Master's Thesis

Defocus and Motion Blur Detection with Deep Contextual Features

Beomseok Kim (김 범 석)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2018


POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

깊은 문맥적 특징들을 이용한 디포커스 블러와 모션 블러 검출

Defocus and Motion Blur Detection with Deep Contextual Features

Defocus and Motion Blur Detection with Deep Contextual Features

by

Beomseok Kim
Department of Computer Science and Engineering
Pohang University of Science and Technology

A thesis submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Computer Science and Engineering

Pohang, Korea
06. 14. 2018
Approved by
Seungyong Lee
Academic advisor

Defocus and Motion Blur Detection with Deep Contextual Features

Beomseok Kim

The undersigned have examined this thesis and hereby certify that it is worthy of acceptance for a master's degree from POSTECH

06. 14. 2018

Committee Chair Seungyong Lee

Member Minsu Cho

Member Suha Kwak

MCSE 김 범 석. Beomseok Kim
20130732
Defocus and Motion Blur Detection with Deep Contextual Features,
깊은 문맥적 특징들을 이용한 디포커스 블러와 모션 블러 검출
Department of Computer Science and Engineering , 2018, 37p, Advisor : Seungyong Lee. Text in English.

ABSTRACT

This thesis proposes a novel approach to detect two kinds of partial blurs: defocus and motion blurs, by training a deep convolutional neural network. Existing blur detection methods concentrate on designing low-level features, but those features have difficulty in detecting blurs in regions without enough textures or edges. To handle such homogeneous regions, we propose a deep encoder-decoder network with long residual skip-connections and multi-scale reconstruction loss functions, exploiting high-level contextual features as well as low-level structural features. However, adopting a deeper network for our task is not straightforward with two reasons. First, the number of images in an existing dataset is not sufficient. We overcome this issue by employing a pre-trained model that has been trained for image recognition with a large dataset. Second, in the dataset, there are no complex scenes that contain defocus and motion blurs together, so our network cannot learn discriminative features between different kinds of blurs. To resolve this issue, we construct a synthetic dataset that contains complex scenes

with various combinations of blurs. Experimental results show that our approach effectively detects and classifies blurs, outperforming other state-of-the-art methods. Our method can be used for various applications, such as photo editing, blur magnification, and deblurring.


POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

Contents

I. Introduction

Partial blurs of photographs are caused by shallow depth-of-field, camera motions, and moving objects unintentionally or intentionally. Unintentional blurs generally degrade image quality, and many existing methods including defocus map estimation [1, 2, 3] and motion blur kernel estimation [4, 5, 6] concentrate on estimating the blur kernels to remove such blurs. However, accurate blur kernel estimation is a severely ill-posed problem and existing methods usually handle only one type of blur. Meanwhile, localizing partial blurs and identifying their types, rather than estimating the amounts of blurs, are also important tasks for various applications, such as image editing [7, 8], image quality assessment [9], image restoration [10], and saliency detection [11]. In addition, blur detection and classification can be used for partial deblurring [4, 5] by specifying blurred regions and their types.

Most blur detection approaches [12, 13, 14, 15, 16, 17] 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 blurs in flat regions without enough texture or edges (Fig. 1.1). Some defocus blur map estimation methods [1, 2, 3] estimate the amounts of blurs around edges, and propagate the estimates to flat regions. However, this propagation based approach still has difficulty in filling large homogeneous regions with accurate blur estimations.

Recently deep convolutional neural networks (DCNN) have led remarkable successes in various image processing tasks, e.g., semantic segmentation [18, 19] and saliency detection [20, 21]. There are a few deep learning approaches for blur detection and estimation [22, 3]. The approaches train CNN features which are

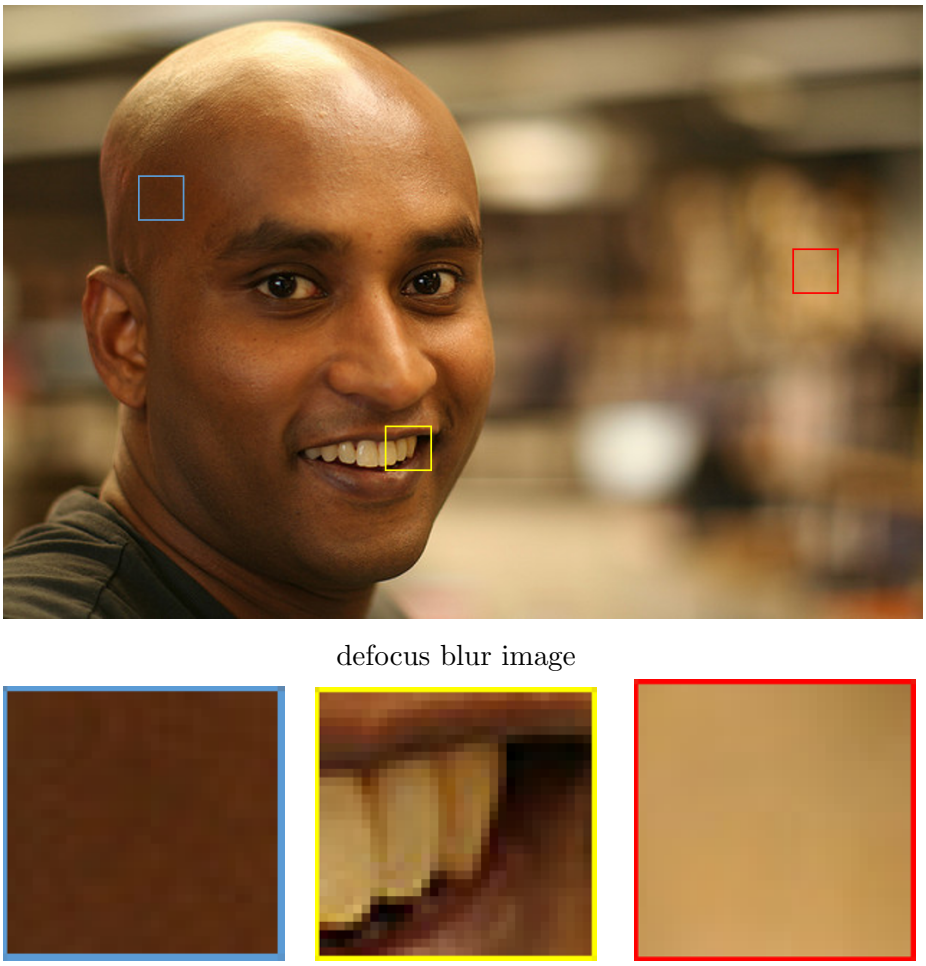


Figure 1.1: Example of a challenging case in blur detection. Blue and yellow frame patches are sharp regions, whereas red frame patch is a defocus blur region. However, it is hard to determine whether a homogeneous patch (e.g., the blue frame patch from a human face) is blurred or not at the patch level.

discriminative to blurs in the unit of small patches, and reconstruct a full blur map by merging or interpolating the patch responses. These methods using CNN outperform previous blur detection and estimation methods, but CNN features with small receptive fields have limitation on detecting blurs in homogeneous regions.

