Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
17/2 (2024), 175-183. DOI: http://dx.doi.org/10.21609/jiki.v17i2.1274

Deep Image Deblurring for Non-Uniform Blur: a Comparative Study of Restormer and BANet

Made Prastha Nugraha, Laksmita Rahadianti

Faculty of Computer Science, University of Indonesia, Depok 16424, Indonesia

E-mail: made.prastha@ui.ac.id

Abstract

Image blur is one of the common degradations on an image. The blur that occurs on the captured images is sometimes non-uniform, with different levels of blur in different areas of the image. In recent years, most deblurring methods have been deep learning-based. These methods model deblurring as an image-to-image translation problem, treating images globally. This may result in poor performance when handling non-uniform blur in images. Therefore, in this paper, the author compared two state-of-the-art supervised deep learning methods for deblurring and restoration, e.g. BANet and Restormer, with a special focus on the non-uniform blur. The GOPRO training dataset, which is also used in various studies as a benchmark, was used to train the models. The trained models were then tested on the GOPRO testing test, the HIDE testing set for cross-dataset testing, and GOPRO-NU, which consists of specifically selected non-uniform blurred images from the GOPRO testing set, for the non-uniform deblur testing. On the GOPRO testing set, Restormer achieved an SSIM of 0.891 and PSNR of 27.66 while BANet obtained an SSIM of 0.926 and PSNR of 34.90. Meanwhile, for the HIDE dataset, Restormer achieved an SSIM of 0.907 and PSNR of 27.93 while BANet obtained an SSIM of 0.908 and PSNR of 34.52. Finally, on the non-uniform blur GOPRO dataset, Restormer achieved an SSIM of 0.911 and PSNR of 29.48 while BANet obtained an SSIM of 0.935 and PSNR of 35.47. Overall, BANet shows the best result in handling non-uniform blur with a significant improvement over Restormer.

Keywords: *deep learning, blur attention, restormer, image deblurring, non-uniform*

1. Introduction

Blurring effects in images occur quite often in image acquisition. These effects usually occur during image capture with long exposure times or when using only one lens to capture different types of scenes. Blurred images result in lower visibility in the image, making the image more difficult to understand and interpret. Thus, is desirable to restore these blurred images to a more visible version. In recent years, the development of camera technology on mobile devices also increases the need to maintain the quality of captured images and reduce blur. This restoration process of blurred images, or deblurring, intends to regain the original sharp image form of a blurring images [1]. An image with better visibility is crucial to most computer vision, such as object detection, face recognition, image classification, or object tracking [2].

The blurring effects in images can be categorized based on the condition in which the images are captured, some of which are shown in Figure 1. Motion blur, shown in Figure 1 (a) is caused by the movement by the device during image capture [3]; gaussian blur, shown in Figure 1 (b), usually appears due to low light imaging [4]; and defocus blur (Figure (1 (c)) which occurs when an object is outside of the camera focus point [5].

Furthermore, blurring can also be defined based on the location distribution of blur effects, i.e., uniform and non-uniform blur. Uniform blur (Figure 1 (d)) describes blur when it occurs with

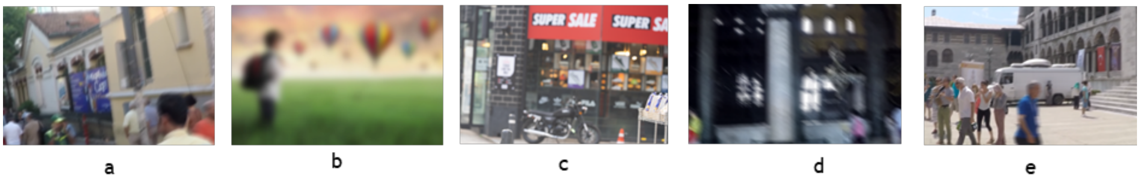

Figure 1. Various type of blur effects on images. (a) motion blur, (b) gaussian blur, (c) defocus blur, (d) uniform blur, and (e) non-uniform blur

