always available in realistic scenarios. In this section, we relax the annotation requirements for our approach and show that it is possible to use the method when only bounding boxes are available.

Semantic segmentation + bounding box annotations. Instance segmentation masks provide annotations to each pixel in an image and specify (i) an instance a pixel belongs to and (ii) class of that instance. If these annotations are not available, one may approximate them with semantic segmentation and bounding boxes annotations. Figure 9 illustrates possible annotation types and the difference between them. Semantic segmentation annotations are also pixel-wise, however they annotate each pixel only with the object category. Instance-specific information could be obtained from object bounding boxes, however this type of annotation is not pixel-wise and in some cases is not sufficient to assign each pixel to the correct instance. As Figure 9 suggests, as long as a pixel in semantic map is covered by only one bounding box, it uniquely defines the object it belongs to (row 1); otherwise, if more than one box covers the pixel, it is not clear which object it comes from (row 2). When deriving approximate instance masks from semantic segmentation and bounding boxes (see Figure 9, column 2), we randomly order the boxes and assign pixels from a semantic map to the corresponding instances. Whenever a pixel could be assigned to multiple boxes we choose a box that comes first in the ordering. Once the procedure for obtaining object masks is established we are back to the initial setting and follow the proposed data augmentation routines described above. As could be seen in Tables 6 and 7 detection performance expiriences a slight drop of 0.6% in single-category and 0.3% in multi-category settings respectively, comparing to using instance segmentation masks. These results are promising and encourage us to explore less elaborate annotations for data augmentation.

Bounding box annotations only. Since we have an established procedure for performing data augmentation with semantic segmentation and bounding boxes annotations, the next step to reducing pixel-wise annotation is to approximate segmentation masks. We employ weakly-supervised learning to estimate segmentations from bounding boxes and use the work of [54]. When trained on the `VOC12train` dataset, augmented with more training examples according to [17], [54], it achieves 65.7% mIoU on the `VOC12val-set`. Unfortunately, we have found that naively applying this solution for estimating segmentation masks and using them for augmentation results in worse performance. The reason for that was low quality of estimated masks. First, inaccurate object boundaries result in non-realistic instances and may introduce biases in the augmented dataset. But more importantly, confusion between classes may hampers the performance. For example, augmenting a category "cow" with examples of a "sheep" class may hurt the learning process. Hence, we need a model with a more discriminative classifier. To this end we propose the following modifications to the segmentation method: we change the architecture from DeepLab_v1 [63] to DeepLab_v4 [35], perform multi-scale inference and process the resulting masks with a conditional random field. The later helps to refine the object edges, which was found not necessary in the original work of [35], when learning with full supervision. By training on the same data as the original method of [54] but with the proposed modifications we achieve 75.8% mIoU, which is more than 10% improvement to the initial pipeline. This accuracy seems to be sufficient to use automatically-estimated

Aug. type	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	avg.
Inst seg	**68.9**	**73.1**	**62.5**	**57.6**	**38.9**	**72.5**	**74.8**	**77.2**	**42.9**	**69.7**	59.5	**63.9**	76.1	70.2	**69.2**	**43.9**	58.3	**59.7**	77.2	64.8	**64.0**
Gt seg	67.8	70.3	61.5	56.6	38.2	71.2	74.7	75.7	41.6	68.3	59.0	63.2	75.6	71.0	68.7	42.6	59.5	59.1	**78.4**	**65.3**	63.4
Weak seg	68.9	71.3	59.0	54.2	37.3	71.9	74.5	75.2	40.8	67.6	**59.8**	62.8	**76.4**	**71.3**	68.4	43.8	**59.9**	57.2	76.6	64.4	63.0
No	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

TABLE 6: Comparison of detection accuracy on `VOC07-test` for the single-category experiment. The models are trained independently on each category, by using the `VOC12train-seg`. The first column specifies the type of object mask used for augmentation: ground-truth instance segmentations (Inst. Seg.), ground-truth semantic segmentation (GT Seg.), or weakly-supervised semantic segmentations (Weak Seg.). Inst. seg. stands for the original instance segmentation ground truth masks. The numbers represent AP per class in %. The best result for a category is in bold. All numbers are averaged over 3 independent experiments.