In this thesis, we propose a novel deep encoder-decoder network with long

residual skip-connections to detect and classify partial blurs, i.e., defocus and motion blurs. Differently from existing methods [22, 3], our network considers deep contextual features with a larger receptive field to detect blurs in homogeneous regions, and predicts the blur at each pixel rather than for each patch. To effectively train our network while exploiting both high-level and low-level features together, we use multi-scale reconstruction loss functions and adopt element-wise summation, instead of concatenation, for residual skip-connections.

However, applying a deeper network for blur detection and classification is not straightforward due to two reasons. First, there is no big enough dataset for training our deeper model. To the best of our knowledge, there exists only one dataset, CUHK blur detection dataset [15], but it contains 1,000 labeled images. To prevent overfitting and facilitate training, we initialize our encoder network using a pre-trained model, VGG-19 [23], 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. Second, all the images in the dataset have only one type of blur and there are no images containing defocus and motion blurs together. Therefore, we generate a synthetic dataset for training our network to discriminate between defocus and motion blurs in complex scenes. We employ our dataset for fine-tuning our network after training with CUHK dataset.

Experimental results show that our approach is effective and outperforms the-state-of-the-are methods. 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 a novel deep encoder-decoder network with long residual skip-connections to detect two kinds of blurs: defocus and motion blurs. Our multi-scale reconstruction loss functions help our network reconstruct pre-

cise blur maps by fully utilizing high-level and low-level features.

- We employ a pre-trained model from a large-scale dataset for our encoder network. It enables our network to overcome the small size of the dataset and exploit high-level semantic features.

- We construct a synthetic dataset that consists of complex scenes including both defocus and motion blurs together. Fine-tuning with the dataset makes our network learn discriminative features between defocus and motion blurs.

II. Related Work

2.1 Blur detection and classification

Many effective blur detection methods based on hand-crafted features have been proposed. Liu *et al.* [12] detect partial blurs using four local blur features and classify two kinds of blurs. Chakrabarti *et al.* [13] analyze directional blurs via local Fourier transform. Su *et al.* [14] 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.* [15] propose a set of blur features in multiple domains for blur detection, such as gradient histogram span, kurtosis, and data-driven local filters. They build the largest real blur detection dataset containing 1,000 images labeled with pixel-level annotation. Tang *et al.* [16] introduce image spectrum residual to detect a coarse blur map, and an iterative update algorithm to refine the blur map by filling flat regions. Golestaneh and Karam [17] detect spatially-varying blurs by applying multiscale fusion to high frequency Discrete Cosine Transform (DCT) coefficients. Although all the methods mentioned above can effectively discriminate blurs around edges, blurs in flat regions are not properly analyzed as their hand-crafted features do not properly handle flat regions.

2.2 Defocus map estimation

Defocus map estimation methods compute the amounts of spatially-varying defocus blurs for a given image. They are closely related to blur detection in that partial blurs are detected and a blur map is generated. Zhuo and Sim [1] obtain a sparse defocus map from the gradient ratio between input and re-blurred

image patches with a Gaussian kernel. A sparse defocus map is then interpolated to produce a full defocus map. Shi *et al.* [15] develop a sparse representation for detecting just noticeable blur and estimating its strength. However, the blur amounts around edges are propagated to homogeneous regions, and the estimated blurs are not accurate in large homogeneous regions and too strong edges such as text boundaries.

2.3 Deep learning based methods

Several deep learning based methods have been introduced to understand image blurs. Rui *et al.* [22] detect a local blur regardless of its type using a CNN, which 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 blurs in homogeneous regions due to narrow receptive fields. Besides, their results are not precisely aligned to original image structures (e.g., object boundaries) as the blurs are detected in terms of patches. Park *et al.* [3] 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 the proposed features in small patches. Second, local blur information around edges is propagated into surrounding smooth regions using matting Laplacian [24]. However, their approach often estimates inaccurate blurs in large flat regions, and does not consider motion blurs.

Our approach is different from previous deep learning based methods in three aspects. First, we construct our network with fully convolutional layers, avoiding computation of duplicated CNN features on many small patches. Second, our network exploits deep contextual features of homogeneous regions with a large

receptive field, and can directly detect blurs in smooth regions in an end-to-end manner without a post-processing step of additional propagation or interpolation. Third, our network can distinguish defocus and motion blurs as well as localizing them.

III. Dataset for Blur Detection and Classification

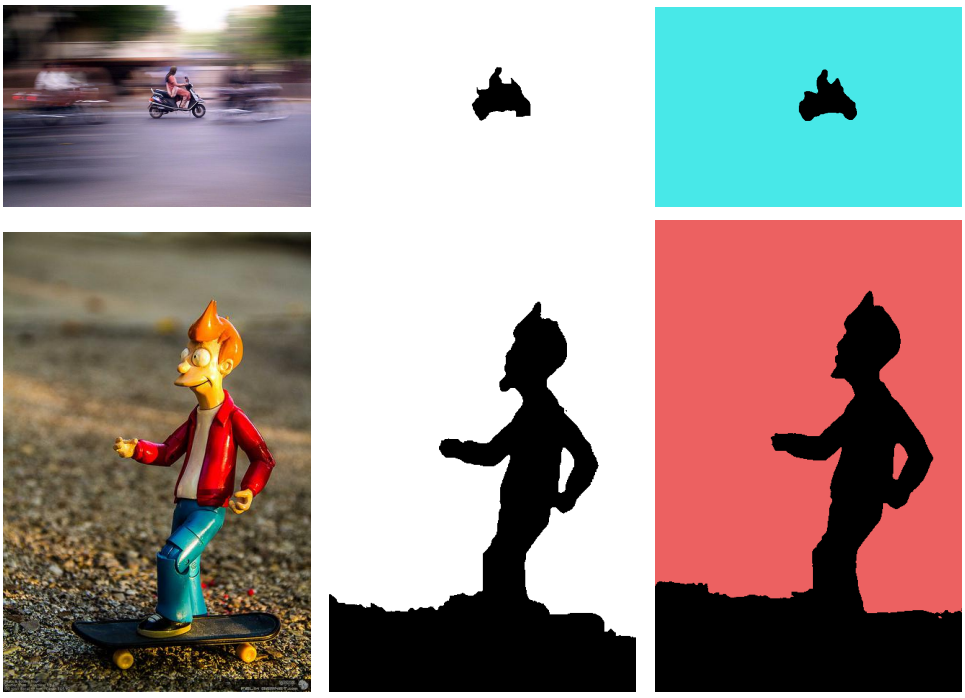
3.1 Blur detection dataset

CUHK dataset [15], which is the only 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, where each pixel of the blur map indicates whether the corresponding pixel in the image is blurred or not. However, it is assumed that there exists only one type of blur in an image, and distinction among defocus and motion blurs is provided in the unit of images, not inside an image. Fig. 3.1 shows examples 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 blurs, we need separate labels for different blur types. We convert the ground-truth labels in CUHK dataset by reflecting the blur types of images onto the pixel labels in the blur maps. Then, using the converted blur maps and our synthetic dataset, which will be discussed next, we train our network to classify two kinds of blurs at each pixel as well as to detect them. The right column in Fig. 3.1 shows results of our network, where blue and red denote motion and defocus blurs, respectively.

