Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

Defocus and Motion Blur Detection with Deep Contextual Features

Beomseok Kim[1]*, Hyeongseok Son[1]*, Seong-Jin Park[1], Sunghyun Cho[2], and Seungyong Lee[1]

[1]POSTECH [2]DGIST

Abstract

We propose a novel approach for detecting two kinds of partial blur, defocus and motion blur, by training a deep convolutional neural network. Existing blur detection methods concentrate on designing low-level features, but those features have difficulty in detecting blur in homogeneous regions without enough textures or edges. To handle such regions, we propose a deep encoder-decoder network with long residual skip-connections and multi-scale reconstruction loss functions to exploit high-level contextual features as well as low-level structural features. Another difficulty in partial blur detection is that there are no available datasets with images having both defocus and motion blur together, as most existing approaches concentrate only on either defocus or motion blur. To resolve this issue, we construct a synthetic dataset that consists of complex scenes with both types of blur. Experimental results show that our approach effectively detects and classifies blur, outperforming other state-of-the-art methods. Our method can be used for various applications, such as photo editing, blur magnification, and deblurring.

CCS Concepts
•***Computing methodologies*** → ***Image processing;***

1. Introduction

Partial blur of photographs is caused by various reasons such as shallow depth-of-field, camera motions, and moving objects. Unintentional blur generally degrades image quality, and many existing methods including defocus map estimation [ZS11, SXJ15, PTCK17] and motion blur kernel estimation [HJSHS11, WSZP12, SCX*17] concentrate on estimating blur kernels to remove such blur. However, accurate blur kernel estimation is a severely ill-posed problem and existing methods usually handle only one type of blur. Meanwhile, blur detection and classification, i.e., localizing partial blur and identifying its type, are also important tasks for various applications, such as image editing [BD07, ARG18], image quality assessment [SBC12], image restoration [EGA*13], saliency detection [JLYP13], and video deblurring [CWL12, LJLS13, ZZH17]. Image deblurring can also benefit from blur detection and classification. While most existing deblurring approaches [HJSHS11, WSZP12] cannot manage different types of blur in one image, simple combination of blur detection and classification with an existing deblurring method could handle such a case (Sec. 6.1).

Most blur detection approaches [LLJ08, CZF10, SLT11, SXJ14, TWH*16, GK17] designed hand-crafted features to distinguish blurred and non-blurred image regions. Even though such features are usually discriminative around edges, it is hard to use them for determining the existence of blur in homogeneous regions without enough texture or edges (Fig. 1). Some defocus blur map estimation methods [ZS11, SXJ15, PTCK17] estimate the amount of blur around edges, and propagate the estimates to homogeneous regions. However, this propagation based approach still has difficulty in filling large homogeneous regions with accurate blur estimation.

Deep convolutional neural networks (DCNN) have led remarkable successes in various image processing tasks, e.g., semantic segmentation [LMSR17, NHH15] and saliency detection [LY15, HCH*17]. A few deep learning based approaches have been proposed for blur detection and estimation [HFF*18, PTCK17]. These approaches detect blur from small patches using CNN features, and reconstruct a full blur map by merging or interpolating the patch responses. These CNN-based methods outperform previous blur detection and estimation methods, but CNN features with small receptive fields have limitation on detecting blur in homogeneous regions. Recently, Ma *et al.* [MFL*18] proposed to utilize high-level features to deal with homogeneous regions. However, their method does not consider any low-level features, so it produces less accurate and blurry results.

In this paper, we propose an effective deep encoder-decoder network with long residual skip-connections to detect and classify different types of partial blur, i.e., defocus and motion blur. Differently from existing methods using either low-level features [HFF*18, PTCK17] or high-level features [MFL*18], our network considers both low-level structural features and high-level contextual features to accurately detect blur around structural edges as well as in ho-

*Both authors have equal contiribution on this paper.

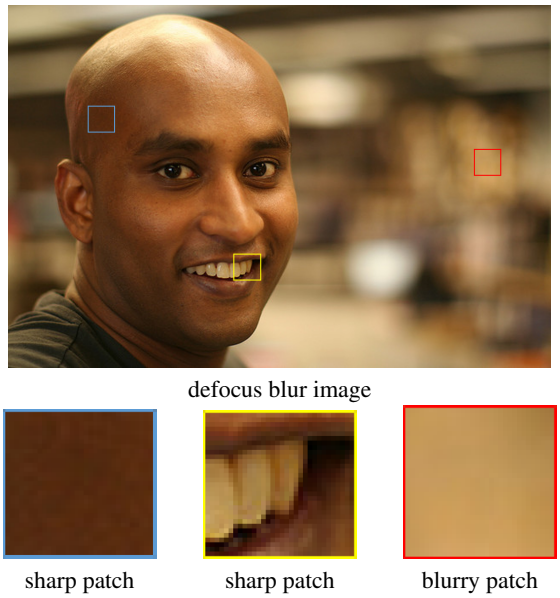


Figure 1: *Example of a challenging case in blur detection. The blue and yellow boxes correspond to sharp regions, whereas the red one corresponds to a defocus-blur region. While the blue box corresponds to a sharp region, it is hard to determine the region is blurred or not at the patch level because it has no edges or textures.*

mogeneous regions. To effectively train our network while exploiting both high- and low-level features together, we use multi-scale reconstruction loss functions.

To properly train our deep network, it is essential to have a good dataset. Unfortunately, there are no such datasets that suit our purpose. To the best of our knowledge, CUHK blur detection dataset [SXJ14] is the only dataset publicly available. However, it contains only 1,000 labeled images. Furthermore, all the images in CUHK dataset have only one type of blur and no images have defocus and motion blur together. To resolve this issue, we generate a synthetic dataset that contains images with both motion and defocus blur together so that our network can be trained to discriminate different types of blur. We employ our dataset for fine-tuning our network after training with CUHK dataset. To prevent overfitting and facilitate training, we also utilize a pre-tained model, VGG-19 [SZ14], to initialize our encoder network, which was trained for image recognition with a large dataset. It provides well-trained low-level features and helps our network effectively exploit high-level semantic information.

Experimental results show that our approach is effective and outperforms the state-of-the-art methods for blur detection. We also demonstrate possible applications of our method: image editing, blur magnification, and deblurring.

