NAFBET: Bokeh Effect Transformation with Parameter Analysis Block based on NAFNet

Xiangyu Kong, Fan Wang, Dafeng Zhang, Jinlong Wu, Zikun Liu
Samsung Research China - Beijing (SRC-B)
{xiangyu.kong,fan.wang,dfeng.zhang,jinlong.wu,zikun.liu}@samsumg.com

Figure 1. Bokeh effect transformation results with or without PAB. SRMNet [7], Uformer [25] and NAFNet [1] results shown in the left column. The model with PAB results shown in the right column. The model with PAB have better performance.

Abstract

Bokeh effect transformation(BET) aims to transform the bokeh effect of one lens to another lens without harming the sharp foreground regions in the image. Recent studies have shown remarkable success in bokeh effect rendering. However, unlike the traditional bokeh effect rendering task, the BET task needs to transform the image into the bokeh effect of the specified lens. The existing bokeh rendering method is invalid or inefficient for BET, because each pair of lens needs to independently build different model. To address this limitation, we propose NAFBET, a scalable approach than can perform bokeh rendering for multiple lens using only a single model. NAFBET is based on the structure of the image restoration model NAFNet and expands it by adding the source and target parameter analysis block(PAB) to adapt to the BET task. This block can be very convenient to apply in UNet-based model, which can greatly improve BET performance. We did a lot of experiments to prove the effectiveness of our method. In particular, NAFBET won the 1st place in the NTIRE 2023 Bokeh effect transformation Challenge.

1. Introduction

The bokeh effect is a very popular photography technique used to make the foreground regions sharp and the background blurred. For SLR cameras, the bokeh effect can be obtained by adjusting the aperture of the lens, the distance to the object, and the focal length of the lens. Different lenses have different bokeh effects. Due to the limitations, the mobile cameras cannot generate the same bokeh as SLR camera. But It can use computational methods to enhance image quality [6] or simulate bokeh effect [16, 19] by optimizing the sharp areas and artificially blurring the background.

Synthetic bokeh effect rendering is a popular vision task in recent years. But the current bokeh rendering task [9, 17] does not take into account the variation bokeh effect between different lenses. NTIRE 2023 [5] hold the first challenge of BET and provide the dataset and benchmark. Different from traditional bokeh rendering, BET aims to convert the bokeh effect of one lens to the bokeh effect of another lens without harming the sharp foreground regions in the image. In other words, for an input image of a known lens type, bokeh effects of different lens types can be generated. Through this task we can transform the image into the any type bokeh effect we want from a certain type of lens. Even convert large aperture images into small aperture image effects, which is different from the current task of bokeh rendering, as shown in the second line in Fig. 4.

In summary, BET is more like an image-to-image translation task. However, there are some differences between them. Image translation can convert images to target domain features, such as converting the gender and age of people [2]. The extra parameters of most current image translation models are only target domain parameters, they do not consider the features of the input image. But for the BET task, the lens parameters of the input image have practical significance. It not only represents the real bokeh effect degree, but also means whether the bokeh transformation target is to render the bokeh effect or reconstruct the image, as shown in Fig. 4. we did experiments to verify this view in Sec. 4.2. Only use the target domain label can not achieve good performance. Therefore, the current image translation model cannot meet the needs of BET tasks.

The BET task is very meaningful for pursuing real bokeh effects and simulating the bokeh effects of different cameras and different lenses. Recent studies have shown remarkable success in traditional bokeh effect rendering and image-to-image translation. However, existing models are ineffective or inefficient for BET task. For the bokeh rendering model, in order to learn all transformations between n different lens, $n(n-1)$ models must be trained. At the same time, image transformation models are ineffective, they do not take into account the current degree of bokeh effect and the direction of transformation.

As a solution to such problems we propose NAFBET. Instead of learning fixed transformations, our model accepts both source image and source-target lens information as input, and learns to flexibly transform the source image to different lens bokeh effects.

Overall, our contributions are as follows:

- We analyze the deficiencies of existing bokeh and image translation methods to solve the BET task, and propose NAFBET. It inherits the advantages of NAFNet in the field of image restoration, and uses the characteristics of the BET task to improve it through the source image lens and the target lens parameter analysis block.
- We design the parameter analysis block of adding the source or target parameters and verify the effectiveness of the method and insertion position.
- Extensive experiments are conducted to demonstrate the effectiveness of our proposed NAFBET. We won the 1st place in the NTIRE 2023 Bokeh Effect Transformation Challenge [5] with NAFBET.

2. Related Works

2.1. Synthetic Bokeh Effect Rendering

Synthetic bokeh effect rendering task aims to highlight the main subject of the image and blur the background to improve the visual effect.