3.2 Synthetic dataset for blur classification

Most images in CUHK dataset do not contain motion and defocus blurs together. While some images have both types of blurs in the dataset, they are not properly annotated and only one type of blur is considered in those images.



input ground truth our result

Figure 3.1: Examples of the ground truths in CUHK dataset and our blur detection results. In a ground truth map, white is blurred and black is non-blurred regardless of the blur type. In our result, blue is motion blur, red is defocus blur, and black is non-blurred.

Consequently, training with only CUHK dataset makes a network less effective to distinguish different types of blurs in one image, as will be shown in Sec. 5.4. To resolve this limitation, we construct a new synthetic dataset that contains both types of blurs in single images with pixel-wise annotations (Fig. 3.2).

We generate our synthetic dataset using CUHK dataset and a salient object detection dataset [20]. 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, and we finally obtain $8,460$ images that contain both defocus and motion blurs. In the algorithm, we use an intermediate image s to produce plausible motion blurs of salient objects, which incorporate the background information.

blurred image ground truth blur map

Figure 3.2: Our synthetic dataset. In CUHK dataset [15], there are only defocus/sharp and motion/sharp labeled images. We generate motion/defocus/sharp images (left) utilizing a salient object detection dataset, where our images contain moving objects in out-focusing photos. A ground truth blur map (right) is labeled in black (sharp), blue (motion), and red (defocus).

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

extract the salient object using m

$s \leftarrow$ image synthesized with the salient object on top of b

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

IV. Blur Detection and Classification using DCNN

4.1 Network architecture

Our network takes a 3-channel RGB image as the 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 [25]. Fig. 4.1 shows the detailed structure of our network.

The encoder network consists of four max-pooling layers and 16 convolutional layers, similarly to VGG-19 [23] but without fully connected layers. Thanks to max-pooling layers and convolutional layers of the encoder, our network has much larger receptive fields than previous deep learning based approaches [22, 3]. 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 deconvolutional and 15 convolutional layers to upsample features and generate an output of the same size as the input.

We add four long symmetric skip-connections with a 1×1 convolutional 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. In contrast to U-Net, we adopt element-wise summation, instead of concatenation, when forwarding features using skip-connections.

In the network, features in the encoder include structural information ex-

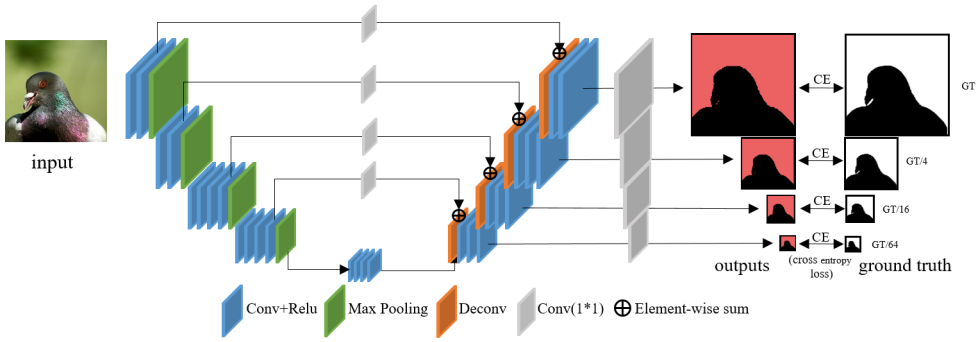


Figure 4.1: Our blur detection network with multi-scale reconstruction loss functions.

tracted from the finest scale, and features in the decoder include contextual information dilated from the coarsest level. With concatenation, the network could selectively use structural and contextual features to construct the final output. Instead, we use element-wise summation to explicitly enforce all features from the encoder and decoder to be used together. This modification helps our network exploit contextual features that are needed to properly handle large homogeneous regions.

4.2 Pre-trained model as encoder

While we have CUHK and our own synthetic datasets, the number of training images is still not enough. Directly training a complex network with such a limited dataset inevitably incurs the over-fitting problem. To resolve this problem, we use a pre-trained model, VGG-19 [23] trained with ImageNet dataset [26] for image classification. Although image classification and blur detection are different tasks from each other, 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 also 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 on propagating well-trained features even only with a limited dataset. We demonstrate the effectiveness of a pre-trained model 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 [27], which is used for stable training of a network (Fig. 4.1). We express our network f as:

$$\hat{\mathbf{b}}_s = f_s(\mathbf{I}; \theta) \tag{4.1}$$

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 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 types of blurs. Then, training is done by minimizing the loss function defined as:

$$L(\theta; \mathbf{I}, \mathbf{b}) = -\sum_s \sum_p \sum_c w_s \mathbf{b}_{s,p,c} \log(\hat{\mathbf{b}}_{s,p,c}), \tag{4.2}$$

where $\mathbf{b}$ is the ground-truth blur map and $\mathbf{b}_s$ is a downsampled version of b 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 using softmax across channels. 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 the 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-level and low-level features so that mid- and high-frequency information can be accurately reconstructed.

4.4 Implementation details

4.4.1 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.

4.4.2 Training setting

We implemented our network with Tensorflow and used an NVIDIA Titan XP GPU to train the network. We used Adam optimizer [28] for training. The encoder is initialized with pre-trained VGG-19 weights as explained before. The decoder is initialized by xavier initialization [29]. 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 $10e-4$ for the decoder and $10e-5$ for the encoder, and the network was trained for $3,000$ epochs. Then, the learning rate was set to $10e-5$ for the decoder, and $10e-6$ for the encoder, and another $3,000$ epochs were run. In the second stage, we use CUHK dataset and our synthetic dataset to further improve the blur detection and classification performance. We set the learning rate to $10e-5$ for the decoder and $10e-6$ 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 evaluations unless otherwise noted.

V. Experimental Results

For all quantitative evaluations in this section, we use 200 images from CUHK dataset that we did not use 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 against previous methods. As our network produces three labels for each pixel, we obtain a binary blur map by merging the motion blur and defocus blur in the output of our network.

We compare our method with previous blur detection methods [14, 15, 16, 17, 22], and defocus map estimation methods [2, 3]. 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. We find the best threshold for each method for comparison. Specifically, we tried 255 uniformly sampled thresholds to the results of previous methods on the test set of 200 images. Then, we chose the best one so that we can compare our results against their best results. As we use two metrics for our quantitative evaluation as discussed later, we chose two thresholds for each method, each of which produces the best results for each metric. Resulting binary blur maps including ours have either 0 or 1, which represent no-blur and blur, respectively.