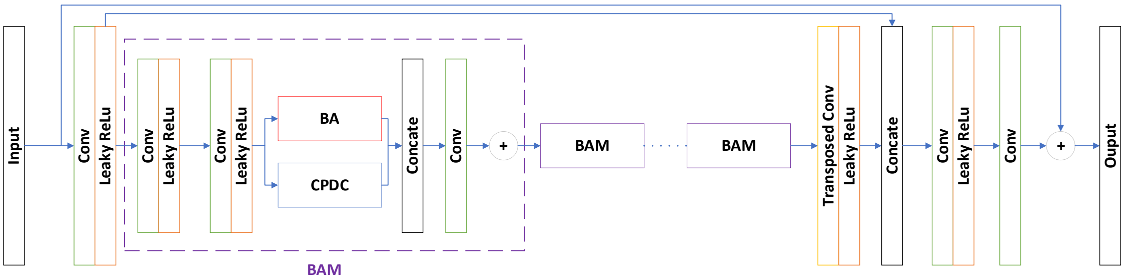

Figure 2. The architecture of BANet, which uses the U-Net [14] architecture with attention modules [8].

the same properties and magnitude across all parts of the image. For non-uniform blur, as shown in Figure 1 (e), different properties and magnitudes of blurring effects are present in various areas of one image. Uniform blur is usually assumed in most deblurring research to simplify the problem at hand. Meanwhile, non-uniform blur is more common in daily life, but the requirement to solve this problem is significantly higher in complexity, the research conducted to tackle non-uniform blur is significantly less than the uniform blur [6]. Restormer [7], is one of the state-of-the-art deep learning models proposed for generic image restoration, including deblurring. Even so, Restormer is still lacking at handling non-uniform blur problems. Restormer is able to enhance the visibility and reduce the blur on most images, but it still leaves some dynamic blur in various parts of the images. On the other hand, BANet [8] is an image deblurring method that focuses on non-uniform blur restoration by using a blur-aware attention module. Despite being proposed before Restormer, BANet does not have the same problem as Restormer on deblurring the non-uniform images. However, training BANet requires a lot of time and high computing capacity.

This research will conduct image deblurring using both Restormer and BANet on the same set of images and the same computing environment. The datasets used are GOPRO [9] and HIDE [10] datasets, which both are standard blurred image datasets that are widely used on various image deblurring problems. The deblurred results will then be compared, with a special focus on non-uniform blur. While various methods can handle most uniform blur problems, they are not yet able to distinguish non-uniform blur, and hence perform worse than those types of images.

In short, through this paper, we will conduct a comparative analysis of two state-of-the-art deblurring methods, i.e., Restormer and BANet. We also put a special emphasis on the performance of deblurring non-uniform blur, which has seldom been addressed in previous works. The results will be compared quantitatively and qualitatively to provide a thorough discussion of the results. We hope to provide a benchmark for both methods and provide more insight for future work in the area of deblurring in general, but more specifically, handling non-uniform blur.

2. Methods

The methods used in this experiment consist of Restormer [7] and BANet [8]. Both methods are image restoration models, proposed to handle various degradation types in images, including blur.

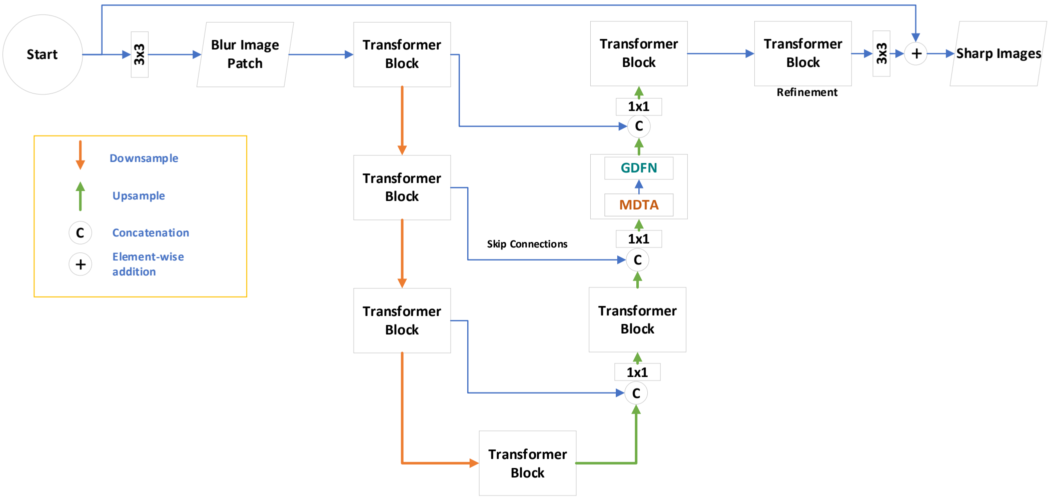

Figure 3. Restomer Architecture, which used transformer based methods with 4 layer deep of transformer block [7].

