Depth-Guided Dense Dynamic Filtering Network for Bokeh Effect Rendering

Kuldeep Purohit Maitreya Suin Praveen Kandula Rajagopalan Ambasamudram
Indian Institute of Technology Madras, India
kuldeeppurohit3@gmail.com, maitreyasuin21@gmail.com, praveen.kandula94@gmail.com, raju@ee.iitm.ac.in

Abstract

Bokeh effect refers to the soft defocus blur of the background which can be achieved with different aperture and shutter settings in a camera. In this work, we present a learning-based method for rendering such synthetic depth-of-field effect on input bokeh-free images acquired using ordinary monocular cameras. The proposed network is composed of an efficient densely connected encoder-decoder backbone structure with a pyramid pooling module. Our network leverages the task-specific efficacy of joint intensity estimation and dynamic filter synthesis for the spatially-aware blurring process. Since the rendering task requires distinguishing between large foreground and background regions and their relative depth, our network is further guided by pre-trained salient-region segmentation and depth-estimation modules. Experiments on diverse scenes show that our model elegantly introduces the desired effects in the input images, enhancing their aesthetic quality while maintaining a natural appearance. Along with extensive ablation analysis and visualizations to validate its components, the effectiveness of the proposed network is also demonstrated by achieving the second-highest score in the AIM 2019 Bokeh Effect challenge: fidelity track.

1. Introduction

Image manipulation is a key computer vision task, aiming at the restoration of a given image, the filling in of missing information, or the needed transformation and/or manipulation to achieve a desired target (with respect to perceptual quality, contents, or performance of applications utilizing such images). Recent years have witnessed an increased interest from the vision and graphics communities in these fundamental topics of research. This work addresses a manipulation task which has recently gathered attention: Bokeh Effect rendering.

Bokeh effect refers to the soft out-of-focus blur of the background which can be achieved with different aperture and shutter settings in a camera. This is generally achieved during image capture to obtain an aesthetically pleasing perception of a scene. This is unlike the imaging systems that approximate a pinhole camera, where the entire image stays in focus and carries no distinction between the object of interest and the irrelevant regions in the scene. Usually, a single-lens reflex (SLR) camera with a large aperture and certain photography skills are needed to render portraitures.

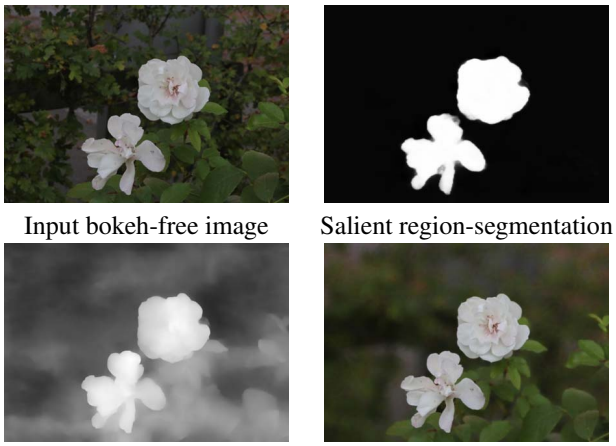


Figure 1. Representative example of bokeh effect rendering on a test image from Bokeh5K dataset.

The portrait mode, which allows users to take DoF photos, is a significant feature of the latest smart phones, e.g., iPhone7+ and Google Pixel 2. Unlike SLR cameras, mobile phone cameras have a small, fixed-size aperture, which generates pictures with everything more or less in focus. Thus, generating DoF effects requires depth-map of the scene, which is usually obtained via specialized hardware in high-end phones. For instance, iPhone 7+ employs dual-lens to estimate depth, while Google Pixel2 uses Phase-Detect Auto-Focus, which can also be regarded as two lenses on the left and right sides.

However, DoF effects by the above systems suffer from several drawbacks due to the usage of specialized hardware. (i) These systems fail or perform poorly if the object of interest is considerably distant from the lenses. (ii) The small baseline of the lenses makes it challenging to estimate depth-values for farther regions, which results in poor DoF

2473-9944/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCVW.2019.00424

rendering. (iii) Most of the commercial systems, except high-end models, do not support such specialized hardware. (iv) These cannot be used for improving images which are already captured using monocular cameras.