We use two metrics to evaluate our binary blur map: accuracy, and F-

measure. Each measure is defined as:

$$Accuracy = \frac{1}{w \times h} \sum_{p} (1 - |B_p - G_p|) \tag{5.1}$$

$$F_\beta = \frac{(1+\beta) precision \times recall}{\beta^2 precision + recall} \tag{5.2}$$

where B is an estimated blur map and G is the ground truth map. $w \times h$ is the image size. β is a parameter for F-measure. We use $\beta = 1$, which means the same importance of precision and recall, where they are defined as:

$$precision = \frac{\sum_p B_p G_p}{\sum_p B_p}, \quad \text{and} \quad recall = \frac{\sum_p B_p G_p}{\sum_p G_p}. \tag{5.3}$$

5.1 Comparison with other blur detection methods

We first compare our method with other state-of-the-art blur detection methods: Su *et al.* [14], Shi *et al.* [15], Tang *et al.* [16], Golestaneh *et al.* [17], and Rui *et al.* [22]. Results of other methods are generated by the authors' implementations or downloaded from their project pages. We note that Rui et al.'s method is also trained using CUHK dataset. They used 200 images from CUHK dataset as a test set. Table 5.1 shows a quantitative comparison between the other methods and ours. As mentioned earlier, we tried 255 different threshold and used the best one for the results of other methods. Nonetheless, our method clearly outperforms all the others by a large margin in terms of both metrics. The overall precision-recall curves are shown in Fig. 5.1.

Fig. 5.2 shows a qualitative comparison. Our results are visually close to the ground truth maps compared to the other results. Our results have less noise and stains, while edges in our results are more accurately aligned to those in the ground truth maps. Morever, blur in homogeneous regions are more accurately detected as well, as shown in the ninth and eleventh rows in Fig. 5.2.

	Su [14]	Shi [15]	Tang [16]	Golestaneh [17]	Rui [22]	Ours (final)
Max accuracy	0.7420	0.7423	0.7701	0.8010	0.7949	**0.9033**
Max F-measure	0.8139	0.8055	0.8295	0.8417	0.8485	**0.9092**

Table 5.1: Quantitative comparison with other blur detection methods on CUHK test set.

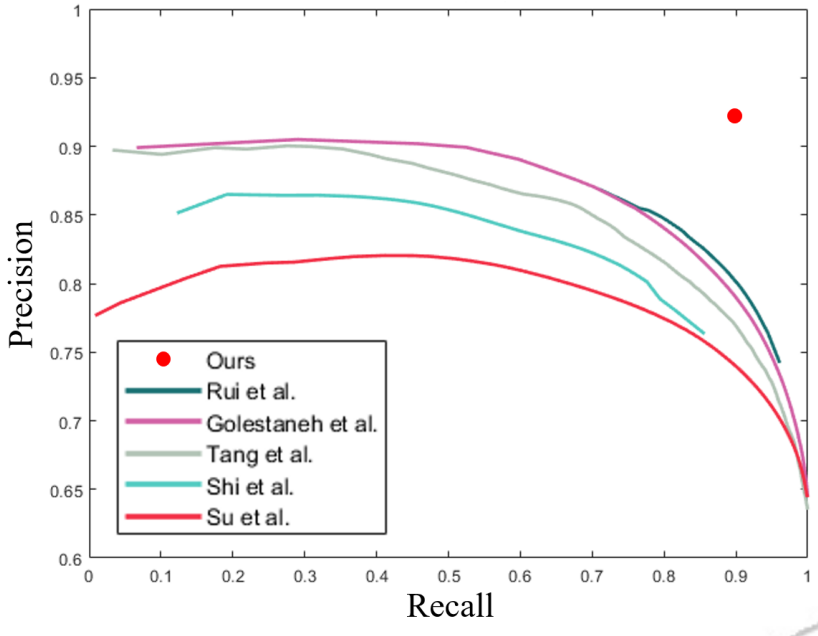
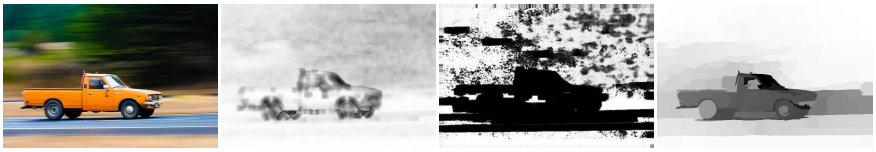
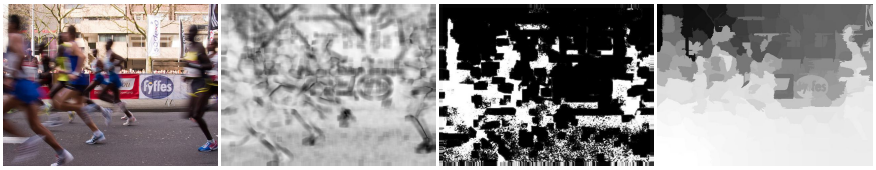


Figure 5.1: Precision-recall graph for blur detection. Note that our result has a fixed precision-recall value because our task is a 3-way classification. Our method achieves much higher precision than other methods with a high recall.

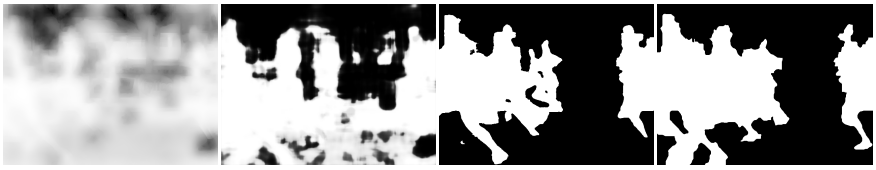


input Su [14] Shi [15] Tang [16]

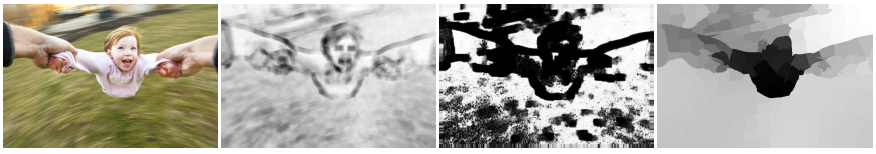
Golestaneh [17] Rui [22] ours ground-truth



input Su [14] Shi [15] Tang [16]



Golestaneh [17] Rui [22] ours ground-truth



input Su [14] Shi [15] Tang [16]