2.1 Restormer

Transformers are one of the latest developments of deep neural network architectures which use self-attention techniques. Self-attention is a module that uses image patches as input, transforms it then returns the value of the token as output. A popular transformer model that is used for image processing is called Visual Transformer (ViT) [11].

Restormer falls into the category of a visual transformer used to model image restoration. Restormer can handle several restoration tasks, such as super-resolution, image deraining, and image deblurring. The strong point of Restormer is that it can restore diverse types of image degradation on a single network architecture, while still providing shorter training time. The architecture of Restormer is shown in Figure 3.

The main components of Restormer are Multi-Dconv Head Transposed Attention (MDTA) and Gated-Dconv Feed-Forward Network (GDFN). MDTA applies Self Attention (SA) on feature dimensions, which calculates the cross-covariances of feature channels to capture attention maps from the input feature. MDTA emphasizes local features and models the relation of global context between pixels while still considering the covariance-based attention maps. GDFN uses a gating mechanism to improve information flow on the network with GELU activation. GDFN used a local context feature, like MDTA.

Lastly, Restormer can model image restoration better using a progressive learning strategy. Progressive learning is a learning strategy that starts by using smaller patches and bigger batch sizes in the early epoch, gradually progressing to bigger patches with smaller batch sizes in the later epoch. This progressive learning approach can also be described as making the network learn a simpler task and gradually moving to a more complex one.

2.2 BANet

BANet uses a general U-Net architecture design while also employing attention blocks on the network. BANet's strong point is being able to solve the non-uniform blur problem on images, largely thanks to its attention modules BANet also provides faster inference time which is useful on real-time image deblurring problems, although having slower training time compared to Restormer [7].

The main component of BANet is the Blur-Aware Module (BAM), which is composed of Blur-aware Attention (BA) and Cascaded Parallel Dilated Convolution (CPDC), as shown in Figure 2.

The Blur-aware Attention (BA) is used to capture the orientation and magnitude of blur both locally and globally, while CPDC is used to learn the blur pattern on multiple scales. BA itself has its components, i.i.e., Multi-Strip Kernel Pooling (MKSP) and Attention Refinement (AR). MKSP's main function is to mask regional and directional artifacts caused by non-uniform blur, while AR aims to further enhance the mask by using element-wise multiplication. CPDC is used to improve network modeling capabilities by implementing atrous convolution to expand the receptive fields of networks, making the networks capable of extracting features from different size images without increasing the kernel size.

3. Experimental Setup

The experiment was conducted by using a Nvidia DGX-1 graphic card with 32GB GPU Memory. Restormer and BANet are used as model baselines, which are later compared against each other using the evaluation metrics designed for this experiment. The implementation of Restormer and BANet were taken as-is from their original papers [12] [13]. The details of experiments design in this study, such as image dataset, hyperparameter settings, and evaluation metrics are explained in the following subsection.

3.1 Dataset

The experiments were conducted by using well-known and benchmarked datasets for image deblurring, i.e., GOPRO [9] and HIDE [10]. The GOPRO dataset is one of the earliest datasets available for image deblurring problems and is already used in various image deblurring methods. Meanwhile, the HIDE dataset consists of images that contain multiple blurs caused by relative movement between the imaging device and the scene, which is suitable for experiments that focus on non-uniform blur. Samples of the GOPRO and HIDE datasets can be seen in Figure 4.

3.1.1 Training Dataset

The GOPRO dataset was used to train both Restormer and BANet models. The training dataset was taken from the entire GOPRO train set as published, which consists of 2101 pairs of blur-sharp images, with various degrees of motion blur.

3.1.2 Testing Dataset

For the testing phase, this experiment used three different datasets. First, we used the published testing set from the previously mentioned GOPRO dataset. Next, the HIDE dataset was used as a second testing set.