Early studies [22, 23] did not consider the relationship between the depth of field and the degree of blur, and only used methods such as Gaussian blur to blur the background uniformly. Later, some researchers [15, 18, 24] used the depth estimation method to use the depth of field as a parameter of the bokeh effect, and performed blurring processing for different depths of field. Although these methods consider the depth of field, they are still different from the real bokeh images taken by SLR cameras. Therefore, many researchers have used real bokeh datasets [10] to develop many models [11] hoping to achieve the real bokeh effect. The bokeh effect generated by this way depends entirely on the type of training dataset. If you want to generate different bokeh effect of different lens parameters, you must train multiple models. In this paper, our proposed method addresses this challenge.

2.2. Image-to-image Translation

Image-to-image translation task aims to change some features of a given image, for example, changing a person's hair from black to brown or changing the seasons from summer to autumn.

The initial research is for the single domain translation, such as pix2pix [12], learning the mapping from input to output image, such as labels to street scene, day to night and edges to photo. After the remarkable success of image-to-image translation in two domains, many researchers proposed multi-domain image-to-image translation. StarGAN [2] is the first method applied to multiple domains, it learns the mapping between multiple domains, and proposes a mask vector method, which enables the model can control all available domain labels. StarGAN v2 [3] did further research on the diversity of image transformations. Mean-

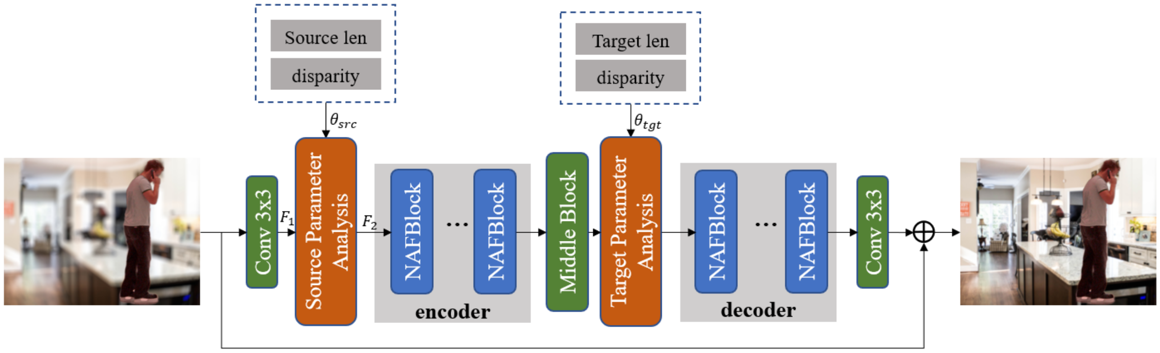
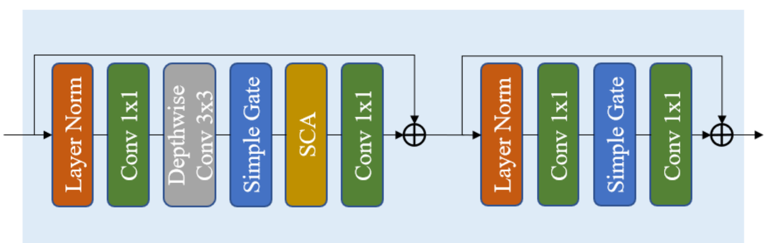
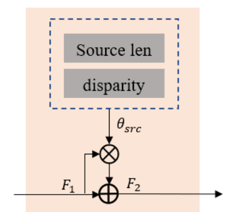


Figure 2. Overall architecture of NAFBET. The input of the model is the source image, source and target lens, and disparity. NAFBlock shown in Fig. 3a. Source Parameter Analysis Block shown in Fig. 3b. The Middle Block consists of the NAFBlock.



(a) NAFBlock.



(b) Source Parameter Analysis Block.

Figure 3. NAFBlock and Source Parameter Analysis Block.

while, in order to solve the problem of data pairs, many unsupervised methods [8, 14] have also been proposed. However, image translation only cares about the parameters of the target domain. Different from the above models, the BET task needs to care about the lens parameters of the input and output domains in order to transform to the bokeh effect of the target lens. We also tried to use only the target domain labels shown in Tab. 3, but it cannot achieve good performance in BET.

3. Methodology

3.1. Bokeh Effect Transformation

Our goal is to train a single model M that learns different bokeh effect mappings among multiple lens. To achieve this, we train a model M to transform the input image x into an output image y conditioned on the source label θ_{src} and target label θ_{tgt}.

$$M(x, \theta_{src}, \theta_{tgt}) \rightarrow y \tag{1}$$

$$\theta_{src} : \{len_{src}, dis\}, \theta_{tgt} : \{len_{tgt}, dis\} \tag{2}$$

where θ_{src} and θ_{tgt} both contain two parameters len and dis. len_{src} and len_{tgt} is the source and target lens. dis is the disparity value which is a broad metric for inverse distance as an indicator for blur strength, i.e. the higher the disparity, the more blurry the image is. It was calibrated by the data collector inspired by previous work MiDaS [20].