Golestaneh [17] Rui [22] ours ground-truth

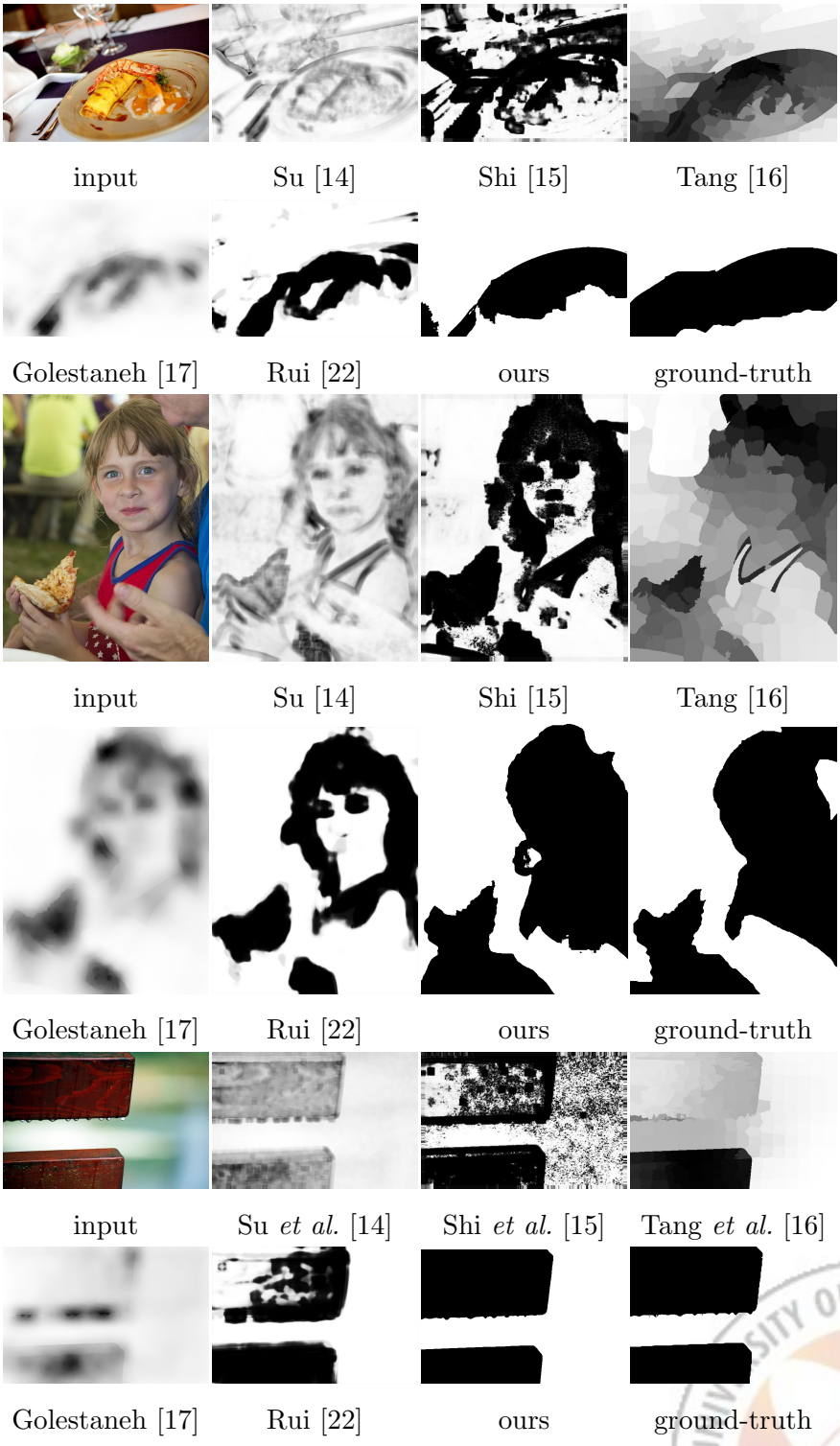


Figure 5.2: Qualitative comparison with other blur detection results. White/black colors in the blur map denotes blur/no-blur regions respectively.

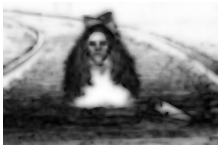
5.2 Comparison with other defocus map estimation methods

We also compare our method with the state-of-the-art defocus map estimation methods [2, 3], 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 5.2 shows a quantitative comparison. Our method again clearly outperforms the others.

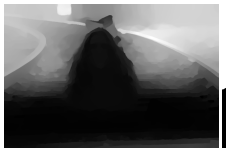
Fig. 5.3 shows a qualitative comparison. We select various defocus images including challenging cases for visual comparison. Previous defocus estimation methods fail in flat regions such as human faces because human faces have too weak edges. While the results of Park et al. [3] in the fourth and eighth 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. Meanwhile, in the case of strong and dense edges such as characters, they propagate wrong defocus blur information . However, our method successfully detected blur in flat regions with much less error around strong edges because our method considers large contextual information.

	Shi [2]	Park [3]	Ours
Max accuracy	0.7903	0.8239	**0.9142**
Max F-measure	0.8393	0.8616	**0.9340**

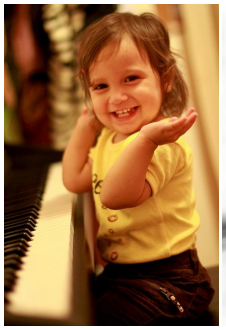
Table 5.2: Quantitative comparison with other defocus map estimation methods on CUHK defocus test set. The results of other methods were converted into binary blur maps.

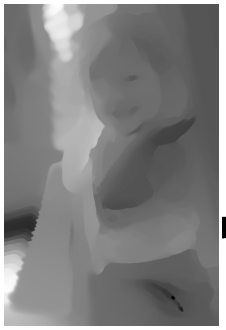

input Shi [2] Shi [2] (binarization)




Park [3] Park [3] ours (binarization) ground-truth



input Shi [2] Shi [2] (binarization)