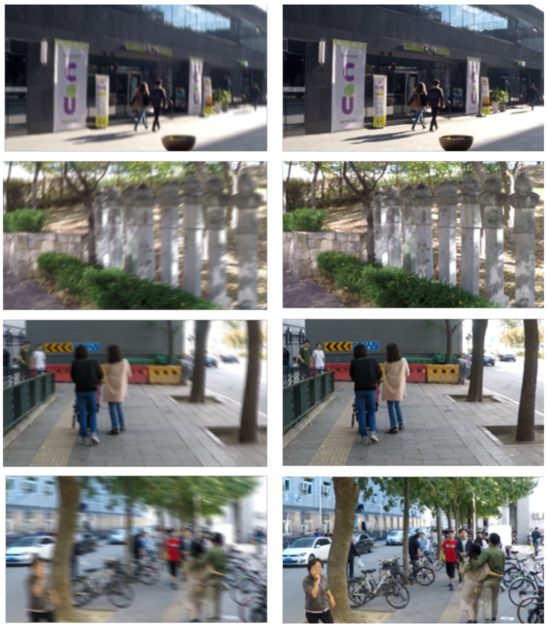

Figure 4. Sample of GOPRO and HIDE dataset image. The left row is the blurred images, right row is the sharp images.

Additionally, the models were also tested with GOPRO-NU, a hand-picked portion of the GOPRO dataset that contains non-uniform blur images, since this study is particularly interested in non-uniform blur evaluation. Three different subfolders were taken from the testing set of GOPRO, which appear to have non-uniform motion blur that was caused by moving objects on a static background. Samples of GOPRO-NU are shown in Figure 5.

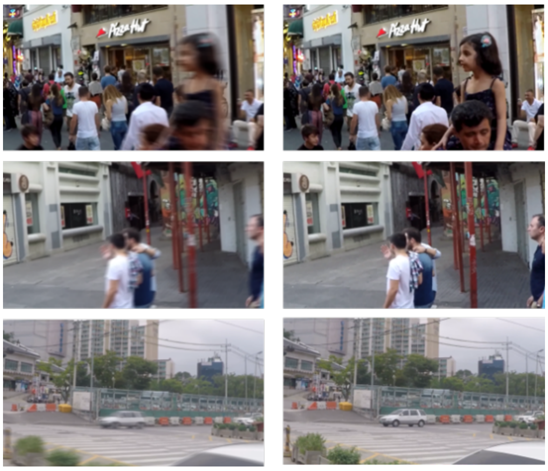

Figure 5. Sample of GOPRO-NU dataset image. The left row is the blurred images, right row is the sharp images.

These datasets are then used for cross-dataset testing and also evaluate models performance on non-uniform blur images. The testing set division is shown in Table 1.

Table 1. Dataset Details

Test Set	GOPRO	HIDE	GOPRO NU
Number of Images	1111	2025	334

3.2 Hyperparameter Tuning

As with most deep learning models, the hyperparameters for training are tuned to optimize model training based on the hardware resources used in the experiment. The hyperparameters used to train Restormer are shown in Table 2, while the hyperparameters for training BANet are shown in Table 3.

Table 2. Hyperparameters of Restormer used in our experiments.

Hyperparameter	**Default**	**Experiment**
Batch Size	[64, 40, 32, 16, 8, 8]	[8, 5, 4, 2, 1, 1]
Epochs	30000	15000
Learning Rate	3e-4 down to 1e-6	3e-4 down to1e-6

As shown in Table 2, the experiment is conducted by changing the batch size and epoch of the Restormer model so the model can run on the available hardware resources.

Table 3. Hyperparameters of BANet used in our experiment.

Hyperparameter	**Default**	Experiment
Epochs	3000	2000
Batch	8	8
Learning Rate	1e-4 down to 1e-7	1e-4 down to 1e-7

The hyperparameter adjustment on BANet reduced the training epochs from 3000 to 2000, as shown in Table 3.

3.3 Evaluation Metrics

The evaluation metrics used in this experiment were Peak-Signal to Noise-Ratio (PSNR) and Structural Similarity Index (SSIM).

PSNR is calculated by using the comparison of MSE between the deblurred image and the ground truth. PSNR is formulated as in Equation (1).

$$PSNR = 10log10[\frac{L-1^2}{MSE}] \quad (1)$$

where L indicates the maximum intensity of the image. In our case of 8-bit images, this value is equal to 255.