Overall Framework. We designed our model based on the NAFNet [1], the overall framework is shown in Fig. 2. Our model is in the form of UNET, the encoder and decoder are composed of NAFBlock [1], as shown in Fig. 3a. The NAFBlock is based on point-wise and depth-wise convolution with channel attention without nonlinear activation functions. The *Simple Gate* operation is to split the input feature along channel dimension and then multiply with element-wise.

The input of the model is the source image and the source-target parameters. First, use 3×3 convolution to map the input image to the high dimensional feature space F_1. Then, the source parameter analysis block fuses the source parameters and the high dimensional feature F_1 as the encoder's input F_2:

$$F_2 = PAB(F_1, \theta_{src}) \tag{3}$$

And the input of the decoder is similar to the encoder. The input of the decoder not only has the output of the encoder, but also the lens parameters of the target image θ_{src}.

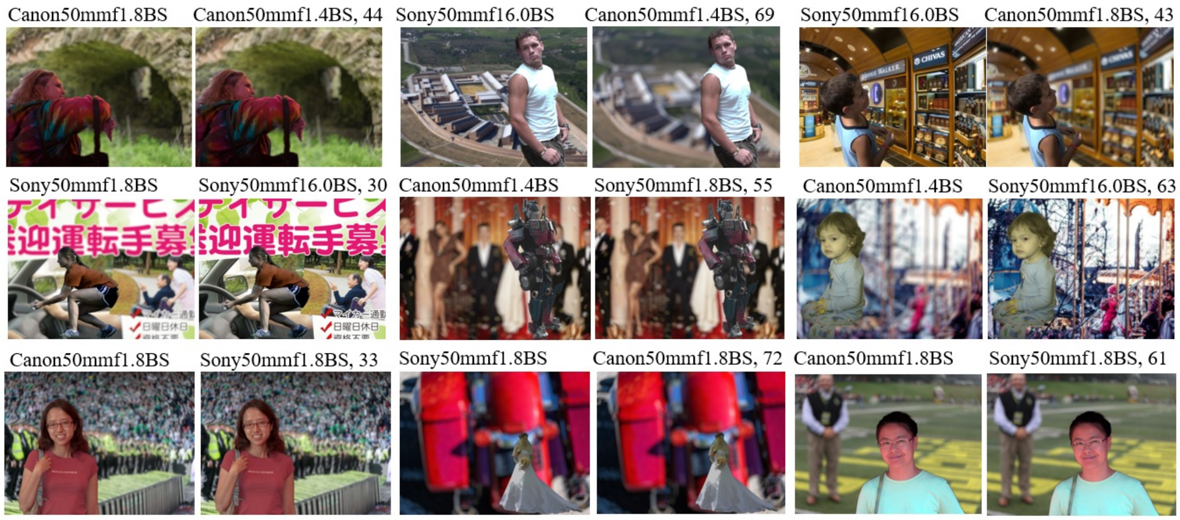


Figure 4. Bokeh effect transformation. Synthetic Data Example. Each pair of images has the source and target lens label and disparity. The first line shows the bokeh effect transformation of the same or different cameras, from small aperture to large aperture mode. This bokeh effect is similar to the traditional synthetic bokeh effect rendering task. The second line is the opposite process of the first line. This effect is more like image restoration tasks. The third line is the bokeh effect transformation between different cameras.

The decoder combines the target label θ_{tgt} to generate the real bokeh effect.

Source and Target Parameter Analysis Block. Unlike the conventional adding parameters method, we make full use of the encoder and decoder to analyze the source and target label. The position of adding the source and target label to convolutions is very important. The source image lens parameters should be incorporated into the encoding process and the lens parameter of the target image should together with the decoder to transform the bokeh effect. We adds the source and target label at the front of the encoder and decoder. The details of the proposed Parameter Analysis Block(PAB) is shown in Fig. 3b.

According to the number of lens types in the dataset, we first encoded the lens type:

$$len = \begin{cases} 0, \text{ if len type} = Canon50mmf1.4BS \\ 1, \text{ if len type} = Canon50mmf1.8BS \\ ... \\ n \end{cases} \tag{4}$$

And then balance the lens coding value len and disparity value dis through α. Finally, the parameters adding degree is adjusted by β. Take the input of the encoder as an example, the insertion method is:

$$F_2 = F_1 + \theta_{src} F_1 \tag{5}$$

$$\theta_{src} = \beta(\alpha len_{src} + dis) \tag{6}$$

where F_1 is the high dimensional feature of the source image extracted by 3 × 3 convolution.

