

Self-supervised Blur Detection from Synthetically Blurred Scenes

Aitor Alvarez-Gila^{a,b,*}, Adrian Galdran^c, Estibaliz Garrote^a, Joost van de Weijer^b

^aTECNALIA, Derio, Spain

^bCVC - Universitat Autònoma de Barcelona, Barcelona, Spain

^cÉcole de Technologie Supérieure, Montréal, Canada

Abstract

Blur detection aims at segmenting the blurred areas of a given image. Recent deep learning-based methods approach this problem by learning an end-to-end mapping between the blurred input and a binary mask representing the localization of its blurred areas. Nevertheless, the effectiveness of such deep models is limited due to the scarcity of datasets annotated in terms of blur segmentation, as blur annotation is labour intensive. In this work, we bypass the need for such annotated datasets for end-to-end learning, and instead rely on object proposals and a model for blur generation in order to produce a dataset of synthetically blurred images. This allows us to perform self-supervised learning over the generated image and ground truth blur mask pairs using CNNs, defining a framework that can be employed in purely self-supervised, weakly supervised or semi-supervised configurations. Interestingly, experimental results of such setups over the largest blur segmentation datasets available show that this approach achieves state of the art results in blur segmentation, even without ever observing any real blurred image.

Keywords: blur, defocus, motion, deep learning, self-supervised learning, synthetic

2010 MSC: 68T45, 62H35

1. Introduction

Image blur is a phenomenon that degrades the definition of an image, producing a loss of detail in the affected regions. The two main causes leading to a fully or partially blurred image are (i) defocus blur, which is inherent to a wide aperture optical image capturing device that projects scene points that are away from the focus plane onto a non-punctual circle of confusion on the sensor and (ii) motion blur, which is caused by the movement of either the camera (*i.e.* camera shake) or the imaged objects during camera exposure time. Even if both defocus and object-motion blurs are sometimes sought-after as part of a creative photographic process (*e.g.* to pop-out the subject or to evoking a sense of motion, respectively), most of the time blur is considered as an undesired effect [1] or image artifact.

In any case, trying to localize blurred regions within an image (or, equivalently, achieving a segmentation in terms of blurry/unblurred parts) is a useful task with a wide range of applications in computational photography, *e.g.* defocus blur magnification [2, 3], image deblurring [4, 5, 6, 7], or camera focus point or depth of focus estimation [3]. In addition, due the underlying correlation between blurred and non-blurred areas within the

same image, blur detection has been also applied in general computer vision tasks, *e.g.* depth estimation [8], saliency prediction [9] or semantic object segmentation [10]. However, blurred region segmentation is a challenging task, due to the fundamental ambiguities existing between the out-of-focus pixels and the originally flat regions or smooth edges. Scale-ambiguity (*i.e.* the difficulty of inferring the level of blur over one single scale [11]) and the dependence of the perception of sharpness on the image size are additional challenges that affect the performance of current approaches.

A number of previous works have investigated blur localization directly or implicitly from the feature engineering and physical modeling approaches, either taking one single [12, 11, 3] or multiple [13, 14, 15] images as input, and aiming at detecting only one [16, 12, 6, 17, 18, 19, 20] or both kinds of blur [21, 22, 23]. Most of these try to leverage information extracted directly from the intensities [21], from the gradients [24, 23, 25, 20], or from transformed domains [22, 16, 12, 11, 26].

More recently, supervised learning-based approaches [25], and particularly those based on the use of Convolutional Neural Networks (CNN), have shown enormous potential for tackling tasks that require a dense, per-pixel prediction, such as semantic segmentation [27, 28], instance segmentation [29] or crowd counting via density map estimation [30]. Blur segmentation can also be viewed as one of such dense prediction tasks, and several works have already explored this approach,

*Corresponding author

Email addresses: aitor.alvarez@tecnalia.com
(Aitor Alvarez-Gila), adrian.galdran-cabello.1@ens.etsmtl.ca
(Adrian Galdran), estibaliz.garrote@tecnalia.com
(Estibaliz Garrote), joost@cvc.uab.es (Joost van de Weijer)

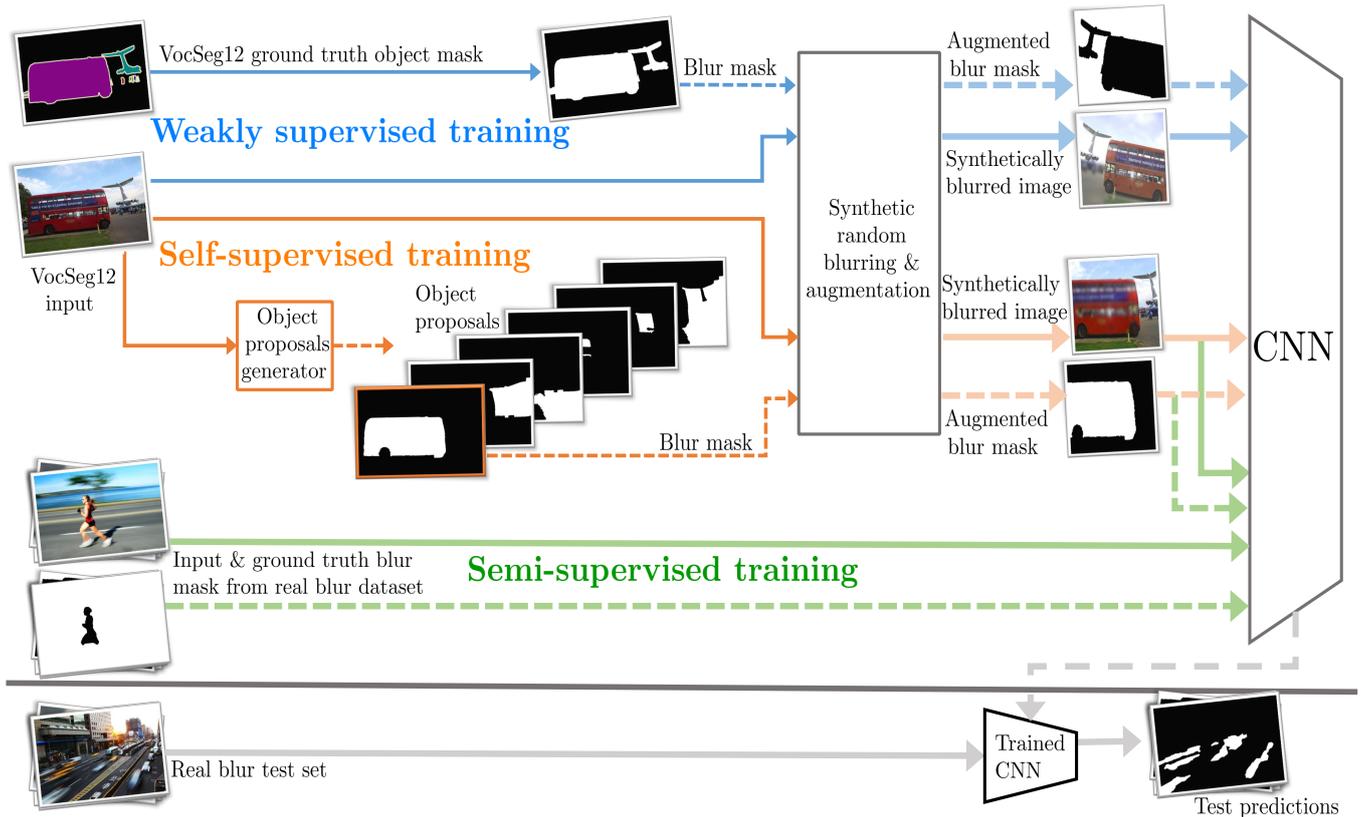


Figure 1: General overview of our framework train and testing processes, with each path color representing one of its three possible instantiations *i.e.* self-supervised, weakly-supervised and semi-supervised approaches.

either for predicting both types [31, 1, 32] or defocus only blur [33, 34, 35, 36].