The main contributions of our work are summarized as follows:

- We propose an effective deep encoder-decoder network with long residual skip-connections to detect two kinds of blur: defocus and motion blur. Our multi-scale reconstruction loss functions help our network reconstruct precise blur maps by fully utilizing both high- and low-level features.
- We construct a synthetic dataset that consists of complex scenes including both defocus and motion blur together. Fine-tuning with the dataset makes our network learn discriminative features between defocus and motion blur.

2. Related Work

Blur detection and classification Many effective blur detection methods based on hand-crafted features have been proposed. Liu *et al.* [LLJ08] detect partial blur using four local blur features and classify two kinds of blur. Chakrabarti *et al.* [CZF10] analyze directional blur via local Fourier transform. Su *et al.* [SLT11] propose a method that employs singular value information to measure blurriness. The method also classifies blur into different types using alpha channel constraints. Shi *et al.* [SXJ14] propose a set of blur features in multiple domains such as gradient histogram span, kurtosis, and data-driven local filters. They also provide a real blur detection dataset containing 1,000 images labeled with pixel-level annotation. Tang *et al.* [TWH*16] introduce image spectrum residual to estimate a coarse blur map, and an iterative update algorithm to refine the blur map by filling homogeneous regions. Golestaneh and Karam [GK17] detect spatially-varying blur by applying multi-scale fusion to high frequency Discrete Cosine Transform (DCT) coefficients. Although all these methods can effectively detect blur around edges, blur in homogeneous regions are not properly analyzed as their hand-crafted features do not properly handle homogeneous regions.

Defocus map estimation Defocus map estimation methods compute the amounts of spatially-varying defocus blur for a given image. They are closely related to blur detection in that partial blur is detected and a blur map is generated. Zhuo and Sim [ZS11] obtain a sparse defocus map from gradient ratios between input and re-blurred image patches. A sparse defocus map is then interpolated to produce a full defocus map. Shi *et al.* [SXJ14] develop a sparse representation for detecting just noticeable blur and estimating its strength. However, as these methods also estimate a sparse defocus map from edges and propagate blur information to homogeneous regions, blur in large homogeneous regions is often inaccurately estimated.

Deep learning based methods Several deep learning based methods have been introduced to understand image blur. Huang *et al.* [HFF*18] detect local blur regardless of its type using a CNN that consists of six layers including fully connected (fc) layers. They showed that data-driven CNN features of multi-scale patches are more discriminative than previous hand-crafted features. However, their method suffers from inefficiencies of fc layers and processing many small patches, and has difficulty in distinguishing blur in homogeneous regions due to narrow receptive fields. Besides, their results are not precisely aligned to original image structures (e.g., object boundaries) as blur is detected in a patch-wise manner. Park *et al.* [PTCK17] present a state-of-the-art defocus map estimation method that uses both hand-crafted and deep CNN features. Their method has two stages. The first stage estimates a sparse defocus map from edges using their proposed features in small patches. The second stage propagates local blur information around edges to nearby smooth regions using matting Lapla-

input ground truth our result

Figure 2: *Examples of the ground truth labels in CUHK dataset and our blur detection results. In a ground truth map, white is blurred and black is non-blurred regardless of blur types. In our result, blue is motion blur, red is defocus blur, and black is no-blur.*

blurred image ground truth blur map

Figure 3: *Our synthetic dataset. Each image in CUHK dataset [SXJ14] has either motion blur or defocus blur, but not both. For handling images with both types of blur, we synthetically generate images with both types of blur (left) and their ground truth blur maps (right) utilizing salient object detection dataset and CUHK dataset. In ground truth blur maps on the right, black, blue and red indicate no-blur, motion blur, and defocus blur, respectively.*

cian [LLW08]. However, their approach often estimates inaccurate blur in large homogeneous regions, and does not consider motion blur. Recently, Ma *et al.* [MFL*18] presented a fully convolutional network for blur detection, which is trained in an end-to-end manner. In their approach, a low-resolution blur map is first obtained using high-level features, and then the final blur map is reconstructed by bilinear upsampling. However, their approach does not consider low-level structural features either in low-resolution blur map estimation or upsampling, so their blur maps are blurry and less accurate near structural edges.

Our approach is different from previous deep learning based methods in two aspects. First, we utilize a well-designed encoder-decoder network to exploit both high- and low-level features together. Our network effectively detects blur in large homogeneous regions with large receptive fields, while preserving structural information such as edges by exploiting low-level features. As a result, our method can produce significantly improved results both quantitatively and qualitatively compared to previous methods. Second, our network can distinguish defocus and motion blur as well as localizing them. For training our network to learn discriminative features between both blur types, we construct a new synthetic dataset that consists of complex scenes including defocus and motion blur together.

3. Dataset for Blur Detection and Classification

3.1. Blur detection dataset

CUHK dataset [SXJ14], which is the only publicly available blur detection dataset, consists of 704 defocus and 296 motion blur images. The images are equipped with ground truth blur maps labeled by human annotators, each pixel of which indicates whether the pixel is blurred or not. While the dataset has both defocus and motion blur images, each image is assumed to be blurred by only one type of blur, and the blur type for each image is given as an additional tag. Fig. 2 shows images of CUHK dataset. The left and middle columns show blurred images and their corresponding ground truth blur maps, respectively.

To train our network for distinguishing defocus and motion blur, we need separate labels for different blur types. We convert the ground truth pixel labels in the blur maps of CUHK dataset by reflecting the blur types of images. Then, using the converted blur maps together with our synthetic dataset, which will be discussed next, we train our network to classify two kinds of blur at each pixel as well as to detect them. The right column in Fig. 2 shows results of our network, where blue and red denote motion and defocus blur, respectively.

3.2. Synthetic dataset for blur classification

Most images in CUHK dataset do not have motion and defocus blur together. While some images in the dataset actually have both types of blur, they are still annotated as if they have only one type of blur. Consequently, training with only CUHK dataset makes a network less effective in distinguishing different types of blur in an image, as will be shown in Sec. 5.4. To resolve this limitation, we construct a new synthetic dataset whose images have both types of blur together with pixel-wise annotations (Fig. 3).