Aug. type	aero	bike	bird	boat	bott.	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	avg.
Inst. seg	**69.9**	73.8	**63.9**	**62.6**	35.3	**78.3**	73.5	80.6	**42.8**	**73.8**	62.7	**74.5**	81.1	73.2	68.9	38.1	**67.8**	64.3	79.3	**66.1**	**66.5**
GT Seg.	68.7	74.5	60.1	60.0	34.9	75.4	**74.4**	**81.7**	41.1	72.4	**64.2**	74.4	**81.3**	**74.6**	**69.6**	**39.7**	67.6	64.2	**80.4**	65.5	66.2
Weak Seg.	69.2	**75.2**	63.2	59.8	**35.6**	77.1	73.4	78.7	41.3	72.9	62.8	72.7	79.6	72.5	68.1	39.2	67.6	**66.1**	79.5	64.2	65.9
No	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

TABLE 7: Comparison of detection accuracy on `VOC07-test` for the multy-category experiment depending on the type of object masks used for augmentation. The models are trained on all categories together, by using the $1\,464$ images from `VOC12train-seg`. The first column specifies the type of object mask used for augmentation: ground-truth instance segmentations (Inst. Seg.), ground-truth semantic segmentation (GT Seg.), or weakly-supervised semantic segmentations (Weak Seg.). Inst. seg. stands for the original instance segmentation ground truth masks. The numbers represent AP per class in %. The best result for a category is in bold. All numbers are averaged over 3 independent experiments.

segmentation masks for augmentation purposes.

When the semantic maps are estimated, we follow the augmentation routines of the previous section with only one difference; specifically, an instance is kept if the bounding box of its segmentation covers at least 40% of its corresponding ground truth box. Otherwise, the object is not used for data augmentation. The results of applying this strategy to the single- and multy-category object detection are presented in Table 6 and 7, respectively. Table 6 shows which categories are unable to provide high-quality masks, even though the quality seems to be sufficient to improve upon the non-augmented baseline. It is surprising that by using object boxes instead of segmentation masks we lose only 0.6% of mAP in the multi-class scenario while still outperforming non-augmented training by 1.6%. These results show that the method is widely applicable even in the absence of segmentation annotations.

4.8 Studying the Importance of Context Modeling Quality for Scene Understanding

First, we make an assumption that the quality of a context model is mainly influenced by the amount of data it has received for training. Hence, to study this relation, we mine a bigger dataset `VOC07-trainval+VOC12-trainval` which results in 16551 images. Then, we proceed by taking subsets of this dataset of increasing size and train the context model on them. Finally, we use the obtained context models to augment `VOC12-trainval` and train BlitzNet300 on it for detection and segmentation. Table 8 summarizes the object detection performance on `VOC07-test` and semantic segmentation performance on `VOC12val-seg`. In the current experiment, 10% of the full set (1655 images) is roughly equal to the size of the `VOC12train-seg` (1464 images) initially used for training the context model. As we increase the data size used for context modeling, we can see how both detection and segmentation improve; however, this gain diminishes as the data size keeps growing. This probably mean that to improve scene understanding, the context model has to get visual context "approximately right" and further improvement is most likely limited by other factors such as unrealistic generated scenes and limited number of instances that are being copy-pasted. On the other hand, if the context model is trained with little data, as in the case of using only 5% of the full set, our augmentation strategy tends to the random one and shows little improvement.

% of data used	0	5	10	25	50	75	100
Det. mAP	64.6	65.3	66.1	66.4	66.7	66.9	66.9
Seg. mIoU	63.3	64.6	65.1	65.3	65.5	65.9	66.0

TABLE 8: Object detection and semantic segmentation performance depending on amount of data used for building the context model. First row depicts the portion (in %) of the `VOC0712trainval` used for training the context model. Second column corresponds to performance of baseline models. The second row gives the final detection mAP % evaluated on `VOC07test`, while the third row lists segmentation mIou in % on `VOC12val-seg`. For both tasks we used BlitzNet300 trained on augmented `VOC12train-seg`.

5 Conclusion

In this paper, we introduce a data augmentation technique dedicated to scene understanding problems. From a

methodological point of view, we show that this approach is effective and goes beyond traditional augmentation methods. 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 was so far unclear for scene understanding, we show that it is in fact crucial when performing data augmentation and learn with fewer labeled data, which is one of the major issues deep learning models are facing today.

ACKNOWLEDGMENT

This work was supported by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes.