In this paper, we propose a deep learning-based solution for DoF effect rendering for the monocular lens, which precludes the usage of any specialized hardware. For the generated image to look natural, the model needs to introduce different levels of blurring in accordance with the depth variations in the scene content. To this end, our model utilizes saliency-segmentation and depth-map estimation modules to guide the rendering process. The segmentation and depth maps are obtained on-the-fly using dedicated pre-trained modules and concatenated with the input image before being propagated through our network. The network follows a densely connected encoder-decoder design wherein the encoder parameters are initialized using a pre-trained classification network. The encoder is connected to two decoder branches among which the first branch of the decoder processes these encoded features using dense upsampling blocks and multi-level pooling to estimate the intensities of the target image. The other decoder branch learns to synthesize dynamic filters to be applied on the input image which fulfills the need of spatially varying processing for the current task. These filters not only depend on the input image but also vary spatially depending on the content. After applying these filters locally on the input image, we add this filtered image with the residual intensities estimated by the first branch to construct the final image with bokeh-effect. A representative result is shown in Fig. 1.

Our contributioutns are:

- We propose a novel automatic model for rendering realistic DoF effect on single bokeh-free images captured using monocular cameras.
- We propose a densely-connected encoder decoder design that leverages the efficacy of joint intensity estimation and dynamic filter synthesis for the spatially-aware blurring process.
- We further employ depth and saliency estimation modules for effectively guiding our fully convolutional network during the rendering process.
- The proposed design exhibits a reasonable balance between parametric efficiency and performance, and achieves the second highest accuracy (based on the PSNR, SSIM scores) in the AIM 2019 Bokeh Effect Challenge [8].

2. Related Works

Segmentation and depth-map guidance: Segmentation and depth maps are used as a guide to improving many restoration tasks in computer vision. [13] proposed a denoising network, where a segmentation network is trained separately. The pre-trained segmentation network is used along with denoising module, allowing segmentation loss to guide the denoising task. An encoder-decoder neural network for image harmonization is proposed in [20]. The features from the encoder are shared by two neural networks one trained for segmentation maps and the other for harmonization result. The features of in intermediate layers of segmentation network are shared with harmonization network, thus guiding harmonization result. An end-to-end network for shadow removal is proposed in [18], where one network is trained for shadow features and the other two networks are used to extract local and global (segmentation map) features. All the above features are used to produce the final shadow-free image. A fish-eye rectification network is proposed in [24] to removal unnecessary distortion from a fish-eye lens. A neural network is trained to give distortion parameters and the other for semantic maps. The above information is used to undistort the image in an end-to-end manner. A deblurring network for out-of-focus images of 3D scenes is proposed in [1], where the network is trained for depth map. Depending on the depth value at different regions in the image, the kernel was inferred in the next step for deconvolution of the blurry input image. Also, a Rolling shutter rectification is proposed in [28], where the network is trained for distortion parameters along with a depth map. A deblurring network for stereo images is proposed in [27], where a network is trained for disparity maps and is subsequently used for deblurring. In [17], the estimated blur probabilities at global and patch-levels are used in a post-processing step to segment the input defocus-blurred image.

DoF Rendering: DoF rendering plays a vital role in real image synthesis as infusing DoF effects helps to improve the depth perception and aesthetic quality of the image. Several works have been proposed for DoF rendering using both conventional and learning-based techniques. 3D cameras are used in [7] to capture RGB and depth map, and the DoF effects are rendered using the depth map. Stereo images are used in [2] to infer depth information and subsequently render DoF effects. Light field rendering [25] and visual abberation [22] use systems that have implicit depth information which helps in rendering DoF effects. However, it is to be noted that most of the real-world images are devoid of any such additional depth sources, making above systems unusable for monocular cameras. Portrait segmentation is used in [19] to render DoF effects. However, their method is suitable only for selfie images and fails to give good rendering effects for generic scenes. A recent approach [23] addresses some of these limitations by designing a CRF-based model guided by depth-estimation network.