We generate our synthetic dataset using CUHK dataset and a salient object detection dataset [LY15]. The salient object detection dataset consists of 4,000 images and their corresponding binary masks indicating salient objects. Algorithm 1 summarizes our process of generating the synthetic dataset. Object motion blur is usually modeled by locally linear blur kernels [CDSAP13, KL14] as it is usually caused by fast moving objects like cars. Therefore, we use linear blur kernels to generate motion blur in an image. We finally obtain 8,460 images that have both defocus and motion blur in our synthetic dataset.

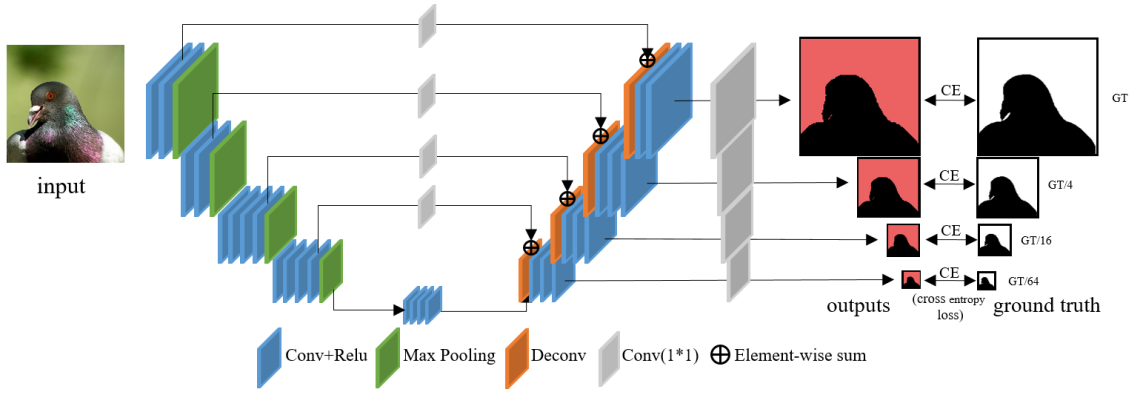


Figure 4: *Our blur detection and classification network with multi-scale reconstruction loss functions.*

Algorithm 1 Synthetic blur dataset generation

$B \leftarrow$ randomly sampled 564 images among the defocus blur images in CUHK dataset
for each image b in B **do**
  $F \leftarrow$ randomly sampled 15 images among the salient object detection dataset.
  for each image f in F **do**
    $m \leftarrow$ salient object mask of f
    $s \leftarrow$ salient object extracted from f using m
    blur s and m using a random linear motion blur kernel
    blend b and the blurred s using alpha-blending with the blurred m to impose the motion-blurred salient object onto b
  end for
end for

4. Blur Detection and Classification using DCNN

4.1. Network architecture

Our network takes a 3-channel RGB image as input, and outputs a 3-channel blur map, each channel of which corresponds to motion blur, defocus blur, or no-blur. Our network is built upon an encoder-decoder framework with long skip-connections, similarly to U-Net architecture [RFB15] used for various image processing problems [SDW*17, ZYW*17, HGO18]. Fig. 4 shows the detailed structure of our network.

The encoder network consists of four max-pooling and 16 convolution layers, similarly to VGG-19 [SZ14] but without fully connected layers. Thanks to max-pooling and convolution layers of the encoder, our network has much larger receptive fields than previous deep learning based approaches [PTCK17, HFF*18]. As a result, our network can utilize higher level features with larger contextual information. We design our network as fully convolutional for efficient computation and for handling images of arbitrary sizes, in contrast to previous blur detection methods that use small patches sampled from the input image. The decoder network consists of four deconvolution and 15 convolution layers to upsample features and generate an output of the same size as the input. Batch-normalization is not used for our network as we found that it decreased the accuracy in our experiments.

We add four long symmetric skip-connections with a 1×1 convolution layer in the middle. The long skip-connections pass features at each scale of the encoder directly to the decoder layer at the same scale, enabling reconstruction of a high-resolution blur map. Recent CNN structures often adopt element-wise summation [HZRS16] and concatenation [SLJ*15] to fuse multiple features. In contrast to U-Net [RFB15], we adopt element-wise summation, instead of concatenation, when forwarding features using skip-connections. In our network, features in the encoder include structural information extracted from the finest scale, and features in the decoder include contextual information dilated from the coarsest level. With concatenation, the network may merely use the structural features for reconstructing the final blur map. We use element-wise summation to forcibly combine both contextual features and structural features passed by skip-connections, strengthening the influence of contextual features to better deal with homogeneous regions.

4.2. Pre-trained model as encoder

Even with our synthetic dataset as well as CUHK datset, the training set is still not big enough. Directly training a complex network with a limited dataset inevitably incurs over-fitting. To resolve the problem, we use a pre-trained model, VGG-19 [SZ14] trained with ImageNet dataset [RDS*15] for image classification. Although image classification and blur detection are different, it is known that various vision tasks such as image recognition and restoration share low-level features such as edges and small textures. In addition, high-level features trained for image classification would be helpful for semantically determining blurriness in homogeneous regions. Therefore, we fine-tune a well-trained model for our encoder network so that the encoder can learn discriminative features effectively based on the features trained for image classification. This fine-tuning also enables our decoder network to learn how to propagate well-trained features even only with a limited dataset. The effectiveness of this approach is shown in Sec. 5.3.

4.3. Multi-scale reconstruction loss functions

As our network classifies defocus blur, motion blur, and no-blur at each pixel, we use cross-entropy losses to train our network. For effective learning of high-level contextual features, we adopt multi-scale supervision [SVI*16] that helps stable training of a network (Fig. 4). We express our network f as:

$$\hat{\mathbf{b}}_s = f_s(\mathbf{I};\theta), \tag{1}$$

where s is a scale index, $\hat{\mathbf{b}}_s$ is a blur map with three channels at scale s, f_s is the output of our network f at scale s, $\mathbf{I}$ is an input RGB image, and θ is a set of network parameters. $\hat{\mathbf{b}}_s$ at the finest scale has the same spatial resolution as the input image $\mathbf{I}$, and the three channels of a blur map correspond to different blur types. Then, training is done by minimizing the loss function defined as:

$$L(\theta;\mathbf{I},\mathbf{b}) = -\sum_s \sum_p \sum_c w_s \mathbf{b}_{s,p,c} \log(\hat{\mathbf{b}}_{s,p,c}), \tag{2}$$

where $\mathbf{b}$ is the ground truth blur map, and $\mathbf{b}_s$ is its downsampled version at scale s. p and c are pixel and channel indices, respectively. $\mathbf{b}_{s,p,c}$ and $\hat{\mathbf{b}}_{s,p,c}$ are values of $\mathbf{b}_s$ and $\hat{\mathbf{b}}_s$ at pixel p and channel c, respectively. $\hat{\mathbf{b}}_{s,p,c}$ is a normalized value across channels using softmax. w_s is a weight for each scale s. We set w_s inversely proportional to the number of pixels at scale s.