Park [3] Park [3] ours (binarization) ground-truth

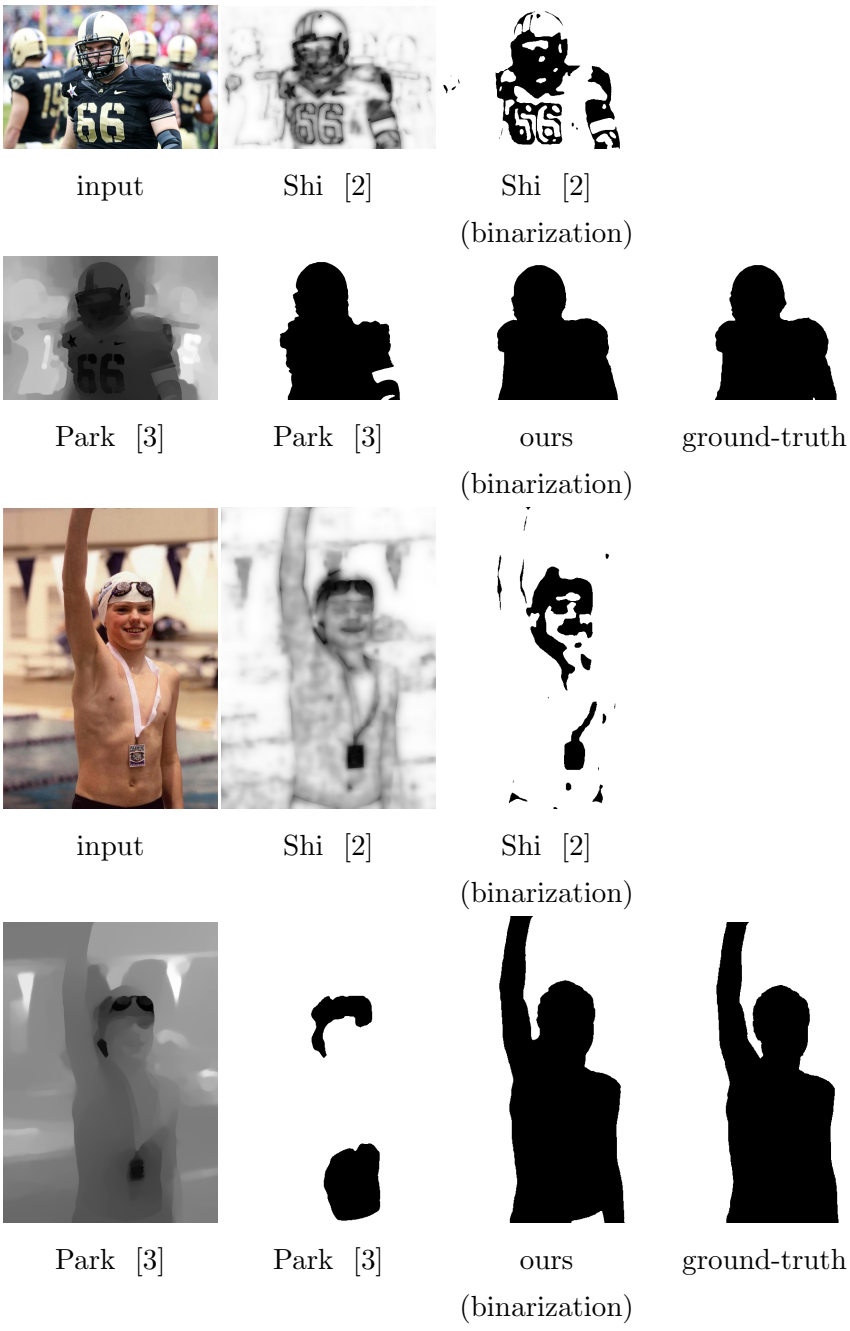


Figure 5.3: Blur detection results compared with other defocus map estimation methods. The results of other methods are binarized to compare blur detection performance. White/black colors in the blur map denotes blur/no-blur regions respectively.

5.3 Ablation study

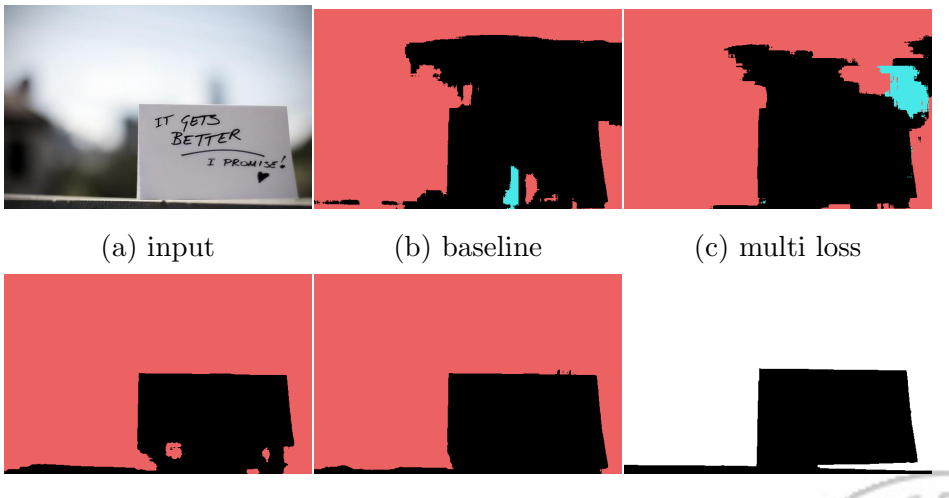
We perform an ablation study to see how each component of our method affects the performance of blur detection. We compare four different models that are trained using different strategies, but use the same network architecture. The quantitative results are shown in Table 5.3. The first three models are trained with the 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 a pre-trained VGG-19 using the multi-scale reconstruction loss functions. The fourth one is our final model, which uses all the components and is fine-tuned using a mixture of CUHK and our synthetic dataset. 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. Even though the difference between our third and final model is not significant, the effect of our final model on blur type classification will be described in the following Sec. 5.4.

The qualitative comparison in Fig. 5.4 clearly reveals the improvement from each component. As shown in Fig. 5.4, the baseline model has many errors in flat regions such as the sky and the paper. The result of the second model shows much less error than the first one, although it still has a significant 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 result of our third and final model has almost no difference in error. This shows that the well-trained features of a pre-trained VGG-19 are very effective to detect and propagate blurriness.

	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**

Table 5.3: Quantitative results of ablation analysis. Baseline model uses single loss and is learned from scratch. Multi loss model uses multi-scale reconstruction loss functions and is learned from scratch. Multi loss + VGG model uses multi-scale losses and initialized by the weights of VGG 19. Note that these three models are trained using CUHK dataset. The final model uses both components and is fine-tuned using both CUHK and our synthetic datasets



(a) input (b) baseline (c) multi loss

(d) multi loss+VGG (e) final (f) ground truth

Figure 5.4: Qualitative results of ablation analysis. The notations are same with the ones used in Table 5.3

5.4 Blur type classification results

We analyze the effect of our synthetic dataset on the quality. To this end, we compare two models: one after the first training stage, and another one after the second training stage as referred in Sec. 4.4. We make 280 test images by using the same method described in Sec. 3.2. In detail, we use 140 defocus blurred images in the CUHK testset and 400 images in the salient object detection dataset [20], none of which are used to make the training dataset.

For the quantitative comparison between these two models, we measure a classification accuracy that is the ratio of correctly predicted pixels among 3 classes: defocus blur, motion blur, and no-blur. Table 5.4 shows the accuracy of the two models for the synthetic test set. The latter model has a much higher accuracy than the former model for blur classification.

Fig. 5.5 and Fig. 5.6 show results of these two models for synthetic and real images having both defocus and motion blur simultaneously. The network trained using only CUHK dataset cannot separate motion blur from the defocus-blurred background, because CUHK dataset has no such images with annotations of both types of blur concurrently. On the other hand, the network trained using both datasets accurately detects both types of blur.

	only CUHK	mixture dataset of CUHK and ours
accuracy	0.8404	0.9075

Table 5.4: Quantitative comparison between our models trained with and without our synthetic dataset that consists of complex scenes including both types of blurs together.

Figure 5.5: Visual results of spatially-varying blur type classification with synthetic scenes including both defocus and motion blurs together. Top row: input images. Second row: results of the model, trained by only the CUHK dataset. Third row: results of the model, fine-tuned by the mixture of CUHK and our synthetic dataset. Bottom row: ground-truth blur map. Red/blue/black colors in the blur map denote defocus blur/motion blur/no-blur regions respectively.