Adaptive Convolutions Jia et al. [9] introduced a novel

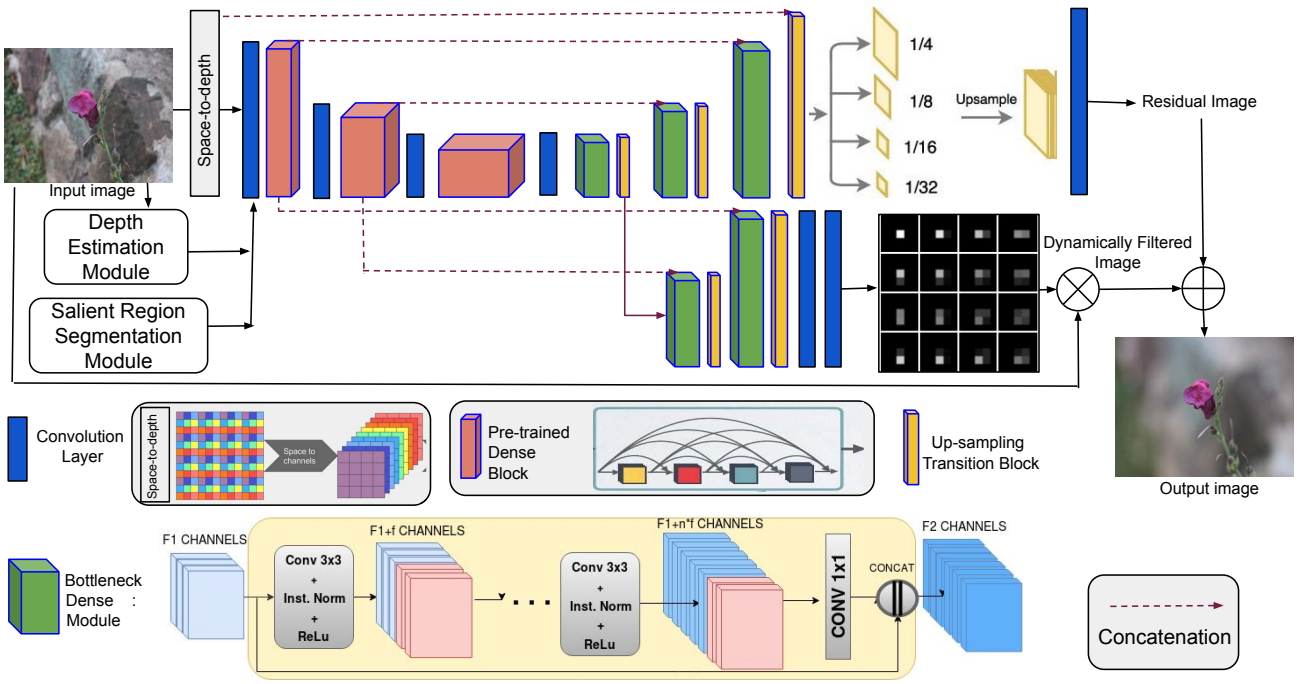


Figure 2. A schematic of our DDDF architecture.

operation of learning global or location-specific dynamic filters. The generated filters are applied to feature-maps and change with input instances. The location-specific filtering is also studied in frame interpolation by Niklaus et al. [14]. Their network produces spatially-adaptive 1-D filters which are applied to a pair of frames to predict the intermediate frame. For the related problem of removing spatially-varying motion blur, [16, 15] presented an adaptive CNN with motion aware dynamic filter-shape adaptation. The video super-resolution method [10] achieved superior results over existing CNN designs using adaptive upsampling convolution. For each location, n^2 kernels are predicted for an up-scaling factor of n.

In this work, we address a relatively unexplored problem of bokeh effect rendering by presenting a depth-guided efficient end-to-end fully convolutional network equipped with image-specific filter synthesis capability and demonstrate its significance in improving performance and interpretability of the rendering process.

2.1. Network Architecture

The proposed network consists of a depth-aware dense encoder-decoder structure with multi-level pyramid pooling and dynamic filtering module for estimating the image with bokeh effect. At the onset, our network utilizes space-to-depth module which divides each input channel in a certain number of blocks and then concatenates those blocks along the channel dimension. Rendering soft out-of-focus blur of the background demands accurate knowledge of scene depth. Motivated by this, we utilize depth and segmentation maps as cues, where depth and segmentation maps are obtained using pre-trained networks for corresponding tasks.

An overview of our Depth-guided Dense Dynamic Filtering Network (DDDF) is shown in Fig. 2. Specifically, we first use off-the-shelf models for single image depth estimation [11] and salient object segmentation [5] to bootstrap our system. The segmentation and depth as appended as guidance maps along with the input image before being fed to our spatially-aware dynamic filtering network (trained using the corresponding DoF result as ground truth) to achieve an efficient and accurate rendering process.

Our encoder is composed of densely connected modules which efficiently address the issue of vanishing gradients and improve feature propagation while substantially reducing the model complexity. We employ a space-to-depth module at the beginning of our encoder. In convolutional operation, the connections are local in space but full along the entire depth of the input volume. Motivated by this, we transform information from pixel space to channel space which increases the receptive field for all the blocks in the encoder. Alongside, this has the added benefit of lowering the computational cost.

Features extracted by a pre-trained deep network are used as powerful image representation in many applica-

tions,such as domain invariant recognition [3], perceptual evaluation [26], and characterizing image statistics [4]. Several mid- and high-level vision tasks like image-segmentation and depth estimation also benefit from using a pre-trained encoder. Since rendering depth-of-field effect also requires the network to understand the scene context, we adopt DenseNet structure for our encoder. The first dense-block contains 12 densely-connected layers, second block contains 16 densely-connected layers and the third contains 24 densely-connected layers. The filters in these layers are initialized using DenseNet-121 network [6] trained on ImageNet dataset. Each layer in a block receives feature maps from all earlier layers, which strengthens the information flow during forward and backward pass making training deeper networks easier. The proposed decoder has two branches where the output of the two branches are added at the end. The first branch has standard densely connected architecture which accepts the features estimated by the encoder at various levels and processes them using residual blocks. As shown in Fig. 2, the decoder contains 3 transition blocks, which increases the spatial resolution through bi-linear up-sampling and convolution. The intermediate features with higher spatial resolution in the decoder are concatenated with the corresponding-sized encoder features. Instance normalization [21] layers are added to the dense blocks in the decoder to normalize the data for increased stability of the network. Empirically, we found that both at training and testing time instance normalization performs better compared to batch normalization.