Nevertheless, the performance gain obtained by these fully-supervised, CNN-based approaches trained end-to-end is relatively modest when compared to gains in other fields. The main bottleneck which hinders the full power of CNNs in their application to the field of blur localization is the absence of large enough datasets with pixel-wise annotations. Several recent efforts have partially addressed this void [11, 1, 35, 36], but these manual annotations are scarce, labour intensive and costly to acquire.

In the absence of large public sets of annotated data, the generation of synthetic image degradations has been successfully applied for training deep CNNs in recent years. This approach has been adopted for computer vision tasks as diverse as image super-resolution [37], inpainting [38], image quality assessment [39] or perceptual similarity estimation [40]. It is also a core component of many self-supervised learning methods, which, facing the lack of large sets of annotated data for a certain final task, define a pretext task (*e.g.* inpainting [38], warped image matching [41], artifact spotting [42]) for which ground truth labels can be automatically derived without manual intervention and be used for feature learning. A particular case of such scenario would be that of pretext and final tasks being equal.

Synthetic blur generation has also been explored as part of deblurring workflows [43, 5], as the blur generation process has been extensively studied, especially within deconvolution approaches for image deblurring. In the context of blur localization, however, synthetic blur operations have only been applied either over very simple partial blur masks (*e.g.* consisting of a hard rectilinear partition of the image in halves [36]) or globally over full patches of reduced size extracted from the whole image [44].

Our main contribution in this work is the introduction of a deep self-supervised partial blur detection framework, which successfully localizes both defocus and object-motion blur types for a single image without making use of any blur segmentation annotation for training the model. Instead, we circumvent the lack of large annotated blur detection image sets by selectively applying procedural synthetic blurring operations to varying regions within images taken from an unrelated non-annotated dataset of natural images. By controlling the definition of the regions being synthetically blurred, we can automatically generate the associated ground-truth blur masks on the fly. Fig. 1 illustrates the process.

The framework comprises three different instantiations, defined by i) how those regions are extracted and ii) whether the training is purely based on synthetically blurred images or not:

1. A **self-supervised** approach, in which the regions to be blurred are defined by an object proposal [45] model, which can generate class-agnostic plausible object masks and thus generalize well across a variety of datasets. This method can extract multiple different blur masks per image, so that the model learns from different image and blur-mask pairs. This allows us to obtain a highly variable training image stream to feed our CNN-based semantic segmentation model at train time.
2. A **weakly-supervised** approach, in which such regions are instead selected from the ground truth labels of any given semantic segmentation image set.
3. A **semi-supervised** approach, in which the synthetically generated blurred image and labels from the object proposal method are used in conjunction with real partially blurred images and their respective manual blur-mask annotations in order to augment the given fully supervised blur segmentation dataset.

By extensively evaluating the learned model on the two largest publicly available pixel-wise annotated real blur localization datasets [11, 35], we show that our approach generalizes adequately for both blur types, improving the performance of all the considered classic approaches and that of recent fully supervised deep CNN-based ones that require large human-labeled training sets. Experimental results show that the proposed solution can be successfully employed to train a deep blur segmentation model, either without the need for any specific blur localization dataset or by making use of a very reduced set of images exhibiting real blur degradations. This is especially important for domains other than that of natural RGB images, for which no labeled blur masks exist, such as infrared or multispectral [46] imagery, medical imaging [47] or text documents [48].

2. Synthesizing realistic blur

In this paper, we are interested in learning to localize blur on partially blurred images without using any labeled example for this specific task (in the self-supervised and weakly supervised approaches). We therefore switch to the problem of generating plausible blurred scenes from non-blurred images.

2.1. Blur mask extraction

The first step of the process is the determination of the parts of the image that will be subject to the synthetic blurring operation, to which we refer as the *blur mask*.

2.1.1. Semantic object masks as blur masks

One possible approach to achieve this is that of using a dataset of images on which several objects have been manually segmented, without any explicit relation to its blur

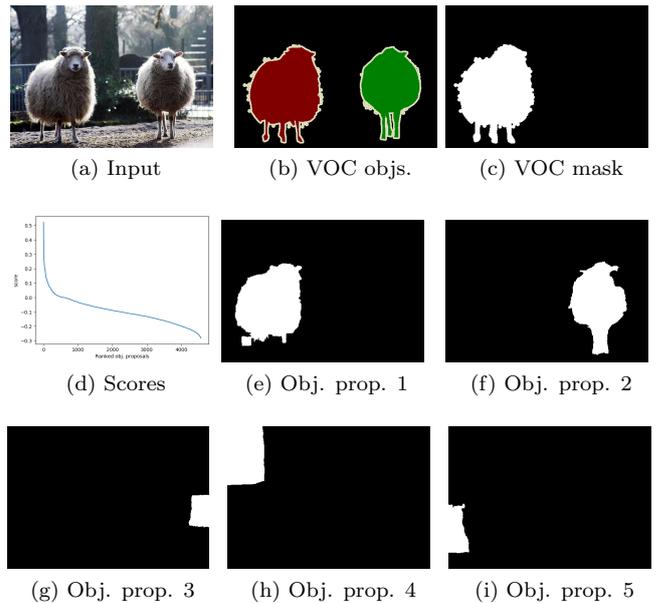


Figure 2: Blur mask extraction from an input image (a) of the Pascal VOC 2012 dataset [49]. (b) Ground truth object masks from the segmentation challenge. (c) Blur mask given by the largest connected component. (d) Sorted scores of objectness given by MCG for this input. (e-i) First five object proposals generated by MCG [50].

content. This effectively implies making use of a distant supervision (one grounded on easier to obtain semantic object annotations) for the final blur segmentation task. We refer to this as our weakly supervised approach.

This kind of labeled data is readily available from datasets generated for semantic segmentation tasks, such as the Pascal VOC Segmentation challenge 2012 (SegVOC12) [49]. This dataset includes ground truth annotations of the most prominent objects present in the image (see Fig. 2b) corresponding to 20 different classes. Under this setting, at train time, we build the blur mask for each input image by computing the connected components of the largest object present in the corresponding ground truth (Fig. 2c).

2.1.2. Object proposals as blur masks

Our purely self-supervised method takes one step ahead by removing the need for manual object segmentation, and replacing it with the inclusion of a class-agnostic object mask proposal generation step. The goal of object proposal generation methods is, given an input image, to yield a set of either bounding boxes or segmentation masks that correspond to different object location hypotheses. Its primary application is serving as a first candidate location filtering stage of two-phase object detection methods, so that more resources can be allocated for the representation and analysis of the resulting subset of regions.