LOSS. We only used L1 loss. But we regulated the foreground and background areas, in order to make the bokeh effect and transitions more realistic:

$$\mathcal{L} = (\gamma - Mask) \left\| y - y_{tgt} \right\|_1 \tag{7}$$

where y represent the bokeh image transformed by model, and y_{tgt} represent the ground-truth target bokeh effect image. $Mask$ is the foreground mask to distinguish the backgrounds. γ is the regulatory parameter.

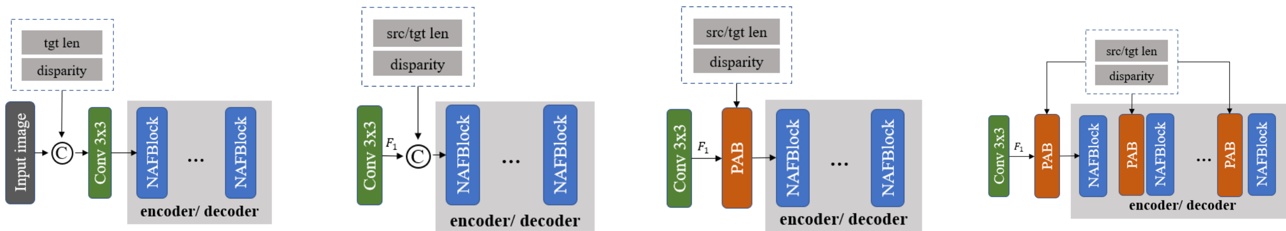
3.2. Training Strategies

We adopted the method of exchange sources and target images for data enhancement and random flipping and rotation. At the same time, in order to prevent overfitting and improve model performance, we have adopted DropPath [13] strategy and local base [4].

4. Experiments

4.1. Dataset

The Bokeh Effect Transformation Dataset(BETD) contains 20000 image pairs provided by NTIRE 2023 Bokeh Effect Transformation Challenge [5]. Each image pair consists of source, target and alpha image. Alpha image is the mask to distinguish the foreground and the background, as shown in the second column in Fig. 7. It is expressed as $Mask$ in Eq. (7). And the target image is artificially blurred with different lenses. Each paird source and target image has lens and used disparity parameters. There are four types



(a) Image-to-image method.

(b) Concat before en-decoder.

(c) The method we proposed.

(d) Add PAB to each layer.

Figure 5. Different parameter adding methods in the ablation study. (a) was set to use image-to-image translation method to concat the only target label to input image. (b) was set to concat the source or target label before encoder or decoder. (c) was set to the method we proposed at Eq. (5), add the PAB before encoder and decoder. In order to prove the correctness of the position, we also set up (d), add PAB to each layer.

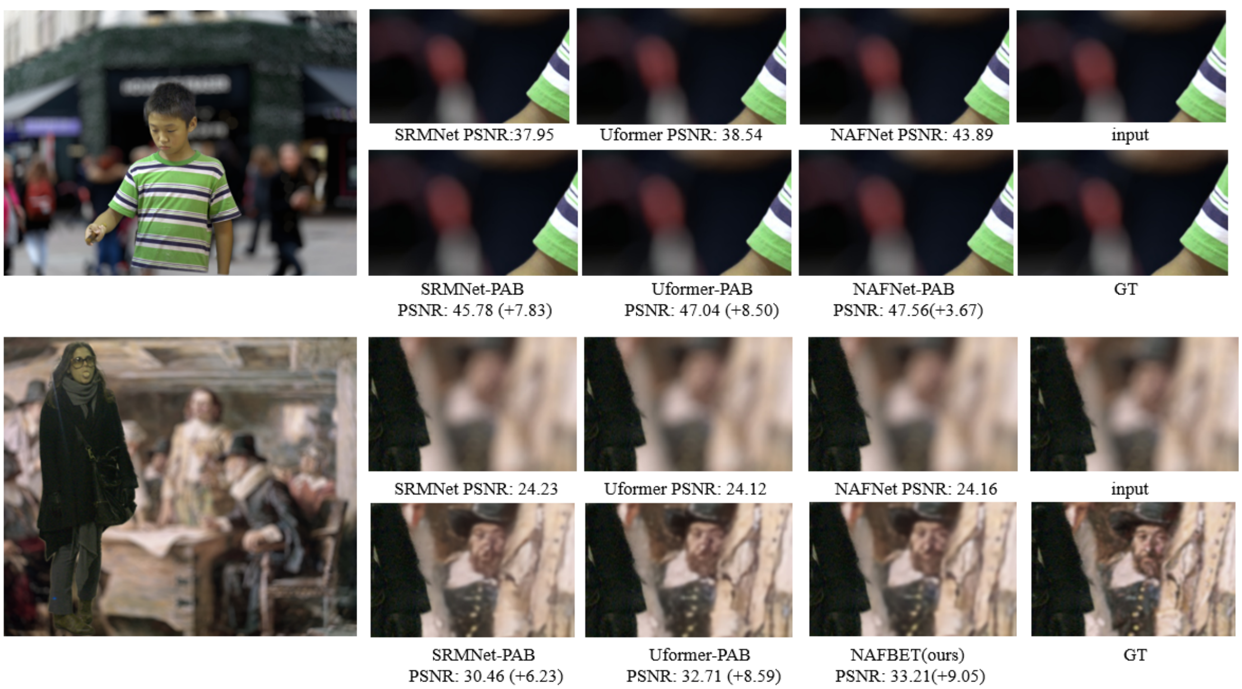


Figure 6. Bokeh effect transformation results with different baseline model on BETD [5]. Models without PAB cannot learn the bokeh mapping well between different lenses.