For fusing information at different scales, we utilize spatial pyramid pooling technique before the final layer. It integrates features at four pyramid levels through pooling and then upsampling. The output of this branch is the residual between the ground-truth image and the dynamically filtered image which we receive from the other branch.

On the second branch we leverage the idea of dynamic filtering [9] due to the necessity of spatially varying operations in synthesizing bokeh effect. Specifically, we generate dynamic blurring filters conditioned on the encoded feature map. These filters are sample and location-specific. The network produces 9×9 filters for each position conditioned on its neighborhood. Each pixel of the dynamically filtered image is created by multiplying these filters locally with the input image patches:

$$I_D(x,y) = \sum_{j=-4}^{+4} \sum_{i=-4}^{+4} F_{x,y}(i+4, j+4) I(x+i, y+j) \quad (1)$$

where $I \in \mathbb{R}^{H \times W}$ and $I_D \in \mathbb{R}^{H \times W}$ are the input image and dynamically filtered image respectively, $F_{x,y}$ is the generated filter. H and W are the height and width of the image whereas x and y denotes the pixel location.

The parameters of the filter-generating decoder are constant during testing but the generated filters are different for each test image. These filters are learned in a self-supervised setting without requiring any manual annotations. The output pixel intensity is a function of the neighboring pixel values within a 9×9 window. Therefore, this adaptive operation based on the estimated per-pixel kernels is less effective for severely blurred regions since the pixel information in the target image is heavily scattered or spread. In such cases, the intensity decoder branch complements the output of the filter decoder branch by providing the required intensities. Outputs from these two decoder branches are added in an element-wise manner to yield the final result.

Table 1. Quantitative comparisons using PSNR (dB) and SSIM scores on images from the AIM 2019 bokeh effect challenge [8].

Rank	Team	Validation Data		Test Data	
		PSNR	SSIM	PSNR	SSIM
1	1^{st} method	25.02	0.89	23.93	0.88
2	**DDDF**	**24.74**	**0.89**	**23.63**	**0.88**
3	3^{rd} method	24.45	0.88	23.43	0.89
4	4^{th} method	24.33	0.88	23.37	0.86
5	5^{th} method	24.01	0.85	23.18	0.89
6	6^{th} method	23.95	0.88	22.90	0.87
7	7^{th} method	23.69	0.86	22.25	0.85
8	8^{th} method	23.64	0.85	22.15	0.84
9	9^{th} method	23.21	0.84	20.88	0.77
10	10^{th} method	22.76	0.84	-	-
11	11^{th} method	22.13	0.77	-	-
12	12^{th} method	21.76	0.79	-	-