The cross-entropy loss at the finest scale helps detailed reconstruction of a blur map, whereas the one at the coarsest scale helps coarse approximation. In other words, the loss at the coarsest scale guides our network to reconstruct an output map using only high-level features, enabling contextual information propagated over a wide range. As a result, blur information around edges can be propagated to smooth homogeneous regions far from the edges. The other losses at finer scales consider mid- and low-level features so that mid- and high-frequency information can be accurately reconstructed.

4.4. Implementation details

Data augmentation To prevent the over-fitting problem during training, we randomly cropped images into 256×256 patches. Then, we flipped the image patches horizontally and rotated them by 90°, 180°, and 270° for data augmentation.

Training setting We implemented our network with Tensorflow and used an NVIDIA Titan XP GPU to train the network. We used Adam optimizer [KB14] for training. The encoder is initialized with pre-trained VGG-19 weights as explained before. The decoder is initialized by Xavier initialization [GB10]. Our training process consists of two stages. In the first stage, we use only CUHK dataset. We used only 800 images from the dataset, keeping other 200 for quantitative evaluation. The learning rate was initially set to 10^{-4} for the decoder and 10^{-5} for the encoder, and the network was trained for $3,000$ epochs. Then, the learning rate was set to 10^{-5} for the decoder, and 10^{-6} for the encoder, and another $3,000$ epochs were run. In the second stage, we use our synthetic dataset to further improve the blur detection and classification performance. We set the learning rate to 10^{-5} for the decoder and 10^{-6} for the encoder, and run about $4,000$ epochs. We used batch size 10 for all training stages. We use this model as our final model for our evaluation unless otherwise noted.

5. Experimental Results

For all quantitative evaluations, we use 200 images from CUHK dataset that are not used for training. Among 200 images, 140 images are defocus blurred, and the other 60 are motion blurred.

We first evaluate the performance of our network as a blur detector that distinguishes between blurred pixels and non-blurred pixels for comparison with previous methods. As our network produces three labels for each pixel, we obtain a binary blur map by merging motion and defocus blur in the output of our network.

We compare our method with previous blur detection methods [SXJ14, TWH*16, GK17, HFF*18, MFL*18] in Sec. 5.1, and defocus map estimation methods [SXJ15, PTCK17] in Sec. 5.2. All these methods output a map of continuous values either representing confidence of blurriness or the size of defocus blur. To compare the performance of blur detection, we convert their results into binary maps by thresholding them. For fair comparison, we found the best threshold for each method. Specifically, we tried 255 uniformly sampled thresholds to the results of previous methods on the test set of 200 images. We then chose the best one so that we can compare our results against their best results. As we use three metrics for our quantitative evaluation as discussed later, we chose three thresholds for each method, each of which produces the best result for the corresponding metric. Resulting binary blur maps, including ours, have either 0 or 1 at each pixel representing no-blur or blur, respectively.

The three metrics used for evaluating our binary blur maps are accuracy, mean intersection of union (mIoU), and F-measure, which are defined as:

$$Accuracy = \frac{1}{w \times h} \sum_p (1 - |B_p - G_p|), \tag{3}$$

$$mIoU = \frac{1}{N_c} \sum_c \left(\frac{B^c \cap G^c}{B^c \cup G^c}\right), \quad \text{and} \tag{4}$$

$$F_\beta\text{-}measure = \frac{(1+\beta)\, precision \times recall}{\beta^2\, precision + recall} \tag{5}$$

where B is an estimated blur map, G is the ground truth map, and $w \times h$ is the image size. c is a blur type and N_c is the number of blur types, which is two in our case. Precision and recall are defined as:

$$precision = \frac{\sum_p B_p G_p}{\sum_p B_p}, \quad \text{and} \quad recall = \frac{\sum_p B_p G_p}{\sum_p G_p}. \tag{6}$$

β is a parameter for F-measure. We use $\beta = 1$, which means the same importance of precision and recall. In addition to these metrics, we also compute the variance (σ^2) of the accuracy for the entire set of test images to measure the stability of each method.

5.1. Comparison with other blur detection methods

We first compare our method with other state-of-the-art blur detection methods: Shi *et al.* [SXJ14], Tang *et al.* [TWH*16], Golestaneh and Karam [GK17], Huang *et al.* [HFF*18], and Ma *et al.* [MFL*18]. Results of other methods were generated by the authors' implementations or downloaded from their project pages. Huang *et al.*'s method was also trained using CUHK dataset, and their test set is the same as ours. There is no publicly available

	Shi [SXJ14]	Tang [TWH*16]	Golestaneh [GK17]	Huang [HFF*18]	Ma [MFL*18]	Ours
Max accuracy	0.7423	0.7701	0.8010	0.7949	0.8827	**0.9033**
σ^2 of accuracy	0.0220	0.0322	0.0227	0.0221	**0.0114**	0.0139
Max F-measure	0.8055	0.8295	0.8417	0.8485	0.8883	**0.9092**
Max mIoU	0.5617	0.6040	0.6349	0.6318	0.7578	**0.8047**

Table 1: *Quantitative comparison of blur detection (blur/no-blur) performance with other blur detection methods on CUHK test set.*

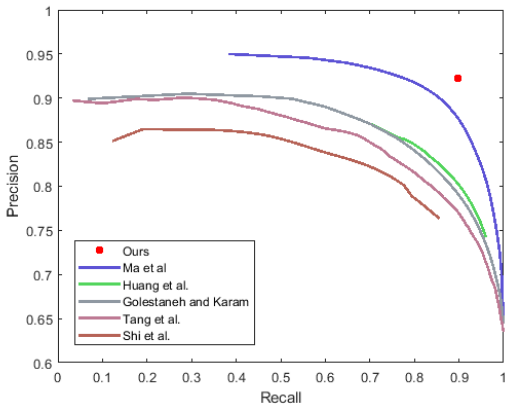


Figure 5: *Precision-recall graphs for the blur detection (blur/no-blur) performance in Table 1. Note that our method has a fixed precision-recall value as our network returns 3-way classification results and we convert them into blur/no-blur maps by merging the motion and defocus blur without any thresholding. Our method achieves higher precision and recall than all the other methods.*

source code of Ma *et al.*'s method [MFL*18], so we implemented the method and trained it with the same dataset as ours. Table 1 shows a quantitative comparison. As mentioned earlier, we tried 255 different thresholds and used the best one for the results of the other methods. Nonetheless, our method clearly outperforms all the others by a large margin in terms of accuracy, F-measure, and mIoU. Overall precision-recall curves are shown in Fig. 5. Our method also shows a low variance of the accuracy, implying high stability in handling various input images.