of lens in dataset, so the coding of the lens type is:

$$len = \begin{cases} 0, \text{ if len type} = Canon50mmf1.4BS \\ 1, \text{ if len type} = Canon50mmf1.8BS \\ 2, \text{ if len type} = Sony50mmf1.8BS \\ 3, \text{ if len type} = Sony50mmf16BS \end{cases} \quad (8)$$

In the training dataset, *disparity* parameters are integer between 30 and 80. Some images of the dataset are shown in Fig. 4. Each image has a lens label and a disparity value. Fig. 4 shows the different bokeh effects from small aperture to the large, from large aperture to small, and the transform between different cameras. The effect of the first line is similar to the traditional bokeh task, but the effect of the second line is more like image restoration task.

4.2. Ablation Study

Implementations Details. All models are trained for 5×10^5 iterations in total with a batch size of 4 by L1 loss function with AdamW optimizer. In each training batch, each paired images are randomly cropped to 512×512 and augmented by random flipping and rotation. The learning rate is initialized as 3×10^{-5}. And all models are implemented by Pytorch and trained with 1 A100 GPU. The PAB parameters were set to $\alpha = 100$, $\beta = 1/500$.

We use the BETD training [5]. In detail, We randomly

parameter	PSNR↑
$\gamma = 1.0$	42.01
$\gamma = 1.2$	42.24
$\gamma = 1.4$	42.11

Table 1. Bokeh effect transformation results with different regulatory Loss parameter γ. ↑ represents the larger the value, the model the better.

selected 500 images as validation sets, and the remaining 19500 images were used as training sets. All ablation study models are used the same training and validation dataset.

Impact of regulatory Loss parameter. The regulatory parameter γ as described in Sec. 3.1 is a important parameter. In this experiment, We set this parameter as 1, 1.2 and 1.4. The result is shown in Tab. 1. When $\gamma = 1.0$, L1 loss only act in the background area. When $\gamma = 1.2$, the weight of the background is 1.2, and the foreground is 0.2. It can be seen through the comparison results that although the bokeh effect is acting on the background area, the foreground areas needs to be considered for a better boundary transition effect.

Impact of Source and Target Parameter Analysis Block. In Sec. 3.1, we proposed the parameter insertion method. In this experiment, we take the only NAFNet without source and target labels as the baseline(first line in Tab. 2) to investigate the impact of the insertion method and location. In detail, we believe that the source image lens parameters should be incorporated into the encoding process. At the same time, the lens parameter of the target image should together with the decoder to transform the bokeh effect. In order to verify our views, we conducted a set of comparison tests. For details, please refer to Fig. 5. The first group shown in Fig. 5a was set to use image-to-image translation method to concat the target label to the channel dimension of the input image. The second group shown in Fig. 5b was concated the source label before encoder and target label before decoder. The result shown in the third line in Tab. 2. The third group was set to the method we proposed at Eq. (5), add PAB before the encoder and decoder shown in Fig. 5c. In order to prove the correctness of the position, we also set up a set of tests to add parameters to each layer, shown in Fig. 5d.

As demonstrated by the results in Tab. 2, our method has significant performance improvements compared to others. Compared with the baseline model, our method has improved 4.83db. And compared with the image-to-image translation method(NAFNet-1), our method has improved 3.90db. Compared with our method, the PSNR that adds parameters to each layer decreased by 0.66dB. These results indicate that the method and position of the parameter insertion is very important.

4.3. Comparison to state-of-the-arts methods

Because this is the first bokeh effect transformation task, there is no previous related work to compare. We selected the UNet-based SOTA models in the field of image restoration as the baseline model for testing.

Baseline Models. We adopt SRMNet [7], Uformer [25] and NAFNet [1] as baseline models, both of which adopt the UNet structure and perform well in image restoration. SRMNet and NAFNet are CNN-based. Uformer is transformer-based. All models adopt two parameter adding methods. One is the standard model without any lens parameter. The other is the PAB method we proposed in Sec. 3.1.

Implementations Details We used the training dataset provide by NTIRE 2023 Bokeh Effect Transformation Challenge. Different from the ablation study, the test results of all models were tested on the online verification set provided by the competition. All the models were trained for 5×10^5 iterations in total with a batch size of 3. And the regulatory parameter γ was set to 1.2. The other is the same as ablation study.