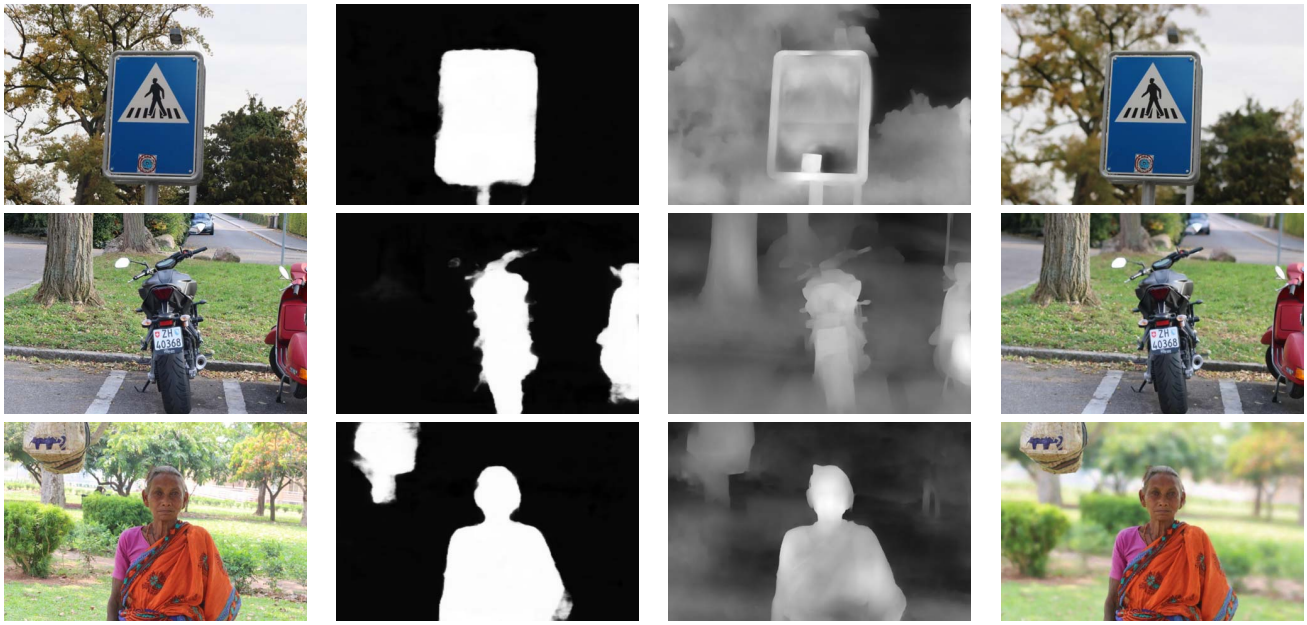
3. Implementation Details

Training and validation data: We utilize a newly released Bokeh5K dataset of newly collected images adapted for the specific goals of the AIM 2019 bokeh effect challenge [8]. The dataset contains Bokeh and Bokeh-free image pairs which are divided into 4694 training, 200 validation and 200 testing pairs. The vertical resolution of images is 1024 pixels and they have a large diversity of contents. The ground-truth images for validation and testing set were unavailable and the scores were obtained using the challenge server.

Training description: L1 reconstruction loss and Adam optimizer with standard settings. Initial learning rate used is 0.0001 with a decay-cycle of 50 epochs. We use a patch-size of 384×384 for training and 640×640 for fine-tuning.

Language and implementation details: Our code is written on Python, and uses standard PyTorch package. We used a system with 2 NVIDIA Titan X GPUs, 256 GB RAM, and Intel Xeon Processor for training. While training with batch-size 8, it uses ≈ 16 GBs of GPU memory.

Training/testing time: We trained our network for approx-



(a) (b) (c) (d)

Figure 3. Results of bokeh effect rendering along with guidance-maps for diverse scenes from validation-set. (a) Input image. (b) Depth map generated using [11]. (c) Saliency map generated using [5], and (d) the result generated using proposed approach.

Figure 4. Results of bokeh effect rendering along with guidance-maps for diverse scenes from test-set. The first row shows the input images while the results generated using our network are shown in the second row.

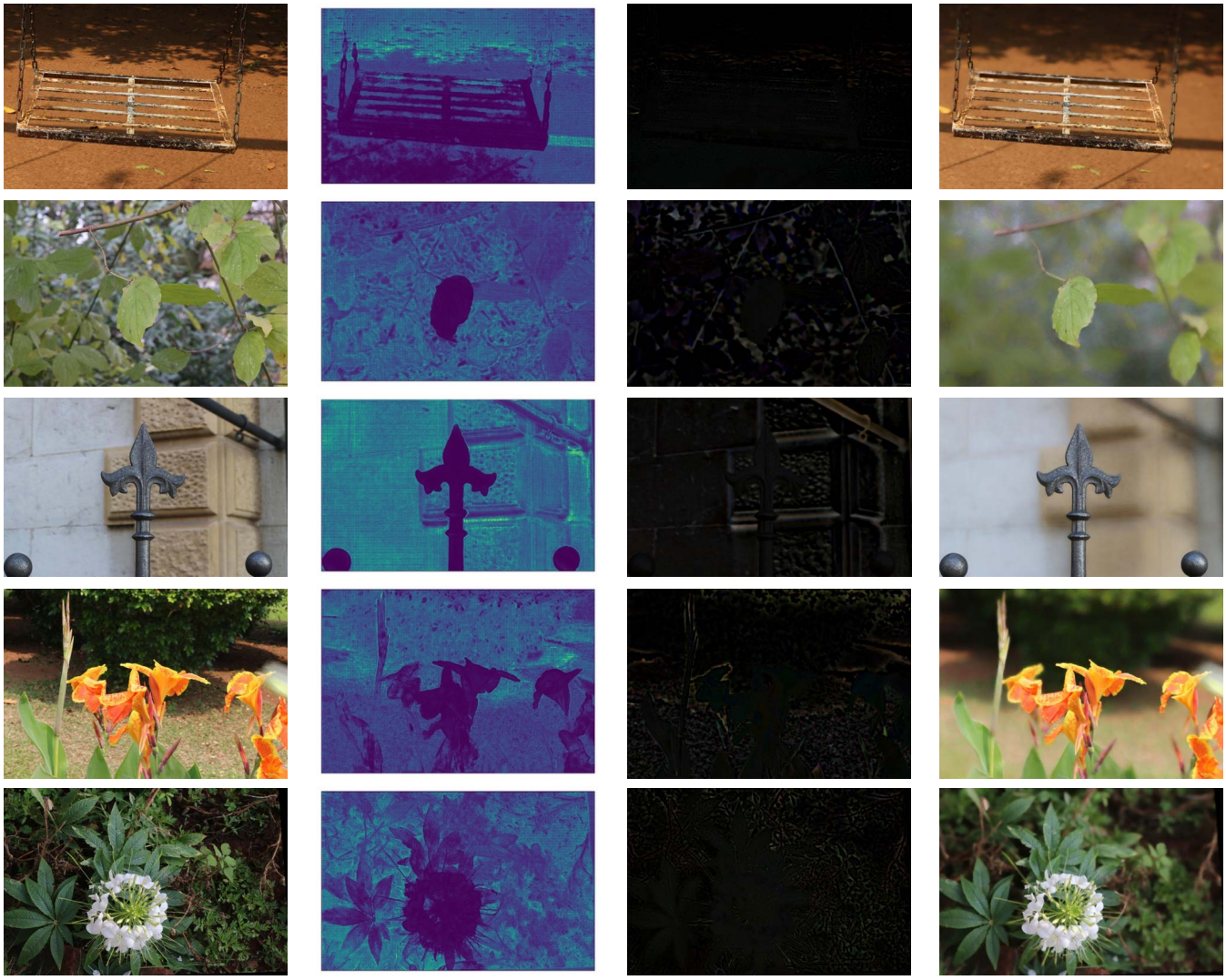
imately 72 hours. While testing, the total time required to process one image is 2.5 seconds.