The Multiscale Combinatorial Grouping (MCG) object proposal generation algorithm [50] represents one of the most accurate approaches of its kind. Moreover, although

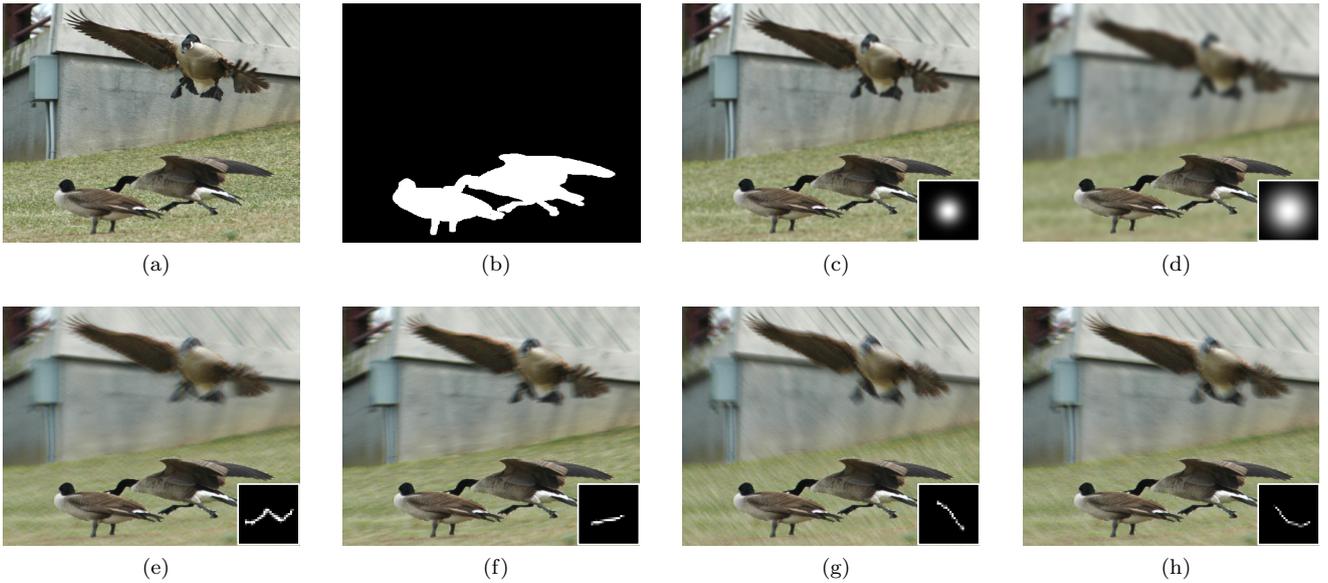


Figure 3: (a) Input image from the VOC dataset (b) Segmented Foreground (c-d) Blurring of input image with Gaussian blur, $\sigma = 1.5, 3$ pixels (e-h) Blurring of input image with randomly-generated non-linear motion blur. Blur kernels displayed at the right-bottom on each image. All images have been blurred with the halo-artifact removal described in section 2.3. Blur differences are better appreciated when focusing in the roof on the top of the image.

some learning-based steps are involved in its model creation process, its ability to generalize across different datasets renders this method virtually parameter-free. It yields a ranked set of segmented object proposals after a process that comprises: (i) a multi-scale input image segmentation step (based on low level features), (ii) a rescaling and alignment of the segmentation results, (iii) the combination of such results onto a merged multi-scale hierarchy of binary spatial image partitions (*i.e.* regions) and (iv) a final combinatorial grouping stage, which explores the region tree looking for sets of regions that, merged together, are likely to represent complete objects. The resulting set of -hundreds of- proposals is then ranked in accordance to a score representing such likelihood. During training, we apply MCG to every input image and, at each epoch, we randomly sample a blur mask from the probability distribution given by applying a softmax operation over the mentioned scores.

Fig. 2 shows an example of blur mask extraction for both ground truth semantic object mask (b-c) and MCG-based object proposal (e-h) methods. It is important to note that, should we directly use the blur mask extracted as described in either case, we would be introducing a strong bias in the blur detection training, favoring the prediction of blurred elements in the foreground or background of the images. This is due to a comparatively significant amount of the extracted blur masks corresponding to foreground objects. In order to mitigate this and promote the invariance of the model to this respect, we invert the blur mask with a probability p_{inv} .

2.2. Synthetic blur

Once the input blur mask has been created, we can now randomly apply different kinds of synthetic blur operations over it. Given an image $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ for which the image domain Ω has been already partitioned into background Ω^B and foreground Ω^F , represented as the blur mask, we generate a partially blurred version of I by first defining a blur kernel K . Then, the resulting artificially blurred image I_b can be easily obtained by computing:

$$I_b(x, y) = \begin{cases} K * I(x, y), & \text{for } (x, y) \in \Omega^B, \\ I(x, y), & \text{for } (x, y) \in \Omega^F. \end{cases} \quad (1)$$

The adequate definition of the kernel K is critical in order to accomplish the goal of generating realistic blur. While blur coming from situations on which an object is in-focus and the background is defocused can be easily simulated by Gaussian kernels K_σ of varying standard deviations σ , blurs of different natures, *e.g.* motion blur, are harder to emulate. Since our aim is to produce fast blurred versions of a given background on the fly, we design a simple pipeline for generating non-linear motion blur:

1. Build a horizontal line of length $1 \times m$ in a discrete domain $\Omega_k \subset \Omega$, obtaining a kernel K_m that defines a linear horizontal motion blur.
2. Rotate it by α degrees, obtaining a kernel $K_{(m,\alpha)}$ that defines a linear diagonal motion blur.
3. Apply an elastic deformation $\mathcal{E} : \Omega_k \rightarrow \Omega_k$ of the underlying image grid Ω_k that turns $K_{(m,\alpha)}$ into a non-linear kernel $K_{(m,\alpha,\mathcal{E})}$.



Figure 4: (a) Original image from the VOC dataset (b) Inpainted foreground (c,d) Naively blurred background (e,f) Result of blurring after inpainting background.

In training time, a random decision of whether to apply defocus blur or motion blur is made for each training image. The definitions of σ or (m, α, \mathcal{E}) are also randomized by drawing parameters from a suitable range of values. In this way, the same image can be transformed in infinitely many ways.

In Fig. (3a) we show one image from our Pascal VOC training set. In Figs. (3c) and (3d) the background, as given by Fig. (3b), is blurred with two Gaussians of varying standard deviations, whereas in Figs. (3e–3h) different non-linear motion blurs are applied. It should be noted that there exist more sophisticated mechanisms to simulate non-linear image blurring based on physical considerations [43]. However, for the purposes of this work we prioritize a simple and efficient strategy like the one described above.

Note that, although ideally we would want to use perfectly sharp images to perform the selective blurring, the employed SegVOC12 dataset does indeed contain some pictures exhibiting preexisting partial blur. This means that the artificially generated ground truth will contain a certain amount of noisy pixel-level labels. In spite of this, as shown in section 4, our framework is able to successfully learn to segment real blur.

2.3. Removing halo artifacts by inpainting

The naive approach of blurring the image, and then placing back the unblurred foreground on its original location has some disadvantages. Namely, this procedure leads to the appearance of halo artifacts around the borders of the foreground, as shown in Figs. (4c) and (4d). The reason of this problem is that sharp intensity jumps at borders of foreground objects artificially distort image statistics when averaging with a blur kernel around those pixels.

If a model is trained with images containing these halos, it will likely learn to localize blur by simply finding the position of such artifacts. In order to avoid this situation, we propose a different approach to obtain a blurred-background image. After extracting the sharp foreground from a given image, we proceed to inpaint the foreground pixels applying the method from [51], as shown in Fig. (4b), before blurring the background. This process allows to remove halo artifacts from the artificially blurred scenes before supplying them to our model, see Figs. (4e) and (4f).

3. Convolutional Neural Networks for Blur Segmentation from Synthetic Data

3.1. Architecture

Section 2 provides us with all the necessary tools to create a procedural -and thus, highly variable- training image flow. By posing blur localization as a dense, per-pixel classification task, we can make use of one of the well-established set of deep architectures devoted to semantic segmentation and feed it with such synthetically distorted image stream for training. We consciously avoid the *ad hoc* design of an architecture specifically suited for our target task, and instead rely on an off-the-shelf deep network, which allows us to isolate the contribution derived from the proposed training procedure.