Quantitative Evaluations The quantitative comparisons with baseline methods are shown in Tab. 3. The model with PAB method is much better. Compared to the baseline method, the SRMNet, Uformer and NAFNet model with PAB has been improved 3.48db, 4.43db and 4.98db. It fully illustrates the effectiveness of the PAB.

Visual Results. Fig. 6 shows the visual comparisons for different baseline models. The model with PAB have better PSNR and visual effect. These images indicate that our PAB plays a very important role in BET task between different lenses. In contrast, models without PAB can't learn the bokeh mapping well between different lenses. Fig. 7 shows the bokeh transform effect with NAFBET. The first line show the bokeh transformation from small to large aperture effects. Last two lines show the bokeh transformation from large to small aperture effects. Fig. 8 shows the bokeh transform effect with NAFBET between different lens types. The image in the red box is the source image, and the other images of the same line are bokeh images transformed by NAFBET.

4.4. NTIRE 2023 Bohek Effect Transformation Challenge Result

Implementations Details The NAFBET model used in Challenge has 16, 8, 12 NAFBlocks for encoder, decoder and Middle Block, in which the feature channels was set to 64. We adopted random flipping and rotation for data enhancement. The source and target lens parameter analysis module was set to $\alpha = 100$, $\beta = 1/500$. The regulatory parameter γ was set to 1.2. The DropPath rate was set to 0.3. And the model was trained with AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ by L1 loss function. The details

model	label	method	podition	PSNR↑	△PSNR
NAFNet-baseline	-	-	-	37.41	
NAFNet-1	target	*concat*	*input*	38.34	+0.93
NAFNet-2	source and target	*concat*	*before en/decoder*	41.01	+3.6
NAFNet-3	source and target	*PAB*	*each layer*	41.58	+4.17
NAFBET(ours)	source and target	*PAB*	*before en/decoder*	42.24	+4.83

Table 2. Bokeh effect transformation results with different parameter insertion method on BETD [5]. *concat* represent concat the label to the feature on channel dimension. *input* represent concat the target label to the input. *before en/decoder* represent add the label before the encoder and decoder. *each layer* represent add the source or target label to each layer of encoder or decoder. The model setting of first line is the NAFNet without any lens parameters. The second to the forth line's model setting are shown in Fig. 5a, Fig. 5b and Fig. 5d.