4. Quantitative results and comparisons on Bokeh 5K Dataset

Our model is primarily proposed for participating in the AIM 2019 Bokeh Effect challenge [8]. The purpose of this challenge is to render synthetic depth-of-field effect in input bokeh-free images. In Track 1: example-based bokeh effect, the target is the fidelity of the output bokeh image results to the ground truth bokeh images. Table 1 shows the performance comparison of the proposed model with models presented by the other participants. As the table illustrates, DDDF achieves the second highest score in the challenge, which validates the efficacy of our proposed framework and establishes its suitability for the task.

5. Qualitative Results and Visualizations

Representative results from validation set are provided in Fig. 3. It can be seen that there is substantial amount of relevant information available in the depth-map and the saliency-based segmentation map. Our network learns to



(a) (b) (c) (d)

Figure 5. Visualisation of our network outputs on diverse scenes from Bokeh5K test-set. (a) shows the input image, (b) spatial distribution of variance calculated from the dynamic filters estimated by the filter-synthesis branch, (c) Result of intensity estimation branch, and (c) Our final result.

Table 2. Quantitative comparison of various versions of our model.

Model	DDDF0	DDDF1	DDDF2	DDDF3	DDDF4	DDDF	DDDF(finetuned)	DDDF(finetuned)$^+$
PSNR	23.67	23.93	24.00	24.30	24.39	24.48	24.70	24.74

utilize them while being robust to the segmentation value errors near the object boundaries. Fig. 4 shows representative results on images from the Bokeh5K test-set. Our network produces perceptually pleasing outputs with natural-appearance.

In Fig. 5, we provide individual outputs of the intensity estimation and filter estimation components of our network. To determine the blurring capability of the estimated 9×9 filters, we calculate the variance of each individual filter and plot its spatial distribution as a map (see Fig. 5(b)). As expected, the variance is higher for the filters acting on the foreground since they mimic a impulse function, and lower for filters in background since they resemble a widely spread gaussian function. It can be visually verified that the filter variance is correlated with the scene depth. The intensity decoder's outputs are shown in Fig. 5(c). We observe that it contains non-zero values for background regions that require large blur. This is expected because at such pix-

(a) Input image (b) Result of [23] (c) Our result

Figure 6. Visual comparison of bokeh effect rendering on a test image from [23]. Our method leads to more potent blurring in the background while preserving sharpness of the foreground regions.

(a) Input image (b) Using iPhone7+ (c) Our result

Figure 7. Visual comparison of bokeh effect rendering on a test image captured using iPhone7+. Our method leads to stronger blurring in the background while preserving sharpness of the foreground regions.

els, input image intensities should spatially spread beyond a 9×9 window, and such intensities are directly supplied as a residual image estimated by this branch.

6. Comparisons with existing techniques

In Fig. 6, we have qualitatively compared our approach with an existing CRF-based algorithm for bokeh-effect rendering [23]. Results on the test image provided in [23] indicate that our model introduces stronger blur in the background, while not affecting the foreground. Similar benefits are visible during comparison with the efect obtained using iPhone7+ (see Fig. 7).