We select the DeepLabv3[28] Network as our reference CNN architecture, since it has recently shown state of the art performance on various dense labeling tasks. This is partly due to the fact that it features an effective receptive field significantly larger than those of other standard architectures. These are enabled by the use of *atrous* convolutions (first introduced in this context in [52, 53]), which also contribute to the attainment of output maps with a large spatial resolution. It has been previously shown that high semantic level features can greatly contribute to solve the task of blur localization [31]. Hence, we can expect that the use of wide receptive fields, covering significant parts of the scene, will bring important benefits to our technique. The DeepLabv3 architecture also benefits from the fusion of feature maps from multiple scales (which has proven crucial for solving the scale ambiguity problem [32]): the use of a new *Atrous Spatial Pyramid Pooling* (ASPP) module helps capture context at various ranges. In this work, we employ the DeepLabv3-ResNet101 variation of the architecture, which is constructed by a Deeplabv3 model with a ResNet-101 backbone. We start from a model initially pre-trained on a subset of the COCO train2017 dataset [54] containing the 20 classes also present in the Pascal VOC 2012 segmentation dataset, and we substitute its final classifier with a $1 \times 1 \times 256 \times 2$ 2D convolution, followed by a *logSoftmax* operation.

3.2. Training procedure

Our training set for the purely self-supervised and weakly supervised approaches is generated by randomly

applying synthetic defocus and motion blurring operations (generated as described in section 2) to image batches (with a batch size of 18) from the training set of the SegVOC12 dataset [49]. Such degradations are selectively applied according to the binary mask created on the fly as mentioned in sections 2.1.1 and 2.1.2 for the ground truth semantic object masks and object proposal-based mask, respectively, and the blur mask itself is used as the ground truth label.

Additionally, the usual standard random data transformations (affine transforms, flips, color jitter, cropping) are applied to the input and target image pairs before resizing them from their original size to 224×224 pixels. A randomly applied JPEG compression-based augmentation is also performed at different stages of the input image pre-processing workflow, with the aim of gaining invariance to low level, dataset-specific regularities.

We employ a negative log-likelihood loss that we minimize using the Adam optimizer, and let the model train until the validation loss stagnates for 20 epochs. A reduced number of hyperparameter tuning configurations was tried for each of the experimental setting, and the setup yielding the lowest validation loss was kept for evaluation. In most of the configurations, this corresponded to a learning rate value of 10^{-5} and a weight decay of $5 \cdot 10^{-4}$.

4. Experimental Results

We evaluate the trained model on the largest publicly available¹ dataset of images annotated in terms of defocus and object-motion blur localization, i.e. that of Shi *et al.* [11]². The dataset consists of 1000 partially blurred natural images with human-made binary (blur/no-blur) pixel-wise annotations, of which 704 correspond to defocus blur and 296 to motion blur-affected images. Following [31], we partition the dataset in an odd and an even subset, both of them containing an approximately equal amount of images affected by both types of blurs), and keep the latter held out for testing purposes. Unless otherwise stated, all the results shown in this section were thus evaluated on the 500 images from Shi *et al.*'s even subset. Meanwhile, a 20% of the odd subset (100 images) is used for validation, and the remaining 400 images are used for training in those experimental setups that require a supervised training component (i.e. our semi-supervised approach and the fully supervised model from Table 1 and Fig 6).

At inference, we run a simple test-time augmentation (TTA) process, consisting of averaging the predictions yielded by the original input image, together with its horizontally flipped version, and upscale the resulting 224×224 pixels-sized predictions to the original, varying image sizes

before performing the evaluation. Even though the raw performance values could, to some extent, benefit from performing a re-scaling of the resulting blur map in the $[0, 1]$ range, we purposefully avoid such kind of post-processing, in order to enable the prediction of completely sharp or blurred scenes. Consequently, we can consider the values of the predicted mask pixels as a measure of absolute blurriness, as opposed to a measure of blur level of each region with respect to the sharpest area of the image.

4.1. Self-supervised setup

We first consider the purely self-supervised instantiation of our framework, in which we directly test the model trained on synthetically blurred images over the 500 even samples of the test dataset. We compare our method against most of the best performing hand-crafted feature-based approaches which do not require any dedicated dataset for training: Liu *et al.* [21], Chakrabarti *et al.* [22], Su *et al.* [23], Shi *et al.* [11], LBP [18] and HiFST [3]. In addition, we include the performance reported for the same even subset by one of the most recent deep CNN-based defocus and motion blur detection methods in the literature, i.e., the *Deep Blur Mapping* approach by Ma *et al.* [31].

Table 1 shows the obtained results in terms of Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) and Average Precision (AP), as computed over the different values of recall given by the Precision-Recall (P-R) curve. Both the AUC and AP values were computed individually for each output map and then averaged over the whole even test subset. The overall performance value (*All*) is also shown disaggregated for the *Defocus* and *Motion* subsets of the database. Note that, as is the case with most of the other methods, in absolute terms our approach performs better over the defocused images than over the motion-blurred ones. This will hold true for the results of all the three variants of our framework.

From Table 1 we can see that our self-supervised approach performs significantly better than all the other non-deep methods for every metric and blur-type subset. Furthermore, the proposed self-supervised learning method, when used to train the off-the-shelf DeepLabv3_resnet101 network and without ever observing a single image with real blur, yields better overall AUC and AP values than Ma *et al.*'s CNN architecture [31], whose design was tuned *ad hoc* for this task and trained end-to-end in a fully supervised setup over the 500 odd samples of the dataset³. In particular, our method performs close to Ma *et al.*'s on the defocus blur subset, but significantly better for the motion blur samples. The last row of Table 1 (i.e. *Fully*

¹Zhang *et al.*'s dataset [1] and Zhao *et al.*'s supplementary train set [36] are not open.

²Code and models are available on <https://github.com/aitorshuffe/synthblur>.

³The performance evaluation protocol in [31] is slightly different, as they compute a single AP value for a one-dimensional vector containing the predictions of all the pixels of every image in the even subset. Although we believe that the AUC/AP values should be computed on a per-image basis, under their protocol our self-supervised overall AP is 0.952, vs. their reported value of 0.880

| Method | AUC | | | AP | | |
|--|--------------|--------------|--------------|--------------|--------------|--------------|
| | Defocus | Motion | All | Defocus | Motion | All |
| Liu <i>et al.</i> [21] | 0.722 | 0.714 | 0.720 | 0.792 | 0.683 | 0.760 |
| Chakrabarti [22] | 0.745 | 0.640 | 0.714 | 0.837 | 0.675 | 0.789 |
| Su <i>et al.</i> [23] | 0.807 | 0.750 | 0.790 | 0.859 | 0.707 | 0.814 |
| Shi <i>et al.</i> [11] | 0.836 | 0.735 | 0.806 | 0.876 | 0.699 | 0.823 |
| LBP[18] | 0.855 | 0.678 | 0.802 | 0.876 | 0.683 | 0.819 |
| HiFST[3] | 0.901 | 0.804 | 0.873 | 0.928 | 0.744 | 0.874 |
| Ma <i>et al.</i> [31] | 0.947 | 0.861 | 0.922 | 0.966 | 0.784 | 0.912 |
| Ours self-supervised | 0.945 | 0.905 | 0.933 | 0.960 | 0.838 | 0.924 |
| Ours weakly supervised (segmentation masks) | 0.941 | 0.897 | 0.928 | 0.959 | 0.849 | 0.926 |
| Ours semi-supervised (joint with 400 odd img.) | 0.956 | 0.904 | 0.941 | 0.974 | 0.840 | 0.934 |
| Fully supervised (finetuned to 400 odd img.) | 0.943 | 0.875 | 0.923 | 0.965 | 0.819 | 0.922 |

Table 1: Quantitative evaluation over Shi *et al.*’s dataset’s [11] even partition. **Best**, **2nd best** and **3rd best** results are highlighted for each metric and blur type.

supervised) shows the results obtained by fine-tuning the DeepLabv3_resnet101 network over Shi’s odd subset. The mixed results of this fully supervised model when compared to Ma *et al.* suggest that the performance gain of our self-supervised method is largely due to the benefits of our training scheme, rather than being just a product of the use of a better architecture.