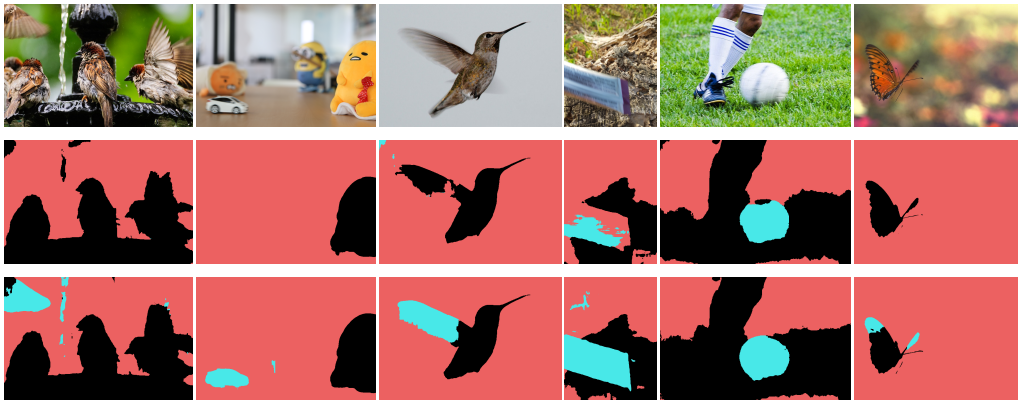


Figure 5.6: Visual results of spatially-varying blur type classification with real scenes including both defocus and motion blurs together. Top row: input images. Second row: results of the model, trained by only the CUHK dataset. Bottom row: results of the model, fine-tuned by the mixture of CUHK and our synthetic dataset. Red/blue/black colors in the blur map denote defocus blur/motion blur/no-blur regions respectively. We took the input images from Flickr.

VI. Applications

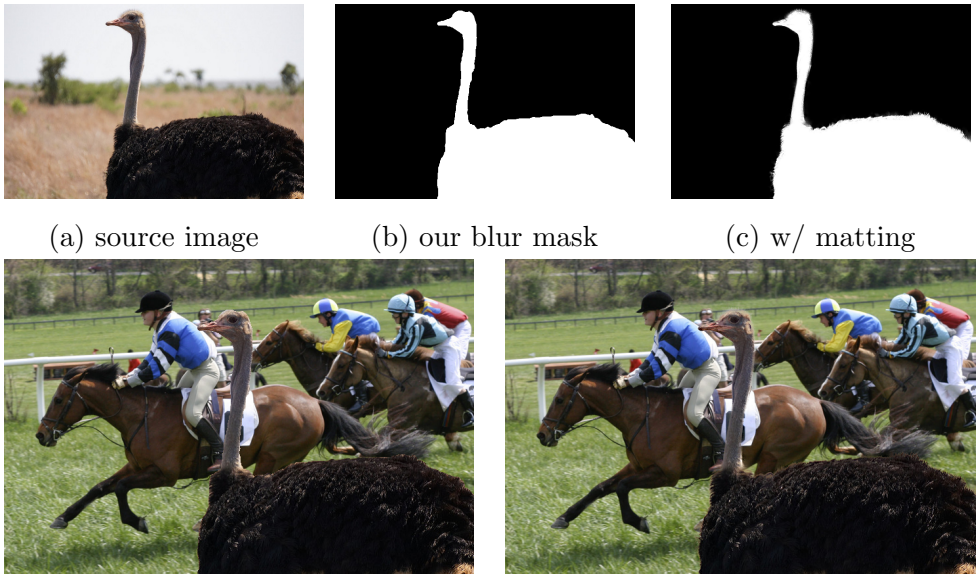
Our method can be used for many graphics and vision applications, some of which are presented in this section.

6.1 Photo editing with image matting

Extracting the 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. 6.1 shows an example. Once extracted, the foreground object can be easily composed into other images without noticeable artifacts (Fig. 6.1d) thanks to our high-quality blur map (Fig. 6.1b). We can employ alpha matting [24] to further improve the quality of composite image. Our blur map can be used for the initial trimap for alpha matting, the composed image with the final alpha map produces a high-quality result (Fig. 6.1e).

6.2 Blur magnification

Intentional defocus blur is a strong cue for highlighting salient objects and enhancing aesthetic image quality. However, smartphone cameras and many consumer cameras are not capable of expressing large defocus blurs because of their physical limitations. Our method provides a simple and plausible solution to this problem. Using our method, we can separate out the foreground objects that are in-focus. Then we can blur the background and re-compose the foreground object back to the blurred background. Fig. 6.2 shows an example, where we can obtain a naturally looking result without any noticeable artifacts.



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

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

Figure 6.1: Image editing example. Our method can be used for foreground extraction from a blurry background. Image matting [24] can also be used to refine our blur mask for handling complicated object boundaries. We took the images from Flickr.

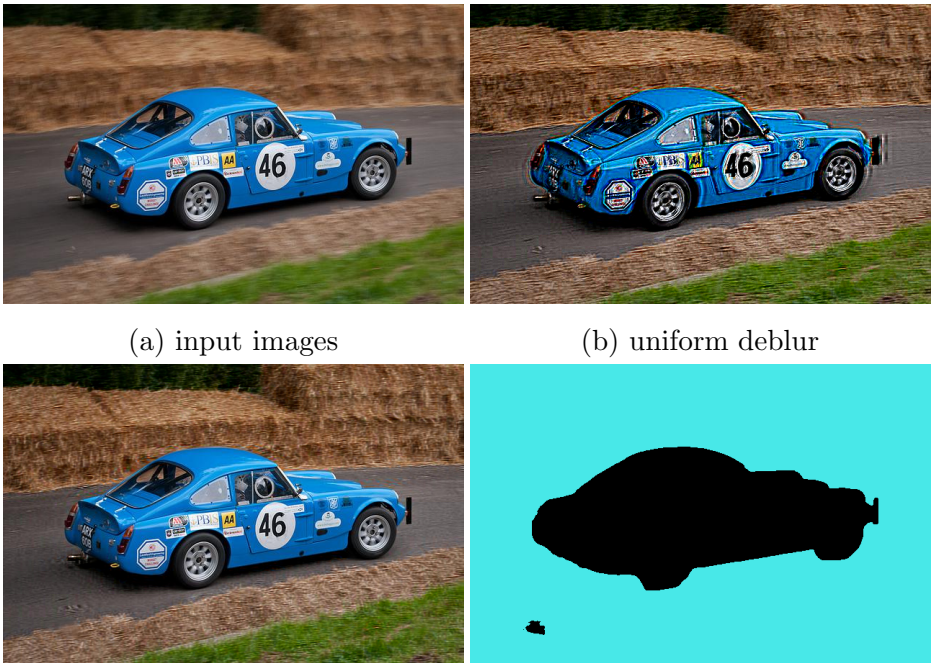
(a) input image (b) result (c) our blur map

Figure 6.2: Blur magnification result. Our accurate blur map enables naturally magnifying blur amounts in the background.