Fig. 6 shows a qualitative comparison. Most of the other methods show errors in homogeneous regions as they do not properly handle such regions. The method of Ma *et al.* [MFL*18] shows high performance quantitatively, but produces blurry and less accurate results near edges as they do not consider low-level features. On the other hand, our results are visually close to the ground truth maps, and contain less noise and stains. Edges in our results are accurately aligned to those in the ground truth maps. Moreover, blur in homogeneous regions is accurately detected as well, as shown in the fifth and sixth rows in Fig. 6. Interestingly, some of our results show more accurate edges than ground truth blur maps annotated by humans, as shown in the second row in Fig. 6.

	Shi [SXJ15]	Park [PTCK17]	Ours
Max accuracy	0.7934	0.8284	**0.9147**
σ^2 of accuracy	0.0191	0.0206	**0.0090**
Max F-measure	0.8393	0.8770	**0.9354**
Max mIoU	0.6181	0.6785	**0.8248**

Table 2: *Quantitative comparison of defocus blur detection performance with other defocus map estimation methods on CUHK defocus test set. The results of other methods were converted into binary blur maps.*

	baseline	multi loss	multi loss + VGG	final
Accuracy	0.8875	0.8955	0.9028	**0.9033**
F-measure	0.8814	0.8909	0.9091	**0.9092**
mIoU	0.7677	0.7829	0.8032	**0.8047**

Table 3: *Ablation study for blur detection (blur/no-blur) performance. 'baseline' uses single-scale loss and is learned from scratch. 'multi loss' uses multi-scale loss and is learned from scratch. 'multi loss + VGG' uses multi-scale loss and is initialized by VGG-19. These three models are trained using CUHK dataset. 'final' uses multi-scale loss and initialization using VGG-19, and is also fine-tuned using both CUHK and our synthetic datasets.*

5.2. Comparison with other defocus map estimation methods

We also compare our method with the state-of-the-art defocus map estimation methods [SXJ15, PTCK17], as they can also be used for blur detection. To compare our method and defocus map estimation methods, we use the same thresholding strategy. We tried 255 different threshold values and chose the best one for each defocus map estimation method. Table 2 shows a quantitative comparison. Our method again clearly outperforms the others in terms of all metrics.

Fig. 7 shows a qualitative comparison. We select various defocus images including challenging cases for visual comparison. Previous defocus estimation methods fail in homogeneous regions such as human faces because human faces have too weak edges. While the results of Park et al. [PTCK17] in the third and fourth rows show that the defocus blur amount of the swimmer and kid is relatively lower than that of the background, they are still detected to be blurry, causing thresholding to fail. On the other hand, our method successfully detected blur in homogeneous regions with much less error around strong edges because our method considers large contextual information.

Figure 6: *Our blur detection results compared with other blur detection methods. White/black colors in the blur map denotes blur/no-blur regions, respectively.*

5.3. Ablation study

We perform ablation study to see how each component of our method affects the performance of blur detection. We compare four different models that are trained using different strategies, but use the same network architecture. The quantitative results are shown in Table 3. The first three models are trained using only CUHK dataset. The first model (baseline) is trained from scratch using a single-scale loss instead of the multi-scale loss described in Sec. 4.3. The second one (multi loss) is also trained from scratch using the multi-scale loss. The third one (multi loss + VGG) is trained from pre-trained VGG-19 using the multi-scale loss. The final one is our final model, which uses the multi-scale loss, and pre-trained VGG-19 parameters, and is trained using both CUHK and our synthetic datasets. The results show that each component effectively increases the performance and the final model exceeds 0.9 in terms of both accuracy and F-measure.

Another thing worth mentioning in Table 3 is that the performance gap between our third and final models does not look significant. This is because our synthetic dataset, which is used for training our final model, is designed for blur type classification of different types of blur, while the quantitative evaluation in Table 3 simply considers blur detection (blur/no-blur) performance. Regarding blur detection, our model trained using only CUHK dataset (our third model) could already achieve high accuracy as blur detection is simpler than blur type classification. The gain of using our synthetic dataset will be shown in Sec. 5.4.

	only CUHK	mixture of CUHK and ours
Accuracy	0.8404	0.9075
mIoU	0.6576	0.8073

Table 4: *Quantitative comparison of blur type classification performance between our models trained with and without our synthetic dataset that consists of complex scenes including both types of blur together.*

Fig. 8 shows a qualitative comparison between our different models. As shown in Fig. 8, the baseline model has many errors in homogeneous regions such as the sky and the paper. The result of the second model that uses the multi-scale loss shows much less error than the baseline, although it still has quite amount of error. This implies that the multi-scale loss plays an important role for achieving high-quality results, as it encourages the network to more effectively use large contextual information captured by coarse-level features. Finally, the results of our third (multi loss+VGG) and final models have only a very small amount of error. This shows that well-trained features of pre-trained VGG-19 are very effective to detect and propagate blurriness.