Other relevant fully supervised CNN based results were left out of this comparison for various reasons: Zhao *et al.* [35] target exclusively images degraded with defocus blur, and their training procedure involves (i) pre-training the model employing very simple artificial defocus blurring operations over half of the image on samples from additional datasets and (ii) further fine-tuning it over 604 out of the 704 images of Shi *et al.*’s defocus blur partition and testing it over the remaining 100. Finally, Zhang *et al.* [1] train their *ad hoc* designed *ABC-FuseNet* architecture end-to-end on their unreleased *SmartBlur* blur segmentation dataset of 10,000 images before evaluating on Shi *et al.* Even with such amount of training samples, their reported AP is 0.869 for the whole dataset, sensibly below our results on the even partition.

Fig.5 contains visual results for a small random subset of images affected by both types of blur (defocus blur in the top seven rows, motion blur in the bottom seven), as predicted for most of the considered methods. We can observe that, even without the utilization of any single ground truth blur segmentation annotation from the target dataset for direct supervision, our self-supervised approach obtains accurate masks, comparable in visual quality to those produced by fully-supervised deep CNN-based methods, such as [31].

Blur type ablation. Finally, Table 2 shows the results obtained over the same sets when only defocus blur or only motion blur synthetic degradations were applied during training in our self-supervised setup. The results suggest

| | AUC | | | AP | | |
|-----|--------------|--------------|--------------|--------------|--------------|--------------|
| | DF | MT | All | DF | MT | All |
| DF | 0.949 | 0.826 | 0.913 | 0.967 | 0.773 | 0.910 |
| MT | 0.934 | 0.894 | 0.922 | 0.953 | 0.831 | 0.917 |
| All | 0.945 | 0.905 | 0.933 | 0.960 | 0.838 | 0.924 |

Table 2: Blur type based ablation test over Shi *et al.*’s dataset’s [11] even partition, in our self-supervised setup. Rows represent the synthetic blur type being applied on training (DF=Defocus, MT=Motion, All=Defocus and Motion). Columns represent the test (sub)set. Bold is best.

that devoting the full capacity of the network to learning to detect only one specific type of degradation does help in the case of defocus blur training, but not so when the training is constrained to observing samples affected solely by object motion. For the latter, the increase of variability introduced by feeding the net with both kinds of degradations seems beneficial, probably due to a regularization effect. Cross-blur type evaluation is asymmetric: while training on motion-only blur achieves decent results on the defocus-only test subset, a model trained uniquely on defocus blur synthetic degradations suffers a significant performance degradation if applied to object motion-affected images.

In addition, this experiment reveals one of the advantages inherent to our blur detection framework: by selectively tuning the ratio of images being synthetically blurred with defocus or motion kernels during training, we can prioritize the performance over either kind of blurs, or operate anywhere in between.

4.2. Weakly supervised setup

The *Ours weakly supervised*-labeled row from Table 1 shows the results achieved with our weakly supervised approach, based on the use of manually annotated seman-

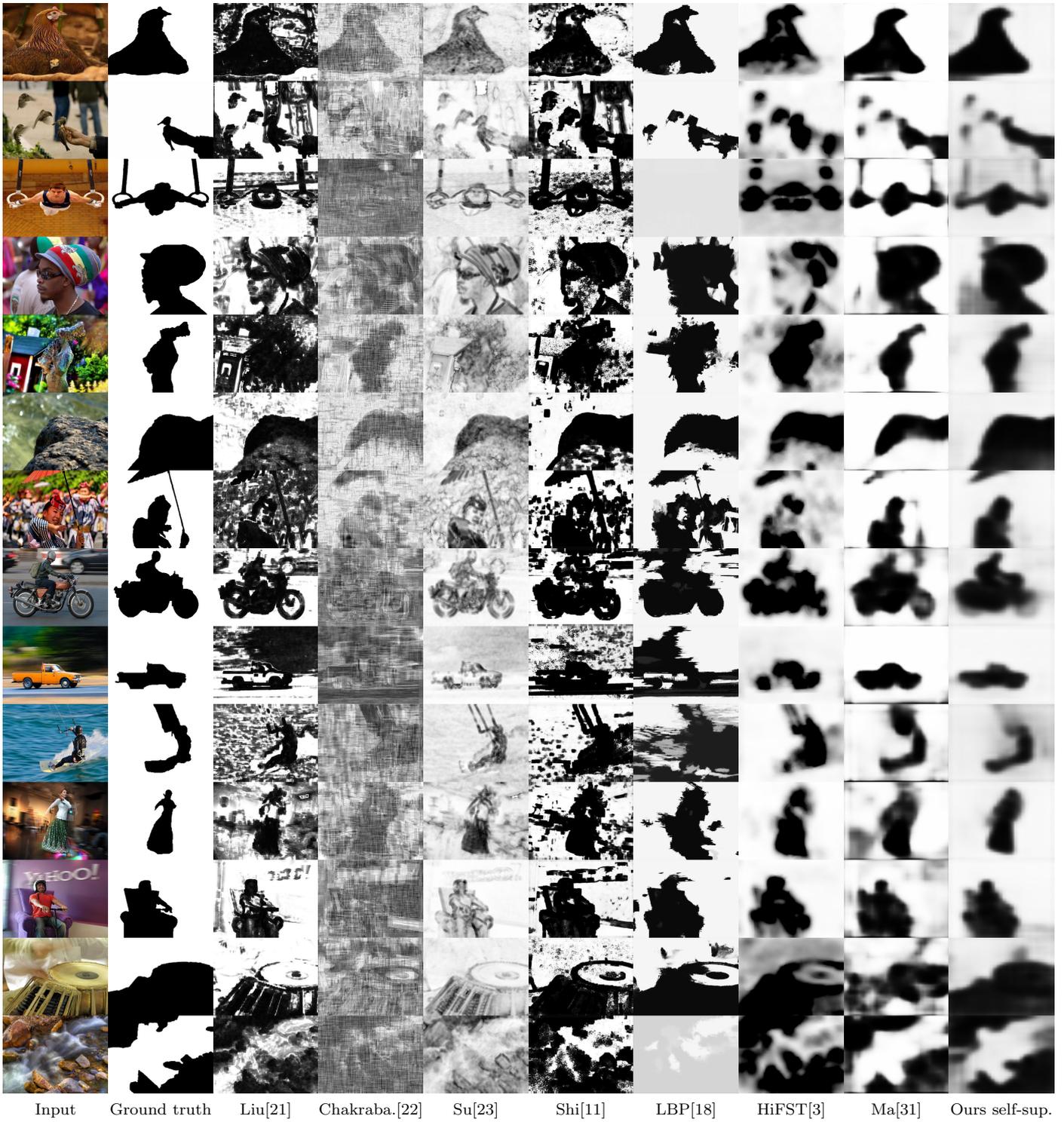


Figure 5: Qualitative results for a sample of images from Shi’s dataset[11] affected by defocus (top 7) and motion (bottom 7) blur processed by the different evaluated algorithms.

tic object segments as blur masks instead of the unsupervised proposals yielded by the MCG algorithm. While one could intuitively think of this as an upper bound for our self-supervised method’s performance, we observe that, in fact, both methods perform almost on par, with the self-supervised approach yielding slightly better results for the images affected by defocus blur and with no clear strongest method in terms of motion-blur and overall results, depending on the metric of interest. This experiment shows that applying blur degradations to object proposals computed with an object proposal method does not negatively impact our results, and that similar outcomes are obtained when compared to a method trained with ground truth segmentation masks.

4.3. Semi-supervised setup

We now introduce a modification in the training process in order to test the usefulness of our proposed synthetically blurred image-based training when a limited number of blur segmentation annotations are available for the target dataset. This corresponds to our semi-supervised experimental setting, in which a joint training is performed: mini-batches are now composed of equal amount of image and ground truth blur mask pairs produced (i) in a MCG-based synthetic blurring operation, and (ii) sampled from the train subset of the target dataset [11].