Unfortunately, the PSNR is only able to calculate signal and noise, which is not necessarily

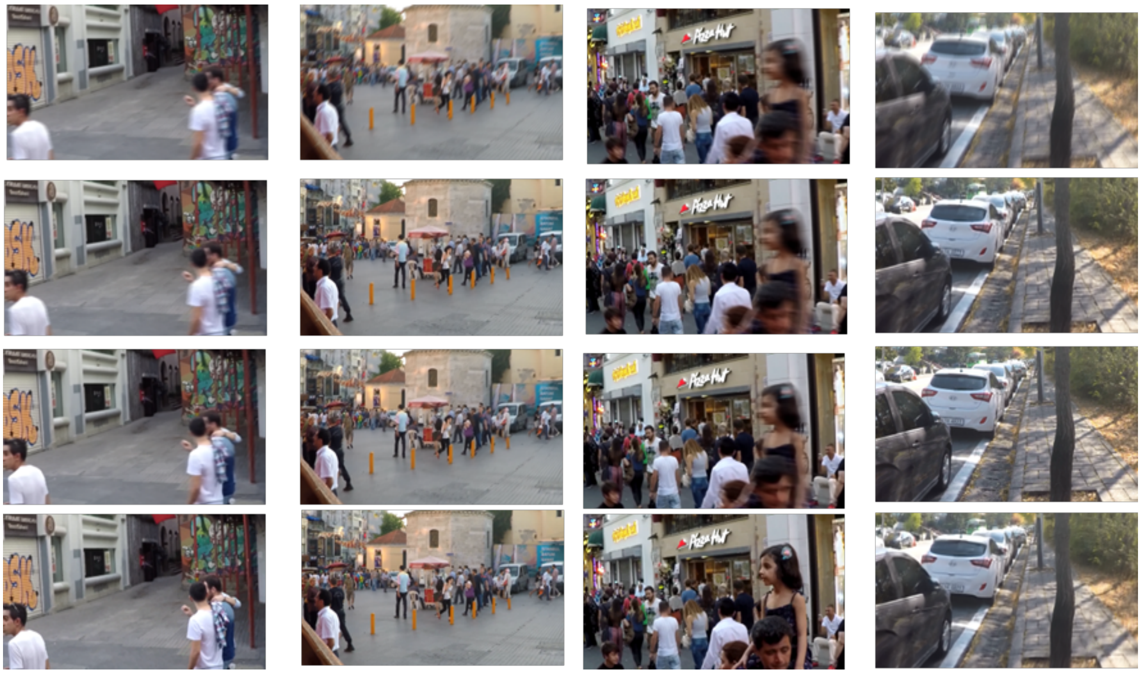

Figure 6. The comparison of Restormer and BANet deblurring result on GOPRO test set. The first column shows the blurred image, second row shows the deblurred result by using Restormer, third row shows the deblurred result by using BANET, while fourth row shows the ground truth images.

a good representation of what is contained and visible in the image. As an alternative, SSIM is also used. SSIM is a metric based on human vision, which calculates the image structure in a way that is similar to the evaluation by human eyes. SSIM can be formulated as described in Equation (2)

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \qquad (2)$$

where μ indicates luminance from the images calculated by averaging the pixel values, while σ indicates the contrast of images which is calculated from the standard deviation of the pixel values.

4. Results and Analysis

The first step of this research is to successfully implement Restormer and BANet on the same platform and computation environment. The models are implemented on the Nvidia DGX-1 GPU, trained, and tested using PyTorch. The models are then trained using the GOPRO training set and hyperparameter settings as described in subsection 3.2.

After the models have been trained successfully, the models are both tested using the test images described in Section 3.1. We conduct 3 levels of testing using the three testing datasets as described in subsection 3.1. The first level of testing was done with the GOPRO test set, next the the HIDE test set, and then with the non-uniform dataset GOPRO-NU.

First, we look at the result of deblurring the GOPRO test set. This first level of testing can be seen in Figure 6. While both methods can deblur most blur images, Restormer still shows some difficulty in deblurring some images. Overall Restormer does not perform as well as BANet, especially for deblurring images with non-uniform blur.

For the next testing phase, we evaluated the robustness of the trained models using the HIDE test set. This demonstrates a cross-dataset test to see if the models can restore general blurred images that may differ from the training set. As seen on Figure 7, both Restormer and BANet can deblur most blur images. As we saw with the GOPRO test result, Restormer also does not perform as well as BANet.

Finally, we perform the last level of testing with a specific focus on non-uniform blur images. For this, we used the GOPRO-NU dataset as described in subsection 3.1.2. The results can be seen in Figures 8 and 9.