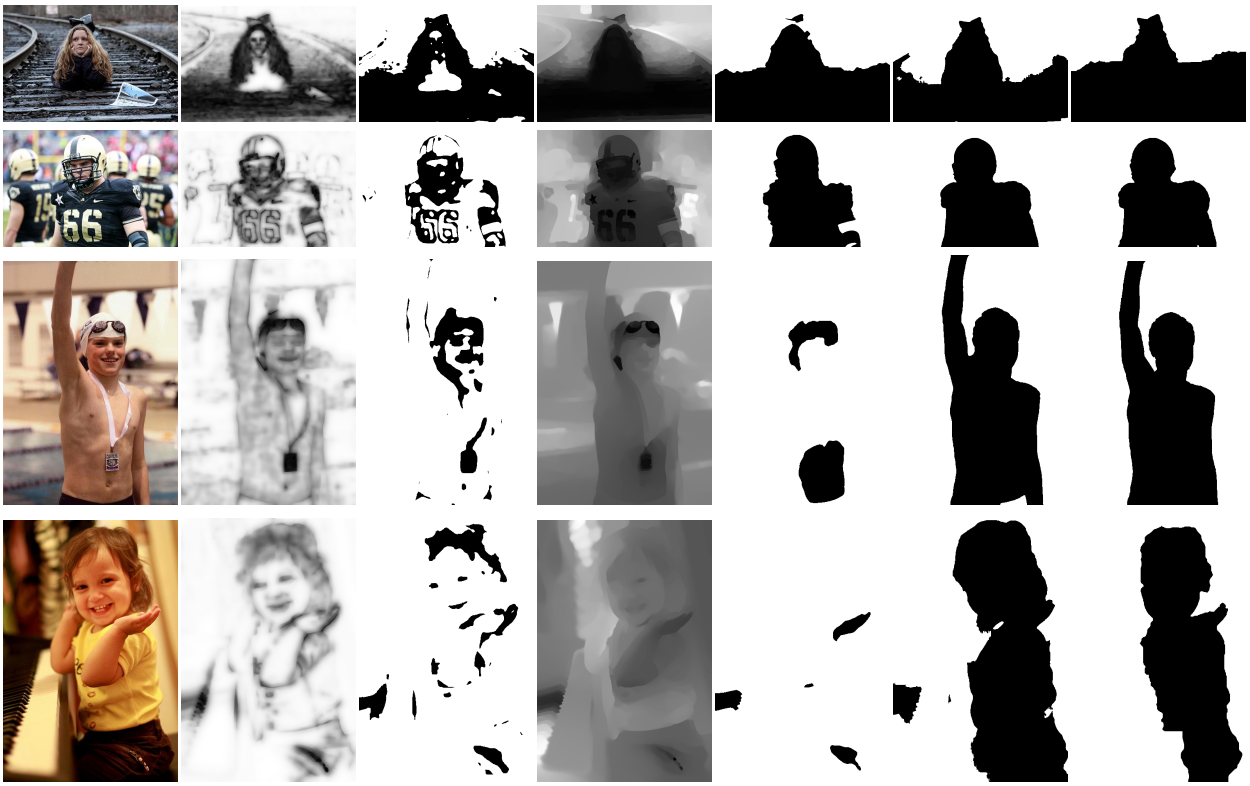


Figure 7: *Our defocus blur detection results compared with other defocus map estimation methods. The results of other methods are binarized to compare blur detection performance. White/black colors in the blur map denotes blur/no-blur regions, respectively.*

Figure 8: *Qualitative results of ablation study. The notations are same with the ones used in Table 3.*

5.4. Blur type classification results

We analyze the effect of our synthetic dataset on blur type classification for defocus blur, motion blur, and no-blur. We compare two models: one after the first training stage, and the other after the second training stage in Sec. 4.4. We make 280 test images using the same method described in Sec. 3.2. We use 140 defocus blurred images in CUHK test set and 400 images in the salient object detection dataset [LY15] that are not used for the training dataset.

For quantitative comparison between the two models, we measured their classification accuracy and mIoU values. Table 4 shows that the latter model that was trained using our synthetic dataset achieves much higher accuracy than the former model that was trained using only CUHK dataset for blur classification.

Figs. 9 and 10 show results of the two models for synthetic and real images having both defocus and motion blur together. The model trained using only CUHK dataset cannot separate motion blur from the defocus-blurred background, as the dataset has no such images with both types of blur together. In contrast, the model trained using both datasets accurately detects both types of blur. We refer the reader to the supplementary material for more results.

6. Applications

Our method can be used for many graphics and vision applications, some of which are presented in this section.

6.1. Deblurring

For deblurring partially blurred images, non-uniform image deblurring methods [WSZP12, HJSHS11] have been proposed. However, these methods assume specific blur models with hard constraints

Figure 9: *Results of blur type classification on synthetic scenes including both defocus and motion blur together. Top row: input images. Second row: results of the model trained using only CUHK dataset. Third row: results of the model fine-tuned using CUHK and our synthetic datasets. Bottom row: ground truth blur maps. Red/blue/black in the blur maps represent defocus blur/motion blur/no-blur regions, respectively.*

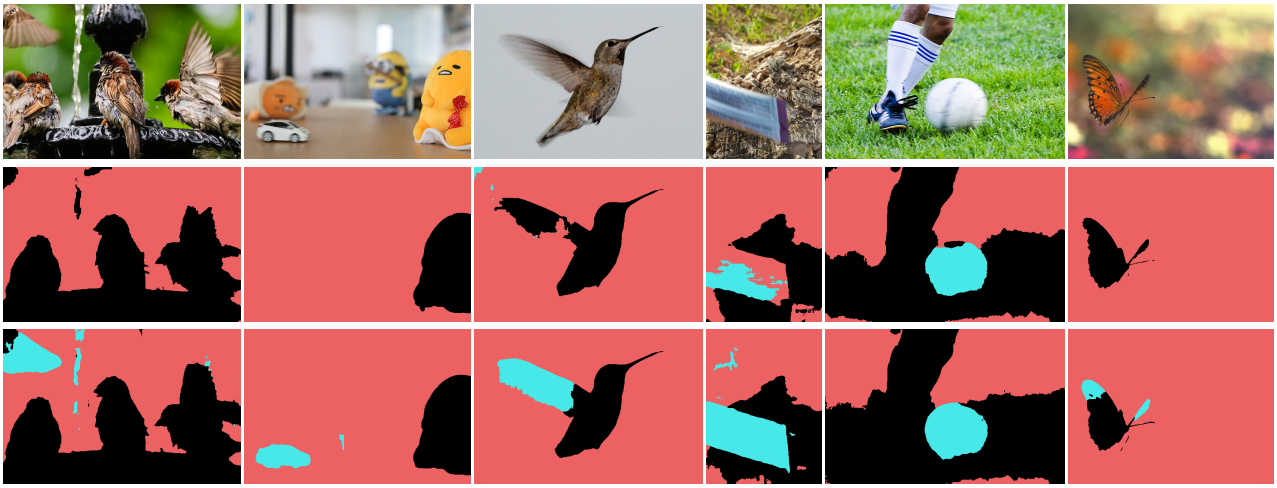
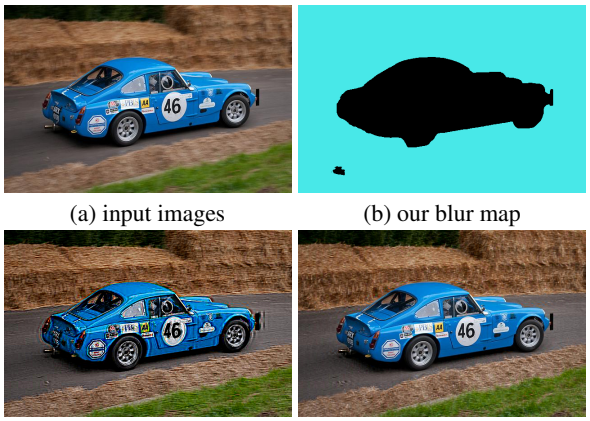