In order to assess the usefulness of this semi-supervised setting, we consider the 400 images from Shi *et al.*’s training set that were previously separated and employ a varying fraction of them for joint training with the synthetically blurred images. Specifically, we conduct the experiment with a 2%, 4%, 5%, 10%, 20%, 40% and 80% of the odd part being used as training aid, which, in absolute terms, correspond to 10, 20, 25, 50, 100, 200, and 400 images, respectively. The evaluation protocol is not affected, and the trained model is then tested on the 500 even pairs of the dataset.

We compare our semi-supervised results with those presented in the preceding sections (i.e. *Ours self-supervised* and *Ours weakly supervised* approaches from Table 1), whose performance metrics do not vary with the number of real images being used for joint training. Finally, for comparison we extend the fully supervised approach presented in section 4.1 by fine-tuning the same deeplabv3_resnet101 model over the same fractions of the odd train partition.

Fig. 6 shows the AUC and AP values obtained for each of the aforementioned methods, both as overall metrics and with disaggregated values for defocus and motion blur affected images. We observe that:

- For all the considered amounts of annotated images, our joint, semi-supervised approach outperforms the fully supervised one by a significant margin for both blur types, especially so as we operate with few real samples. This means that the use of synthetically generated blurred image-ground truth pairs has proven

useful as additional source of training data for improving the performance of fully supervised blur detection approaches trained end-to-end.

- The concrete AUC and AP values obtained by these variations for 400 real blur images were also added, for comparison, to Table 1. As shown there, the semi-supervised training scheme achieves the best overall results and, for every subset and metric, there is always one of the three instantiations of our framework outperforming every other deep fully supervised alternative. In particular, our best method (semi-supervised setting) reaches an overall AUC of 0.941 and an AP of 0.934, 0.019 and 0.022 points better than Ma *et al.*’s, respectively.
- Even the self-supervised or the technically simpler but more annotation-dependent weakly supervised variants of our framework can clearly outperform both their semi-supervised counterpart and, most notably, the fully supervised training in the lower part of the range.
- This gap is closed by the fully supervised training scheme only for the defocus blur subset, as we keep adding images with real blur, but not so in the case of the motion blur subset.

4.4. Cross-dataset generalization

One of the most frequent limitations of current end-to-end trained CNN-based solutions to many computer vision tasks is the inability of models trained on a certain dataset to generalize to other datasets with some underlying distribution shift, either revealed as some readily apparent visual difference (e.g. illumination, object appearance) or due to some hard-to-perceive low level statistical regularities within the dataset. Domain adaptation solutions [55] aim at mitigating the harmful effect of such domain shifts, but they are frequently based on domain-adversarial training strategies that can be cumbersome to implement, and mostly ineffective in providing a significant performance gain when both domains are easily told apart.

The following experiment seeks to evaluate this cross-dataset generalization ability for our self-supervised approach as compared to that of other methods. To that end, we introduce the dataset of defocus blurred images provided by Zhao *et al.* in [35], which will serve for testing purposes. Table 3 shows the results obtained by Zhao *et al.* themselves, along with Ma *et al.*’s and the aforementioned fully supervised DeepLabv3 model. The reported values are all result of the direct application of the models learned over Shi *et al.*’s dataset. These are further compared with our weakly-supervised and self-supervised training methods, showing that the latter exhibits a significantly better generalization ability (0.950 vs. 0.923 of AUC for Ma *et al.*’s approach).

As a general conclusion, we show that synthetically blurring parts of images with a certain semantic coherence is,

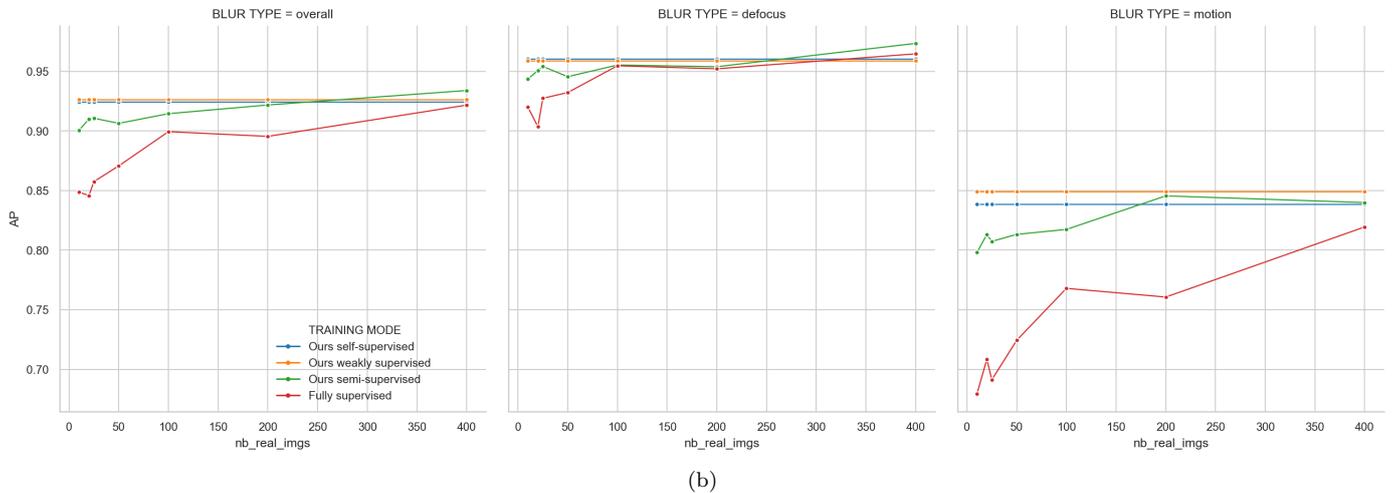
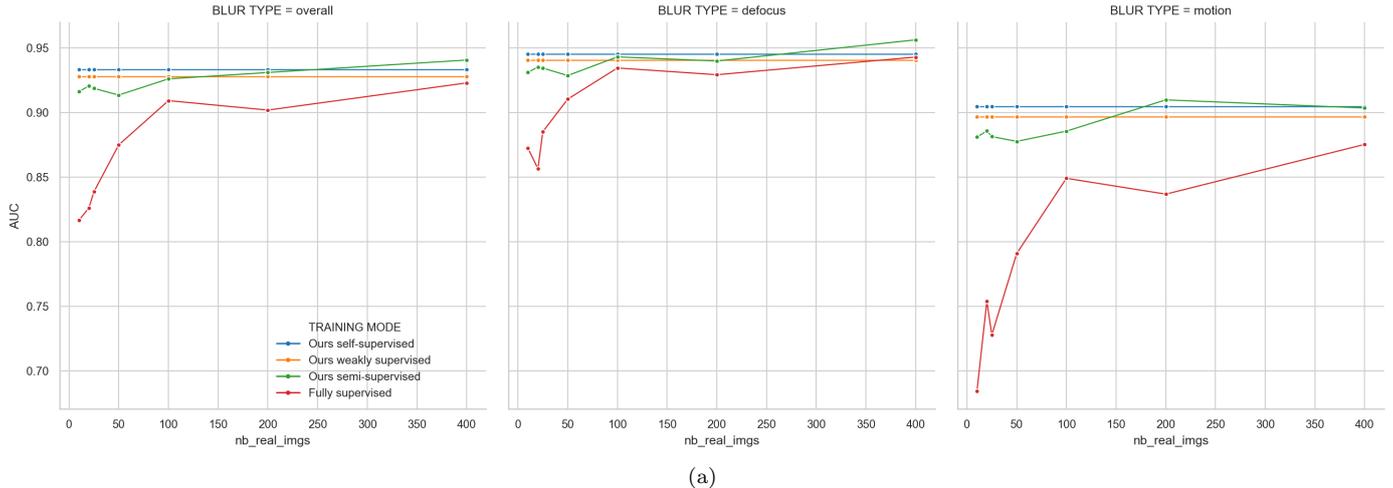