Nikita Dvornik recieved the bachelor degree at Moscow Institute of Physics and Technology (MIPT) and master degree at INP Grenoble. He is currently working towards the PhD degree at INRIA Grenoble under supervision of Cordelia Schmid and Julien Mairal. His research interests include scene understanding tasks, such as object detection and semantic segmentaion, data augmentation, few-shot learning and learning general image representations under constraints.

Julien Mairal (SM16) received the Graduate degree from the Ecole Polytechnique, Palaiseau, France, in 2005, and the Ph.D. degree from Ecole Normale Superieure, Cachan, France, in 2010. He was a Postdoctoral Researcher at the Statistics Department, UC Berkeley. In 2012, he joined Inria, Grenoble, France, where he is currently a Research Scientist. His research interests include machine learning, computer vision, mathematical optimization, and statistical image and signal processing. In 2016, he received a Starting Grant from the European Research Council and in 2017, he received the IEEE PAMI young research award. He was awarded the Cor Baayen prize in 2013 and the test-of-time award at ICML 2019.

Cordelia Schmid holds a M.S. degree in Computer Science from the University of Karlsruhe and a Doctorate, also in Computer Science, from the Institut National Polytechnique de Grenoble (INPG). She is a reserach director at Inria Grenoble. She has been an editor-in-chief for IJCV (2013–2018), a program chair of IEEE CVPR 2005 and ECCV 2012 as well as a general chair of IEEE CVPR 2015. In 2006, 2014 and 2016, she was awarded the Longuet-Higgins prize for fundamental contributions in computer vision that have withstood the test of time. She is a fellow of IEEE. She was awarded an ERC advanced grant in 2013, the Humbolt research award in 2015 and the Inria & French Academy of Science Grand Prix in 2016. She was elected to the German National Academy of Sciences, Leopoldina, in 2017. In 2018 she received th Koenderink prize for fundamental contributions in computer vision that have withstood the test of time. Starting 2018 she holds a joint appointment with Google research.

REFERENCES

[1] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[4] Y. LeCun, "The mnist database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[5] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[6] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *arXiv preprint arXiv:1702.07836*, 2017.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[9] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[10] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[13] A. Torralba and P. Sinha, "Statistical context priming for object detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001.

[14] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[16] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[17] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[18] R. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[20] W. Chu and D. Cai, "Deep feature based contextual model for object detection," *Neurocomputing*, vol. 275, pp. 1035–1042, 2018.

[21] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "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.

[22] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[23] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[24] E. Barnea and O. Ben-Shahar, "On the utility of context (or the lack thereof) for object detection," *arXiv preprint arXiv:1711.05471*, 2017.

[25] R. Yu, X. Chen, V. I. Morariu, and L. S. Davis, "The role of context selection in object detection," in *British Machine Vision Conference (BMVC)*, 2016.

[26] B. Yao and L. Fei-Fei, "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] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[28] X. He, R. S. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[29] J. Yang, B. Price, S. Cohen, and M.-H. Yang, "Context driven scene parsing with attention to rare classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[30] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[31] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision (IJCV)*, vol. 43, pp. 29–44, 2001.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[34] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, pp. 834–848, 2018.

[36] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, pp. 1352–1366, 2018.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[38] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," *arXiv preprint arXiv:1801.02385*, 2018.

[39] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[42] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision (IJCV)*, vol. 126, pp. 973–992, 2018.

[43] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[45] W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[46] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem, "Rendering synthetic objects into legacy photographs," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, p. 157, 2011.

[47] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[48] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "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.

[49] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] R. Barth, J. Hemming, and E. J. van Henten, "Improved part segmentation performance by optimising realism of synthetic images using cycle generative adversarial networks," *arXiv preprint arXiv:1803.06301*, 2018.

[51] L. Sixt, B. Wild, and T. Landgraf, "Rendergan: Generating realistic labeled data," *Frontiers in Robotics and AI*, vol. 5, p. 66, 2018.

[52] Z. Liao, A. Farhadi, Y. Wang, I. Endres, and D. Forsyth, "Building a dictionary of image fragments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[53] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] P. Prez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (SIGGRAPH'03)*, vol. 22, no. 3, pp. 313–318, 2003.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[57] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[59] J. Yang, J. Lu, D. Batra, and D. Parikh, "A faster pytorch implementation of faster r-cnn," *https://github.com/jwyang/faster-rcnn.pytorch*, 2017.

[60] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[61] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[63] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *International Conference on Learning Representations (ICLR)*, 2015.