Figure 10: *Results of blur type classification on real scenes including both defocus and motion blur together. Top row: input images. Middle row: results of the model trained using only CUHK dataset. Bottom row: results of the model fine-tuned using both CUHK and our synthetic datasets. Red/blue/black in the blur maps represent defocus blur/motion blur/no-blur regions, respectively. Input images are from Flickr.*

to relieve the severe ill-posedness of the problem. Therefore, it is not easy to apply the methods to partially blurred images containing quite different shapes of blur kernels, e.g., an image taken by a panning shot with focus on a moving object, where the object has a point blur kernel and the background has a motion blur kernel. With our accurate blur detection results, these partially blurred images can be handled by uniform deblurring, which is less complex and more effective than non-uniform deblurring.

Fig. 11 shows an example. To estimate a blur kernel for the region specified by our blur map, we modified the kernel estimation part in an existing uniform deblurring method [PHSY14] to consider only the blurry region while excluding sharp regions. To deblur only the region specified by our blur map with the estimated blur kernel, we also modified the non-blind deconvolution method of [WSZ14]. Fig. 11d shows a result of non-uniform deblurring using our method. For comparison, Fig. 11c shows a result of applying the methods of [PHSY14, WSZ14] to the entire image without using our blur map.



(a) input images (b) our blur map

(c) uniform deblur (d) deblur w/ our blur map

Figure 11: *Deblurring example. (a) non-uniformly blurred scene. (c)&(d) deblurred images obtained by a uniform deblurring method [PHSY14] w/o and w/ our blur map, respectively. The uniform deblurring method estimates a wrong blur kernel for the input image (a), so the deblurred result (c) contains ringing artifacts. In contrast, our blur map (b) can guide the uniform deblurring method to exclude the sharp foreground region in kernel estimation and deconvolution, resulting in a better result.*

6.2. Photo editing with image matting

Extracting a foreground object from the background is one of the most widely used tools for image editing. Our method can be used for foreground segmentation when the foreground and background have different blurriness. Fig. 12 shows an example. Once extracted, the foreground object can be easily composed into other images without noticeable artifacts (Fig. 12d) thanks to our high-quality blur map (Fig. 12b). We can also employ alpha matting [LLW08] to further improve the quality of a composite image. We can use our blur map to obtain the initial trimap, from which an alpha map can be computed (Fig. 12c). A high-quality composition result can be obtained using the alpha map, as shown in Fig. 12e.

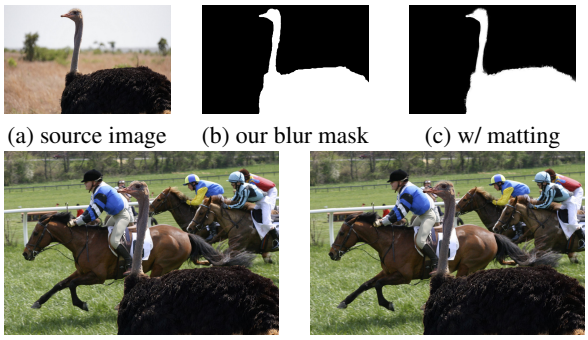
6.3. Blur magnification

Intentional defocus blur is a strong tool for highlighting salient objects and enhancing aesthetic image quality. However, smartphone cameras and many consumer cameras are not capable of expressing large defocus blur because of their physical limitations. Our method provides a simple and plausible solution to this problem. Using our method, we can separate out foreground objects that are in-focus. Then we can blur the background and re-compose the foreground objects back to the blurred background. Fig. 13 shows an example of a natural-looking result without any noticeable artifacts.

7. Conclusions

This paper proposed a novel blur detection method using a deep encoder-decoder framework with long skip-connections. Our network effectively exploits both low-level features for capturing blur around structural edges and high-level features for propagating blur information to homogeneous regions. Despite the limited size of the available dataset, we successfully trained our network using a pre-trained model and multi-scale reconstruction losses. In addition, we built a synthetic dataset for classifying defocus and motion blur and fine-tuned our network to improve the blur classification performance.

In this paper, we considered detection of different kinds of blur, but limited ourselves only to detection rather than estimation. We did not consider a more complicated case either, where an image region may contain both blur types at the same time, e.g., a moving object in a defocused region. For such an image region, our network would detect the dominant blur type, as shown in the supplementary material. Future work includes estimation of blur amounts while handling complex images with more complicated combinations of different types of blur.



(a) source image (b) our blur mask (c) w/ matting

(d) composition using (b) (e) composition using (c)

Figure 12: *Image editing example. Our method can be used for foreground extraction from a blurry background. Image matting [LLW08] can also be used to refine our blur mask for handling complicated object boundaries. The source and the background images are from Flickr.*

(a) input image (b) result (c) our blur map

Figure 13: *Blur magnification example. Our accurate blur map enables naturally magnifying blur amounts in the background.*

Acknowledgements

This work was supported by the Ministry of Science and ICT, Korea, through IITP grant (IITP-2015-0-00174) and NRF grant (NRF-2017M3C4A7066317). It was also supported by the DGIST Start-up Fund Program of the Ministry of Science and ICT (2017040005).