6.3 Deblurring

For deblurring partially blurred images, many non-uniform image deblurring methods [5, 4] have been proposed. However, these methods assume specific blur models with hard constraints 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, where the foreground 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. 6.3 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 [30] 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 [31], generating the deblurred image in Fig. 6.3c. For comparison, Fig. 6.3b shows the result of applying the methods of [30, 31] to the entire image without using our blur map.



(a) input images (b) uniform deblur

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

Figure 6.3: Deblurring result. (a) non-uniformly blurred scene. (b)&(c) deblurred images obtained by a uniform deblurring method [30] w/o and w/ our blur map, respectively. The uniform deblurring method estimates a wrong blur kernel for the input image (a), and the deblurred result (b) contains ringing artifacts. In contrast, our blur map (d) can guide the uniform deblurring method to exclude the sharp foreground region in kernel estimation and deconvolution, resulting in a better result.

VII. Conclusions

This thesis 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 blurs at structural edges and high-level features for providing 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 blurs and fine-tuned our network to improve the blur classification performance.

In this thesis, we considered detection of different kinds of blurs, but limited ourselves only to detection rather than estimation. Future work includes estimation of blur amounts when different kinds of blurs are combined in complex scene images.

References

[1] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011.

[2] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 657–665, 2015.

[3] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. 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.

[4] Michael Hirsch, Christian J. Schuler, Stefan Harmeling, and Bernhard Schölkopf. Fast removal of non-uniform camera shake. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[5] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision (IJCV)*, 98(2):168–186, 2012.

[6] Jian Sun, Wenfei Cao, Zongben Xu, Jean Ponce, et al. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Soonmin Bae and Frédo Durand. Defocus magnification. *Computer Graphics Forum*, 26(3):571–579, 2007.

[8] Benish Amin, Muhammad Mohsin Riaz, and Abdul Ghafoor. A hybrid defocused region segmentation approach using image matting. *Multidimensional Systems and Signal Processing*, pages 1–9, 2018.

[9] Michele A Saad, Alan C Bovik, and Christophe Charrier. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on Image Processing (TIP)*, 21(8):3339–3352, 2012.

[10] Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, and Anat Levin. Accurate blur models vs. image priors in single image super-resolution. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[11] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[12] Renting Liu, Zhaorong Li, and Jiaya Jia. Image partial blur detection and classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[13] Ayan Chakrabarti, Todd Zickler, and William T Freeman. Analyzing spatially-varying blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[14] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *ACM international conference on Multimedia (MM)*, 2011.

[15] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[16] Chang Tang, Jin Wu, Yonghong Hou, Pichao Wang, and Wanqing Li. A spectral and spatial approach of coarse-to-fine blurred image region detection. *IEEE Signal Processing Letters*, 23(11):1652–1656, 2016.

[17] S Alireza Golestaneh and Lina J Karam. 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.

[18] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528, 2015.

[20] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] Rui Huang, Wei Feng, Mingyuan Fan, Liang Wan, and Jizhou Sun. Multi-scale blur detection by learning discriminative deep features. *Neurocomputing*, 285:154–166, 2018.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):228–242, 2008.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

[29] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[30] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via L0-regularized intensity and gradient prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[31] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *International Journal of Computer Vision (IJCV)*, 110(2):185–201, 2014.

Acknowledgements

대학원에 진학하고 벌써 2년이 지나 석사과정 졸업을 앞두고 있습니다. 부족함이 많았던 저이지만 따뜻하고 냉철하게 지도해주신 이승용 교수님께 진심으로 감사의 말씀을 드립니다. 교수님은 저의 의견들을 나무라지 않고 모두 들어주었고, 목표와 다르게 연구를 하고 있으면 바로 잡아 주었습니다. 또한, 바람직한 연구자로 사는 삶을 몸소 보여주어 앞으로 제가 어떤 자세로 살아야 할지 배울 수 있었습니다. 지난 2년 동안 학업적인 성장은 물론이고, 교수님은 제가 무엇을 하든 잘할 수 있는 자신감을 만들어 주셨습니다. 아직도 배울 것들이 많지만, 교수님의 가르침들을 바탕으로 이 학교를 떠나 새로운 시작을 하려고 합니다.

바쁘신 와중에도 제 학위논문 심사를 맡아주신 조민수 교수님, 곽수하 교수님께도 감사드립니다. 개인적으로 존경하던 두 교수님 앞에서 발표할 수 있어서 영광이었습니다. 심사 도중 두 분의 날카로운 지적들과 아낌없는 조언들은 그동안 부족했던 점을 되돌아볼 수 있게 하였고 향후 연구들에 반영하겠습니다. 또한, 논문 작성에 있어 함께 토의하고 큰 도움을 주신 조성현 교수님께도 감사드립니다. 연구실 친한 선배처럼 자유롭게 토의할 수 있게 해주셨고, 그간 많은 조언에 감사드립니다.

2년 동안 함께 했던 연구실 사람들에게도 감사드립니다. 자유롭고 친근한 연구실 분위기는 제가 대학원 생활에 잘 적응하게 만들어 주었습니다. 모르는 게 있으면 자유롭게 물어보고, 연구가 정리가 안 되거나 아이디어가 있으면 토의를 하면서 많은 것들을 배울 수 있었습니다. 다들 똑똑하고 잘하는 것들도 매우 다양한데, 함께 지내면서 다양한 장점들을 배울 수 있어서 좋았습니다. 학업뿐 아니라 심적으로도 동고동락하며 연구실 생활에서 큰 힘이 되었습니다. 다들 지금 하는 연구를 잘 마무리했으면 좋겠고, 졸업 후에도 자주 보고 싶습니다.

그리고 약 7년가량 학부와 대학원 생활 동안 만났던 모든 인연에 감사드립니다. 특히, 대학교 1학년 때부터 함께 많은 추억을 남긴 일레븐 동기들, 수업도 같이 듣고 저와 친하게 지냈던 컴공과 후배들에게 감사드립니다. 지금까지 타지 포항 생활을 하는 데 큰 힘이 되었고, 이런 소중한 인연들을 만난 것을 행운이라 생각합니다.

마지막으로 항상 제 편인 가족들에게 감사드립니다. 저를 믿고 지지해주신 아버지, 저를 위해 헌신하신 어머니, 개구쟁이지만 형을 잘 따르는 동생 용석이 모두에게 감사드립니다. 가족들의 믿음과 사랑이 있었기에 지금의 제가 존재하고, 앞으로도 자랑스러운 아들 또는 형이 되기 위해 노력하겠습니다.

지금까지 포항에서의 학부와 대학원 생활 중 있었던 배움과 추억들을 잊지 않을 것이고 다시 한번 감사드립니다. 졸업은 끝이 아니라 새로운 시작이라는 생각을 가지고 항상 겸손하고 열정적으로 사는 김범석이 되겠습니다.

2018년 6월 김범석 올림

Curriculum Vitae

Name : Beomseok Kim

Education

2009. 3. – 2016. 8. Department of Computer Science and Engineering, Pohang University of Science and Technology (B.S.)

2016. 9. – 2018. 8. Department of Computer Science and Engineering, Pohang University of Science and Technology (M.S.)


POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교