As we can see in Figure 8, the deblurred image with Restormer in (c) was not able to restore the blurring effect caused by the moving person, as the object appears to have a different magnitude of blur compared to other regions of the image. Meanwhile, Figure 8 (d) shows the deblurring results using BANet. BANet was able to solve the different magnitudes of blur on the same test image and provide a better deblurred image overall as the result.

Another deblurred image is shown in Figure

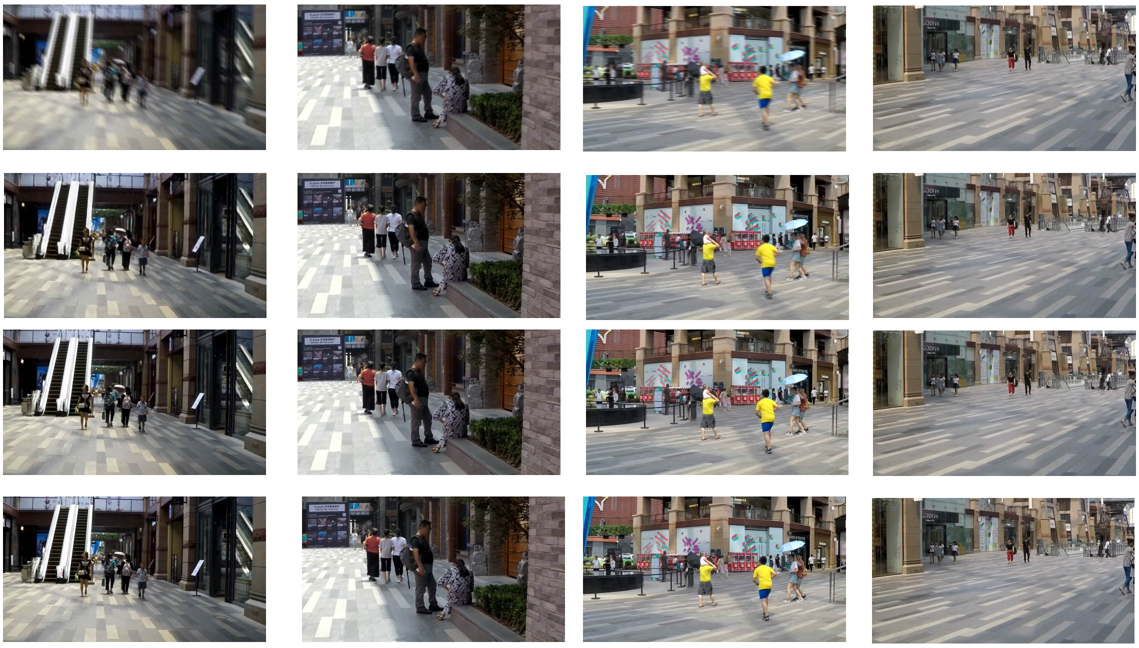

Fig. 7. The comparison of Restormer and BANet deblurring result on HIDE test set. The first column shows the blurred image, second row shows the deblurred result by using Restormer, third row shows the deblurred result by using BANET, while fourth row shows the ground truth images.

9 (c). This time, Restormer appears to be successful at deblurring the image rather well. While Restormer performed better in Figure 9 compared to the previous image in Figure 8, it still results in a blurrier image compared to BANet. As can be seen in Figure 8 (d), BANet produces a relatively better result for the deblurred image.

As for the quantitative result by using PSNR and SSIM Metrics, can be seen in Table 5.

Table 5. PSNR results by using Restormer and BANet.

Methods	**Test Set**		
	GOPRO	HIDE	GOPRO-NU
Restormer	33.16	33.21	33.50
BANet	**34.90**	**34.52**	**35.47**

Table 5 shows the result of deblurring using Restormer and BANet on 3 different test datasets which were calculated using PSNR metrics. These results show that the deblurred image by Restormer still has lower quality and more noise compared to the deblurred image produced by BANet.

Table 6. SSIM results by using Restormer and BANet.

Methods	**Test Set**		
	GOPRO	HIDE	GOPRO-NU
Restormer	0.847	0.833	0.871
BANet	**0.926**	**0.908**	**0.935**

Table 6 shows the SSIM score of Restormer and BANet on 3 used test datasets. Similar to PSNR metrics, BANet also outperformed Restormer on all test datasets. The main difference in this metric is that the results of both methods are not far off each other. This means that Restormer and BANet can fix the image structure of the blurred image.