7. Ablation Studies

We analyze the effect of individual components of our network on its training and testing performance. The test scores of various ablations our network are reported in Table 2.

Effectiveness of the dynamic filter-synthesis module: To validate importance of dynamic filter-synthesis, we compare our proposed model DDDF with 3 different versions, featuring different decoder designs while having the same encoder structure. In the version named DDDF4, we merge the filter and residual estimation branches of our network together. Specifically, we remove the layers from our filter decoder and instead, connect the final filter estimation layers to last block of the intensity decoder. This reduces the representational capacity of the network and leads to a notable drop in performance. In version DDDF3, we entirely remove the filter estimation module from DDDF4 and force the network to directly estimate the intensities of the target image. The resulting degradation in performance demonstrates the importance of image-dependent spatially-varying filter learning capability. We also observe that introducing dynamic filter estimation shows higher gains in the early stage of training.

Effectiveness of the backbone structure: In DDDF2, we replace all the bottleneck dense blocks with ordinary bottleneck blocks [6]. The significant difference in PSNR can be attributed to the efficacy of instance-norm based densely-connected blocks and validates our design choice. Finally, to verify the effectiveness of pretraining, we train a similar network DDDF0 wherein the encoder layers are not pre-trained (using DenseNet). The PSNR and loss difference clearly shows that such pre-training leads to better initialization in parameter space and eventually, better convergence.

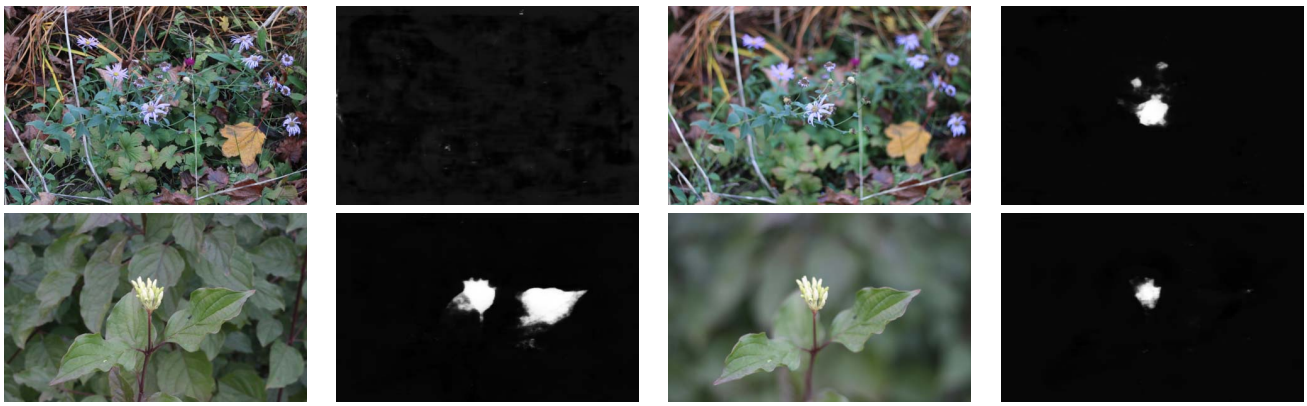
Importance of the guidance-maps We trained another baseline, DDDF1, which does not employ depth-estimation and salient region segmentation modules. The performance difference with respect to DDDF2 indicates the advantage of depth-based guidance.

Effect of training with large patch-size We fine-tuned our final trained network, DDDF at higher patch-size of 640×640 and obtained an improved model DDDF(finetuned). The performance improvement with respect to DDDF in Table 2 is significant and this observation is in agreement with other image restoration tasks. Training with larger patches improves a network's capability to capture contextual information and perform well on high-resolution test images.

Ensembles strategy: In order to maximize the potential performance of our final model, we adopt the self-ensemble strategy [12]. During the test time, we flip and rotate the input image to generate seven augmented inputs. We pass each of the these augmented images through our network, generate corresponding output results, and then further apply inverse transforms which bring them back to the original configuration of the input image. Finally, we perform pixel-wise averaging over the transformed outputs to obtain the self-ensemble result. This method has an advantage over other ensembles as it does not require additional training of separate models. It is beneficial especially when the model size or training time matters.The PSNR and SSIM scores of our model DDDF(finetuned)$^+$ (which utilises self-ensemble strategy) on validation images is 24.74 dB and 0.89, which is a notable improvement over DDDF(finetuned).

(a) Input image (b) Saliency map using [5] (c) Depth map using [11] (d) Our result

Figure 8. A failure case of bokeh effect rendering. Due to presence of a reflective surface on the foreground, the guidance-maps are erroneous, leading to undesired blurring in the foreground.