Figure 6: (a) AUC and (b) AP as a function of the number of images with real blur from the Shi *et al.*'s dataset [11] used in the training process, in the following setups: (i) Joint training on images with synthetic and real blur (*Ours semi-supervised.*) (ii) Direct fully-supervised fine-tuning on images from Shi *et al.* (*Fully supervised.*) The following setups are shown for comparative purposes, but do not use any image from Shi *et al.*: (iii) MCG object proposals-based self-supervised training (*Ours self-supervised.*) (iv) SegVOC12 semantic segmentation masks-based weakly supervised training (*Ours weakly supervised.*) The set of images with real blur ar part of Shi *et al.*'s odd subset, containing both defocus and motion blur. All the experiments were done using the DeepLabv3 architecture [28] with a Resnet101 backbone.

| Method | AUC | AP |
|-------------------------|--------------|--------------|
| Zhao <i>et al.</i> [35] | 0.913 | 0.946 |
| Ma <i>et al.</i> [31] | 0.923 | 0.956 |
| Ours self-supervised | 0.950 | 0.976 |
| Ours weakly supervised | 0.915 | 0.953 |
| Fully supervised | 0.904 | 0.952 |

Table 3: Direct testing of models from Table 1 on Zhao *et al.*'s defocus blur dataset [35], together with Zhao *et al.*'s [35] own results. None of the models in this table have seen Zhao *et al.*'s dataset during training. Bold is best.

on its own, a useful technique to perform self-supervised or weakly supervised blur localization, and that its use as aid when performing supervised end-to-end training (even in extreme few-shot cases) can help boost detection accuracy. Those domains where annotated data is scarce can particularly benefit from such approaches when applied in conjunction with other downstream computer vision tasks. Histological imaging (to focus a potential disease classification model on sharp parts of a digitized slide), multispectral imaging (where a significant, hardly avoidable blur effect is often found due to the chromatic aberrations derived from the large bandwidth of the captured spectra, and it is difficult to tell it apart from motion or defocus blur effects) or document scanning are some examples of such situations where a self-supervised approach could have a large impact.

5. Conclusions

This paper presents a framework for deep defocus and object motion blur segmentation built upon the procedural application of both types of synthetic blurring distortions over regions of images. The self-supervised and weakly supervised versions of the framework exploit different ways of obtaining the candidate blur masks for automatic ground truth generation, and can be applied without any blur localization annotation. In the semi-supervised case, this source of data augmentation can leverage the availability of a few labeled images to further improve the obtained segmentation accuracies. Extensive quantitative and qualitative experiments show that a segmentation CNN trained on this kind of synthetic data under any of the three mentioned framework configurations is able to accurately localize the blurred regions of a hold-out set, showing performances well above other recent CNN-based approaches.

6. Acknowledgements

This research was partially funded by the Basque Government's Industry Department under the ELKARTEK program's project ONKOIKER under agreement KK-2018/00090. We thank the Spanish project TIN2016-79717-R and mention Generalitat de Catalunya CERCA Program.

References

References

- [1] S. Zhang, X. Shen, Z. Lin, R. Mch, J. P. Costeira, J. M. F. Moura, Learning to Understand Image Blur, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6586–6595. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Learning_to_Understand_CVPR_2018_paper.html
- [2] S. Bae, F. Durand, Defocus Magnification, Computer Graphics Forum 26 (3) (2007) 571–579. doi:10.1111/j.1467-8659.2007.01080.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01080.x>
- [3] S. A. Golestaneh, L. J. Karam, Spatially-Varying Blur Detection Based on Multiscale Fused and Sorted Transform Coefficients of Gradient Magnitudes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5800–5809. URL http://openaccess.thecvf.com/content_cvpr_2017/html/Golestaneh_Spatially-Varying_Blur_Detection_CVPR_2017_paper.html
- [4] X. Zhang, R. Wang, X. Jiang, W. Wang, W. Gao, Spatially variant defocus blur map estimation and deblurring from a single image, Journal of Visual Communication and Image Representation 35 (2016) 257–264. doi:10.1016/j.jvcir.2016.01.002. URL <http://www.sciencedirect.com/science/article/pii/S1047320316000031>
- [5] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, M.-H. Yang, Soft-Segmentation Guided Object Motion Deblurring, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 459–468. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Pan_Soft-Segmentation_Guided_Object_CVPR_2016_paper.html
- [6] J. Shi, L. Xu, J. Jia, Just noticeable defocus blur detection and estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 657–665.
- [7] J. Gast, A. Sellent, S. Roth, Parametric Object Motion From Blur, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1846–1854. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Gast_Parametric_Object_Motion_CVPR_2016_paper.html
- [8] S. Gur, L. Wolf, Single Image Depth Estimation Trained via Depth From Defocus Cues, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7683–7692. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Gur_Single_Image_Depth_Estimation_Trained_via_Depth_From_Defocus_Cues_CVPR_2019_paper.html
- [9] X. Ding, Z. Chen, Improving Saliency Detection Based on Modeling Photographer's Intention, IEEE Transactions on Multimedia 21 (1) (2019) 124–134. doi:10.1109/TMM.2018.2851389.
- [10] S. Pertuz, M. A. Garcia, D. Puig, Focus-aided scene segmentation, Computer Vision and Image Understanding 133 (2015) 66–75. doi:10.1016/j.cviu.2014.09.009.
- [11] J. Shi, L. Xu, J. Jia, Discriminative Blur Detection Features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2965–2972. doi:10.1109/CVPR.2014.379.
- [12] C. Tang, C. Hou, Z. Song, Defocus map estimation from a single image via spectrum contrast, Optics Letters 38 (10) (2013) 1706–1708. doi:10.1364/OL.38.001706. URL <https://www.osapublishing.org/ol/abstract.cfm?uri=ol-38-10-1706>
- [13] P. Favaro, S. Soatto, M. Burger, S. J. Osher, Shape from Defocus via Diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (3) (2008) 518–531. doi:10.1109/TPAMI.2007.1175.