References

[ARG18] AMIN B., RIAZ M. M., GHAFOOR A.: A hybrid defocused region segmentation approach using image matting. *Multidimensional Systems and Signal Processing* (2018), 1–9. 1

[BD07] BAE S., DURAND F.: Defocus magnification. *Computer Graphics Forum 26*, 3 (2007), 571–579. 1

[CDSAP13] COUZINIÉ-DEVY F., SUN J., ALAHARI K., PONCE J.: Learning to estimate and remove non-uniform image blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 1075–1082. 3

[CWL12] CHO S., WANG J., LEE S.: Video deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 64:1–64:9. 1

[CZF10] CHAKRABARTI A., ZICKLER T., FREEMAN W. T.: Analyzing spatially-varying blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010). 1, 2

[EGA*13] EFRAT N., GLASNER D., APARTSIN A., NADLER B., LEVIN A.: Accurate blur models vs. image priors in single image super-resolution. In *IEEE International Conference on Computer Vision (ICCV)* (2013). 1

[GB10] GLOROT X., BENGIO Y.: Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics* (2010). 5

[GK17] GOLESTANEH S. A., KARAM L. J.: Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1, 2, 5, 6, 7

[HCH*17] HOU Q., CHENG M.-M., HU X., BORJI A., TU Z., TORR P.: Deeply supervised salient object detection with short connections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1

[HFF*18] HUANG R., FENG W., FAN M., WAN L., SUN J.: Multiscale blur detection by learning discriminative deep features. *Neurocomputing 285* (2018), 154–166. 1, 2, 4, 5, 6, 7

[HGO18] HENZ B., GASTAL E. S. L., OLIVEIRA M. M.: Deep joint design of color filter arrays and demosaicing. *Computer Graphics Forum 37*, 2 (2018), 389–399. 4

[HJSHS11] HIRSCH M., J. SCHULER C., HARMELING S., SCHÖLKOPF B.: Fast removal of non-uniform camera shake. In *IEEE International Conference on Computer Vision (ICCV)* (2011). 1, 8

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 4

[JLYP13] JIANG P., LING H., YU J., PENG J.: Salient region detection by ufo: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision (ICCV)* (2013). 1

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2014). 5

[KL14] KIM T. H., LEE K. M.: Segmentation-free dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 2766–2773. 3

[LJLS13] LEE D., JEONG S., LEE Y., SONG B. C.: Video deblurring algorithm using accurate blur kernel estimation and residual deconvolution based on a blurred-unblurred frame pair. *IEEE Transactions on Image Processing (TIP) 22*, 3 (2013), 926–940. 1

[LLJ08] LIU R., LI Z., JIA J.: Image partial blur detection and classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008). 1, 2

[LLW08] LEVIN A., LISCHINSKI D., WEISS Y.: A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 30*, 2 (2008), 228–242. 3, 10

[LMSR17] LIN G., MILAN A., SHEN C., REID I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1

[LY15] LI G., YU Y.: Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 1, 3, 8

[MFL*18] MA K., FU H., LIU T., WANG Z., TAO D.: Deep blur mapping: Exploiting high-level semantics by deep neural networks. *IEEE Transactions on Image Processing (TIP) 27*, 10 (2018), 5155–5166. 1, 3, 5, 6, 7

[NHH15] NOH H., HONG S., HAN B.: Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1520–1528. 1

[PHSY14] PAN J., HU Z., SU Z., YANG M.-H.: Deblurring text images via L0-regularized intensity and gradient prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014). 9, 10

[PTCK17] PARK J., TAI Y.-W., CHO D., KWEON I. S.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1, 2, 4, 5, 6, 8

[RDS*15] RUSSAKOVSKY O., DENG J., SU H., KRAUSE J., SATHEESH S., MA S., HUANG Z., KARPATHY A., KHOSLA A., BERNSTEIN M., ET AL.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252. 4

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015). 4

[SBC12] SAAD M. A., BOVIK A. C., CHARRIER C.: Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on Image Processing (TIP) 21*, 8 (2012), 3339–3352. 1

[SCX*17] SUN J., CAO W., XU Z., PONCE J., ET AL.: Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1

[SDW*17] SU S., DELBRACIO M., WANG J., SAPIRO G., HEIDRICH W., WANG O.: Deep video deblurring for hand-held cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 4

[SLJ*15] SZEGEDY C., LIU W., JIA Y., SERMANET P., REED S., ANGUELOV D., ERHAN D., VANHOUCKE V., RABINOVICH A.: Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 4

[SLT11] SU B., LU S., TAN C. L.: Blurred image region detection and classification. In *ACM international conference on Multimedia (MM)* (2011). 1, 2

[SVI*16] SZEGEDY C., VANHOUCKE V., IOFFE S., SHLENS J., WOJNA Z.: Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2818–2826. 5

[SXJ14] SHI J., XU L., JIA J.: Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014). 1, 2, 3, 5, 6, 7

[SXJ15] SHI J., XU L., JIA J.: Just noticeable defocus blur detection and estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 657–665. 1, 5, 6, 8

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 2, 4

[TWH*16] TANG C., WU J., HOU Y., WANG P., LI W.: A spectral and spatial approach of coarse-to-fine blurred image region detection. *IEEE Signal Processing Letters 23*, 11 (2016), 1652–1656. 1, 2, 5, 6, 7

[WSZ14] WHYTE O., SIVIC J., ZISSERMAN A.: Deblurring shaken and partially saturated images. *International Journal of Computer Vision (IJCV) 110*, 2 (2014), 185–201. 9

[WSZP12] WHYTE O., SIVIC J., ZISSERMAN A., PONCE J.: Non-uniform deblurring for shaken images. *International Journal of Computer Vision (IJCV) 98*, 2 (2012), 168–186. 1, 8

[ZS11] ZHUO S., SIM T.: Defocus map estimation from a single image. *Pattern Recognition 44*, 9 (2011), 1852–1858. 1, 2

[ZYW*17] ZENG K., YU J., WANG R., LI C., TAO D.: Coupled deep autoencoder for single image super-resolution. *IEEE Transactions on Cybernetics 47*, 1 (2017), 27–37. 4

[ZZH17] ZHANG L., ZHOU L., HUANG H.: Bundled kernels for nonuniform blind video deblurring. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 9 (2017), 1882–1894. 1