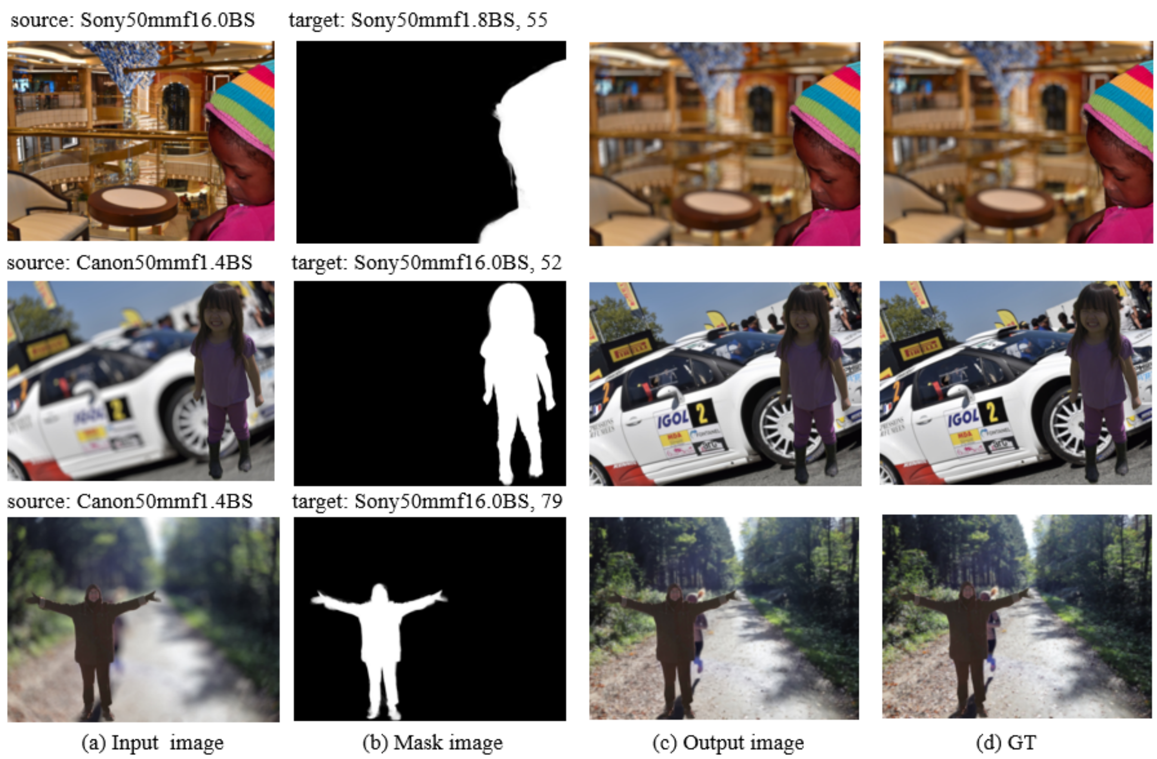
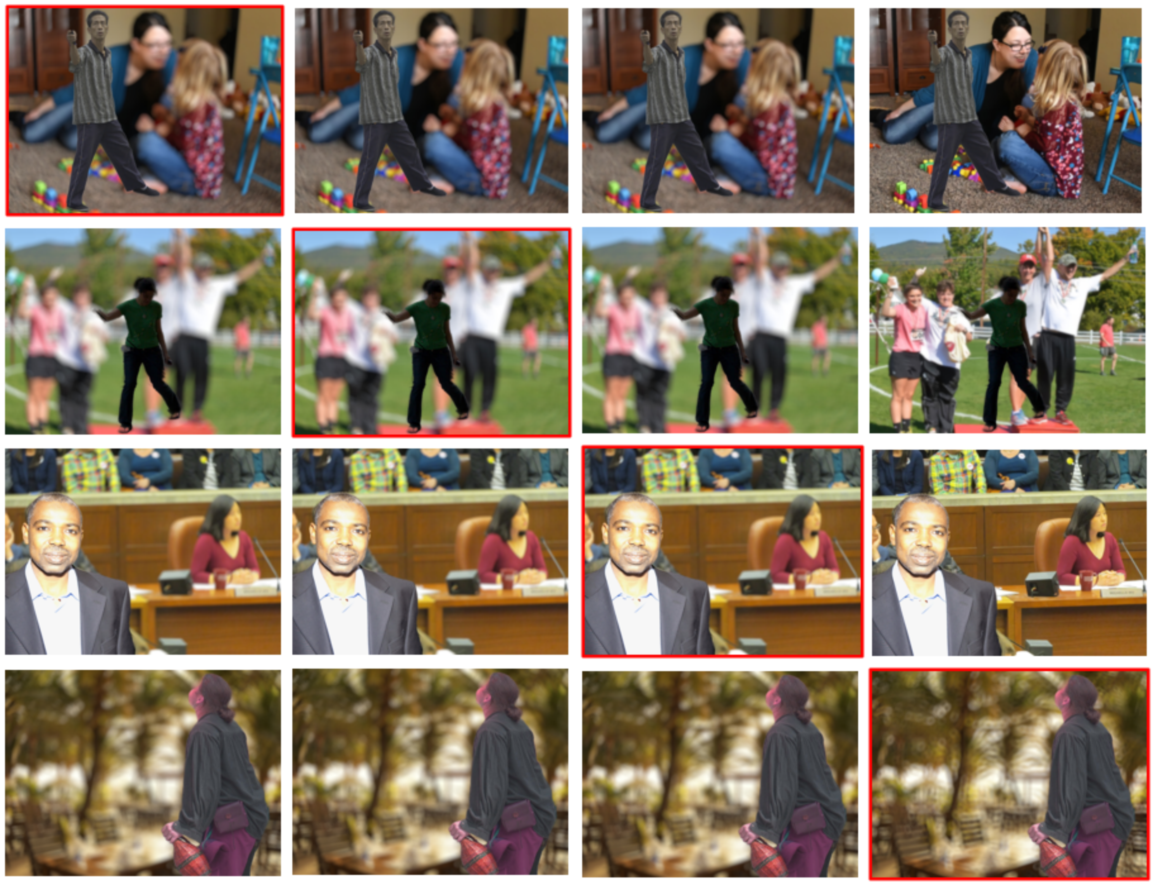


Figure 7. Small to large and large to small aperture bokeh effect transformation results with NAFBET on BETD [5]. The first column are the source images, the second column are Mask images, the third column are the bokeh images generated by NAFBET, the fourth column are ground truth images. The lens parameters and disparity of the source and target image are marked above the image.

of training steps are as follows:

(1) In the first step, the model was trained with 512×512 image patches, and the learning rate is initialized set as 3×10^{-5} with cosine learning rate decay. The network is trained for 5×10^5 iterations in total. (2) In the second step, the model was initialized with the pretrained weights in step 1 and trained with 512×512 image patches. The learning rate is initialized set as 2×10^{-5}. The network is trained for 2×10^6 iterations in total. And we added the data enhancement method of exchange source and target images. (3) In the third step, the model was initialized with the pretrained weights in step 2 and trained with 640×640 image patches. The learning rate is initialized set as 1×10^{-5}. The network is trained for 5×10^5 iterations in total. Also added the data enhancement method of exchange source and target images.