(a) (b) (c) (d)

Figure 9. An additional application of our bokeh effect rendering model. (a) Input image. (b) Output of [5] using input image. (c) Result of our DDDF on the input image, and (d) Output of [5] using our result.

7.1. Failure cases

Although the depth-map and saliency-map prediction modules in our network provide meaningful guidance during the bokeh-effect rendering process, they are trained on differently captured datasets and hence are prone to generalization issues and yield erroneous guidance-maps on challenging images. An example is shown in Fig. 6, where the foreground contain a reflective surface and hence leads to faulty depth-prediction. The salient-region segmentation network also fails on reflective surfaces and tends to assign higher saliency to people over objects. Thus, our network yields a results with undesired blur in the foreground and insufficient blur in the background. This issue can potentially be addressed by training all the modules on a common dataset.

8. Application: Improving Saliency Prediction

Saliency prediction is directly related to the tasks of discerning the object of interest from the background, it is closely related to bokeh effect rendering process. We found that converting a bokeh-effect into an ordinary image leads to improvement in the performance of saliency detection. A few examples are shown in Fig. 9. Its shows that the outputs of saliency detection network [5] are more accurate when the image contains the Depth-of-Field effect.

9. Conclusions

Bokeh effect refers to the soft out-of-focus blur of the background which can be achieved with different aperture and shutter settings in a camera. In this work, we present a learning-based model for rendering the synthetic depth-of-field effects in input bokeh-free images captured using ordinary cameras. The proposed network is built upon an efficient densely connected encoder-decoder backbone structure with a pyramid pooling module. Since this task requires distinguishing between large foreground and background regions and their relative depth, our network is equipped with pre-trained salient-region segmentation and depth-estimation modules. We significantly improve the network's capacity by introducing a dynamic filter synthesizing module for directly introducing the desired effect in the input image. Exhaustive ablation analysis and visualizations are shown to validate the components and understand their effect on the performance of the proposed network. Our network attained the second highest score in AIM 2019 bokeh effect challenge (Track 1), establishing its superiority for rendering realistic bokeh effect. We believe our spatially-aware design can be utilized for other manipulation tasks as well, and we shall explore them in the future.

References

[1] Saeed Anwar, Zeeshan Hayder, and Fatih Porikli. Depth estimation and blur removal from a single out-of-focus image. In *BMVC*, volume 1, page 2, 2017.

[2] Jonathan T Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4466–4474, 2015.

[3] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[5] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[7] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straßer. Realistic depth blur for images with range data. In *Workshop on Dynamic 3D Imaging*, pages 84–95. Springer, 2009.

[8] Andrey Ignatov, Jagruti Patel, Radu Timofte, et al. Aim 2019 challenge on bokeh effect synthesis: Methods and results. In *ICCV Workshops*, 2019.

[9] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016.

[10] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3224–3232, 2018.

[11] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[13] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S Huang. When image denoising meets high-level vision tasks: A deep learning approach. *arXiv preprint arXiv:1706.04284*, 2017.

[14] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.

[15] Kuldeep Purohit and AN Rajagopalan. Efficient motion deblurring with feature transformation and spatial attention. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4674–4678. IEEE, 2019.

[16] Kuldeep Purohit and AN Rajagopalan. Spatially-adaptive residual networks for efficient image and video deblurring. *arXiv preprint arXiv:1903.11394*, 2019.

[17] Kuldeep Purohit, Anshul B Shah, and AN Rajagopalan. Learning based single image blur detection and segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2202–2206. IEEE, 2018.

[18] Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson WH Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4067–4075, 2017.

[19] Xiaoyong Shen, Aaron Hertzmann, Jiaya Jia, Sylvain Paris, Brian Price, Eli Shechtman, and Ian Sachs. Automatic portrait segmentation for image stylization. In *Computer Graphics Forum*, volume 35, pages 93–102. Wiley Online Library, 2016.

[20] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797, 2017.

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[22] Jiaze Wu, Changwen Zheng, Xiaohui Hu, Yang Wang, and Liqiang Zhang. Realistic rendering of bokeh effect based on optical aberrations. *The Visual Computer*, 26(6-8):555–563, 2010.

[23] Xiangyu Xu, Deqing Sun, Sifei Liu, Wenqi Ren, Yu-Jin Zhang, Ming-Hsuan Yang, and Jian Sun. Rendering portraitures from monocular camera and beyond. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–50, 2018.

[24] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[25] Xuan Yu, Rui Wang, and Jingyi Yu. Real-time depth of field rendering via dynamic light field generation and filtering. In *Computer Graphics Forum*, volume 29, pages 2099–2107. Wiley Online Library, 2010.

[26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[27] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy S Ren. Davanet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10996–11005, 2019.

[28] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4551–4560, 2019.