- [14] C. Zhou, S. Lin, S. Nayar, Coded aperture pairs for depth from defocus, in: Proceedings of the International Conference on Computer Vision, 2009, pp. 325–332. doi:10.1109/ICCV.2009.5459268.
- [15] C. Zhou, O. Cossairt, S. Nayar, Depth from Diffusion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1110–1117. doi:10.1109/CVPR.2010.5540090.
- [16] X. Zhu, S. Cohen, S. Schiller, P. Milanfar, Estimating Spatially Varying Defocus Blur From A Single Image, IEEE Transactions on Image Processing 22 (12) (2013) 4879–4891. doi:10.1109/TIP.2013.2279316.
- [17] D. J. Chen, H. T. Chen, L. W. Chang, Fast defocus map estimation, in: Proceedings of the IEEE International Conference on Image Processing, 2016, pp. 3962–3966. doi:10.1109/ICIP.2016.7533103.
- [18] X. Yi, M. Eramian, LBP-Based Segmentation of Defocus Blur, IEEE Transactions on Image Processing 25 (4) (2016) 1626–1638. doi:10.1109/TIP.2016.2528042.
- [19] Y. Pang, H. Zhu, X. Li, J. Pan, Motion blur detection with an indicator function for surveillance machines, IEEE Transactions on Industrial Electronics 63 (9) (2016) 5592–5601.
- [20] T. Zhu, L. J. Karam, Efficient perceptual-based spatially varying out-of-focus blur detection, in: Proceedings of the IEEE International Conference on Image Processing, 2016, pp. 2673–2677. doi:10.1109/ICIP.2016.7532844.
- [21] R. Liu, Z. Li, J. Jia, Image partial blur detection and classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587465.
- [22] A. Chakrabarti, T. Zickler, W. T. Freeman, Analyzing spatially-varying blur, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2512–2519. doi:10.1109/CVPR.2010.5539954.
- [23] B. Su, S. Lu, C. L. Tan, Blurred Image Region Detection and Classification, in: Proceedings of the ACM International Conference on Multimedia, MM '11, ACM, New York, NY, USA, 2011, pp. 1397–1400. doi:10.1145/2072298.2072024. URL <http://doi.acm.org/10.1145/2072298.2072024>
- [24] S. Zhuo, T. Sim, Defocus map estimation from a single image, Pattern Recognition 44 (9) (2011) 1852–1858. doi:10.1016/j.patcog.2011.03.009. URL <http://www.sciencedirect.com/science/article/pii/S003132031100094X>
- [25] Y. Pang, H. Zhu, X. Li, X. Li, Classifying Discriminative Features for Blur Detection, IEEE Transactions on Cybernetics 46 (10) (2016) 2220–2227. doi:10.1109/TCYB.2015.2472478.
- [26] C. Tang, J. Wu, Y. Hou, P. Wang, W. Li, A Spectral and Spatial Approach of Coarse-to-Fine Blurred Image Region Detection, IEEE Signal Processing Letters 23 (11) (2016) 1652–1656. doi:10.1109/LSP.2016.2611608.
- [27] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2016) 1–1. doi:10.1109/TPAMI.2016.2572683.
- [28] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv:1706.05587 [cs]. URL <http://arxiv.org/abs/1706.05587>
- [29] K. He, G. Gkioxari, P. Dollr, R. Girshick, Mask R-CNN, in: Proceedings of the International Conference on Computer Vision, Venice, Italy, 2017. URL http://openaccess.thecvf.com/ICCV2017_search.py
- [30] X. Liu, J. van de Weijer, A. D. Bagdanov, Leveraging Unlabeled Data for Crowd Counting by Learning to Rank, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018. URL <http://arxiv.org/abs/1803.03095>
- [31] K. Ma, H. Fu, T. Liu, Z. Wang, D. Tao, Deep Blur Mapping: Exploiting High-Level Semantics by Deep Neural Networks, IEEE Transactions on Image Processing 27 (10) (2018) 5155–5166. doi:10.1109/TIP.2018.2847421.
- [32] B. Kim, H. Son, S.-J. Park, S. Cho, S. Lee, Defocus and Motion Blur Detection with Deep Contextual Features, Computer Graphics Forum 37 (7) (2018) 277–288. doi:10.1111/cgf.13567. URL <https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.13567>
- [33] J. Park, Y.-W. Tai, D. Cho, I. So Kweon, A Unified Approach of Multi-Scale Deep and Hand-Crafted Features for Defocus Estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1736–1745. URL http://openaccess.thecvf.com/content_cvpr_2017/html/Park_A_Unified_Approach_CVPR_2017_paper.html
- [34] K. Zeng, Y. Wang, J. Mao, J. Liu, W. Peng, N. Chen, A Local Metric for Defocus Blur Detection Based on CNN Feature Learning, IEEE Transactions on Image Processing 28 (5) (2019) 2107–2115. doi:10.1109/TIP.2018.2881830.
- [35] W. Zhao, F. Zhao, D. Wang, H. Lu, Defocus Blur Detection via Multi-Stream Bottom-Top-Bottom Fully Convolutional Network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3080–3088. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhao_Defocus_Blur_Detection_CVPR_2018_paper.html
- [36] W. Zhao, F. Zhao, D. Wang, H. Lu, Defocus Blur Detection via Multi-Stream Bottom-Top-Bottom Network, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1–1doi:10.1109/TPAMI.2019.2906588.
- [37] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, arXiv:1609.04802 [cs]. URL <http://arxiv.org/abs/1609.04802>
- [38] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, Context Encoders: Feature Learning by Inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [39] X. Liu, J. van de Weijer, A. D. Bagdanov, RankIQA: Learning From Rankings for No-Reference Image Quality Assessment, in: Proceedings of the International Conference on Computer Vision, 2017.
- [40] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The Unreasonable Effectiveness of Deep Networks as a Perceptual Metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. URL <https://richzhang.github.io/PerceptualSimilarity/>
- [41] D. Novotny, S. Albanie, D. Larlus, A. Vedaldi, Self-Supervised Learning of Geometrically Stable Features Through Probabilistic Introspection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3637–3645. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Novotny_Self-Supervised_Learning_of_CVPR_2018_paper.html
- [42] S. Jenni, P. Favaro, Self-Supervised Feature Learning by Learning to Spot Artifacts, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2733–2742. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Jenni_Self-Supervised_Feature_Learning_CVPR_2018_paper.html
- [43] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8183–8192. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Kupyn_DeblurGAN_Blind_Motion_CVPR_2018_paper.html
- [44] K. Purohit, A. B. Shah, A. N. Rajagopalan, Learning Based Single Image Blur Detection and Segmentation, in: Proceedings of the IEEE International Conference on Image Processing, 2018, pp. 2202–2206. doi:10.1109/ICIP.2018.8451765.

- [45] P. Krahenbuhl, V. Koltun, Learning to Propose Objects, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1574–1582.
URL https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Krahenbuhl_Learning_to_Propose_2015_CVPR_paper.html
- [46] S. Chen, H. Shen, Multispectral Image Out-of-Focus Deblurring Using Interchannel Correlation, IEEE Transactions on Image Processing 24 (11) (2015) 4433–4445. doi:10.1109/TIP.2015.2465162.
- [47] X. M. Lopez, E. D’Andrea, P. Barbot, A.-S. Bridoux, S. Rorive, I. Salmon, O. Debeir, C. Decaestecker, An Automated Blur Detection Method for Histological Whole Slide Imaging, PLOS ONE 8 (12) (2013) e82710. doi:10.1371/journal.pone.0082710.
URL <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0082710>
- [48] S. Maheshwari, P. K. Rai, G. Sharma, V. Gandhi, Document Blur Detection Using Edge Profile Mining, in: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP ’16, ACM, New York, NY, USA, 2016, pp. 23:1–23:7. doi:10.1145/3009977.3009982.
URL <http://doi.acm.org/10.1145/3009977.3009982>
- [49] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes Challenge: A Retrospective, International Journal of Computer Vision 111 (1) (2015) 98–136. doi:10.1007/s11263-014-0733-5.
URL <https://link.springer.com/article/10.1007/s11263-014-0733-5>
- [50] J. Pont-Tuset, P. Arbelez, J. T. Barron, F. Marques, J. Malik, Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (1) (2017) 128–140. doi:10.1109/TPAMI.2016.2537320.
- [51] A. Telea, An Image Inpainting Technique Based on the Fast Marching Method, Journal of Graphics Tools 9 (1) (2004) 23–34. doi:10.1080/10867651.2004.10487596.
URL <https://doi.org/10.1080/10867651.2004.10487596>
- [52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, in: International Conference on Learning Representations, 2015, arXiv: 1412.7062.
URL <http://arxiv.org/abs/1412.7062>
- [53] F. Yu, V. Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, in: International Conference on Learning Representations, 2016.
URL <http://arxiv.org/abs/1511.07122>
- [54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, Springer, Cham, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.
URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48
- [55] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to Adapt Structured Output Space for Semantic Segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7472–7481. doi:10.1109/CVPR.2018.00780.
URL http://openaccess.thecvf.com/content_cvpr_2018/html/Tsai_Learning_to_Adapt_CVPR_2018_paper.html