Result. NAFBET has achieved the best quantitative result in the NTIRE 2023 Bohek Effect Transformation Challenge [5]. Our model finally reached 35.264dB in the real test dataset.The challenge results shown in Tab. 4. Our result was 0.692dB higher than the second place and 0.721dB higher than baseline mothed [21]. It proves the effectiveness of the method we propose and provides a good foun-



(a) Canon50mmf1.4BS (b) Canon50mmf1.8B (c)Sony50mmf1.8BS (d)Sony50mmf16.0BS

Figure 8. The bokeh transform effect with NAFBET between different lens types on BETD [5]. The image in the red box is the source image, and the other images of the same line are bokeh images transformed by NAFBET.

Baseline Model	label	method	PSNR↑
SRMNet	-	-	37.14
SRMNet	source and target	*PAB*	40.62(+3.48)
Uformer	-	-	37.29
Uformer	source and target	*PAB*	41.72(+4.43)
NAFNet	-	-	37.11
NAFNet(ours)	source and target	*PAB*	42.09(+4.98)

Table 3. The quantitative comparisons with baseline methods with different parameter insertion method on the online verification set. *PAB* represent the parameter analysis block we proposed.

Team	PSNR↑	SSIM↑
Samsung Research China-Beijing(ours)	35.264	0.9362
AIA-Smart	34.572	0.9361
NUS-LV-Bokeh	32.326	0.9333
baseline [21]	34.543	0.9350

Table 4. NTIRE 2023 Bokeh effect transformation Challenge results.

dation for future research.

5. Conclusion

In this paper, we proposed NAFBET, a model that transform the bokeh effect between multiple lenses. The parameter analysis block we designed can make full use of the encoder and decoder processing source and target lens label. This block can be very convenient to apply in UNet-based model, which can greatly improve BET performance. A large number of experiments have proved the effectiveness of this block. We hope that our work provide new ideas for bokeh effect transformation or even image-to-image translation tasks.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 17–33. Springer, 2022. 1, 3, 6

[2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 2

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 2

[4] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 53–71. Springer, 2022. 4

[5] Marcos V Conde, Manuel Kolmet, Tim Seizinger, Tom E Bishop, and Radu Timofte. Lens-to-lens bokeh effect transformation. ntire 2023 challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023. 2, 4, 5, 7, 8

[6] Marcos V Conde, Florin Vasluianu, Javier Vazquez-Corral, and Radu Timofte. Perceptual image enhancement for smartphone real-time applications. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1848–1858, 2023. 2

[7] Chi-Mao Fan, Tsung-Jung Liu, Kuan-Hsien Liu, and Ching-Hsiang Chiu. Selective residual m-net for real image denoising. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 469–473, 2022. 1, 6

[8] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 3

[9] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1676–1686, 2020. 2

[10] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 418–419, 2020. 2

[11] Andrey Ignatov, Jagruti Patel, Radu Timofte, Bolun Zheng, Xin Ye, Li Huang, Xiang Tian, Saikat Dutta, Kuldeep Purohit, Praveen Kandula, et al. Aim 2019 challenge on bokeh effect synthesis: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3591–3598. IEEE, 2019. 2

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[13] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016. 4

[14] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 3

[15] Chenchi Luo, Yingmao Li, Kaimo Lin, George Chen, Seok-Jun Lee, Jihwan Choi, Youngjun Francis Yoo, and Michael O Polley. Wavelet synthesis net for disparity estimation to synthesize dslr calibre bokeh effect on smartphones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2407–2415, 2020. 2

[16] Xianrui Luo, Juewen Peng, Ke Xian, Zijin Wu, and Zhiguo Cao. Bokeh rendering from defocus estimation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 245–261. Springer, 2020. 2

[17] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16283–16292, 2022. 2

[18] Kuldeep Purohit, Maitreya Suin, Praveen Kandula, and Rajagopalan Ambasamudram. Depth-guided dense dynamic filtering network for bokeh effect rendering. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3417–3426. IEEE, 2019. 2

[19] Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, and Jian Cheng. Bggan: Bokeh-glass generative adversarial network for rendering realistic bokeh. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 229–244. Springer, 2020. 2

[20] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3

[21] Tim Seizinger, Marcos V Conde, Manuel Kolmet, Tom E Bishop, and Radu Timofte. Efficient multi-lens bokeh effect rendering and transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023. 7, 8

[22] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. In *John Wiley Sons, Ltd.*, pages 93–102, 2016. 2

[23] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 92–107. Springer, 2016. 2

[24] Fan Wang, Yingjie Zhang, Yibo Ai, and Weidong Zhang. Rendering natural bokeh effects based on depth estimation to

improve the aesthetic ability of machine vision. *Machines*, 10(5):286, 2022. 2

[25] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 1, 6