Based on Table 5 and Table 6 results, BANet performs better than Restormer on both image quality metrics. This is caused by the incapability of Restormer to deblur the blur on the image that has different magnitudes of blur on various regions of the image, while BANet has no problem deblurring the same problem and can perform better on the overall result. This can also be seen from the GOPRO-NU dataset which has non-uniform blur. While both methods achieve higher metrics, Restormer still performs worse than BANet. Another comparison is performed on the HIDE dataset which deblurred using the model that has been trained using the GOPRO dataset. In this cross-dataset experiment, BANet still performs better compared to Restormer. HIDE dataset which consists of both uniform and non-uniform blur images reduced the performance of Restormer.

5. Discussion

Additionally, we also compared inference times for both Restormer and BANet. This way we can

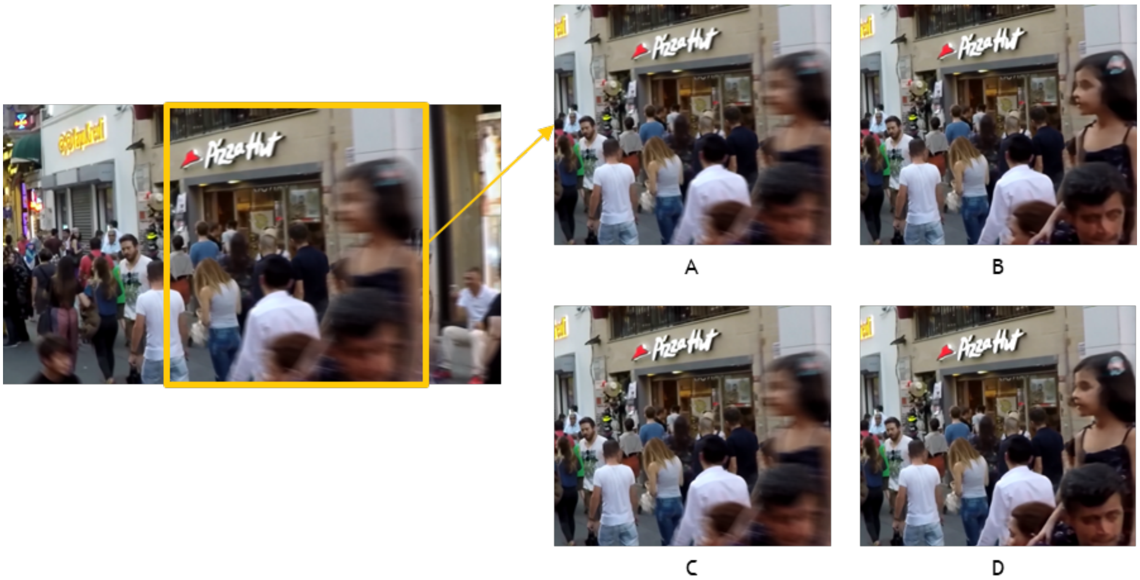

Figure 8. Closer comparison of Restormer and BANet deblurring result on GOPRO-NU test set. (a) shows the blurred image, (b) shows the ground truth image, (c) shows the deblurred result by using Restormer, while (d) shows the deblurred result by using BANet.

estimate how feasible it would be to deploy these models for real-time deblurring. We measured the inference time for Restormer on 50 test images, and on average we obtained an average of 300ms. For BANet, we measured the average of 50 test images in the same way, and we obtained 3ms per image. Despite being slower on training phase, BANet provide faster inference time and better result compared to Restormer.

To further show certain model performs better than another model, a significance test needs to be conducted. One of the tests available is the paired samples t-test. Using a selected significance level (in this experiment, we use $\alpha = 0.05$), we will use that as a threshold to p-value, which can be calculated by using two populations data as shown in (3).

$$t = \frac{\bar{x}_{diff}}{(^{S_{diff}}/_{\sqrt{n}})} \quad (3)$$

Where t indicates test statistic, $\bar{x}_{diff}$ indicates the sample mean of the differences, S_{diff} indicates the sample standard deviation of the differences, and n indicates the sample size.

In our experiment we test the null hypothesis of Restormer perform equal or better compared to BANet againts the alternate hypothesis Restormer perform worse than BANet, with the PSNR and SSIM result for both Restormer and BANet as μ_1

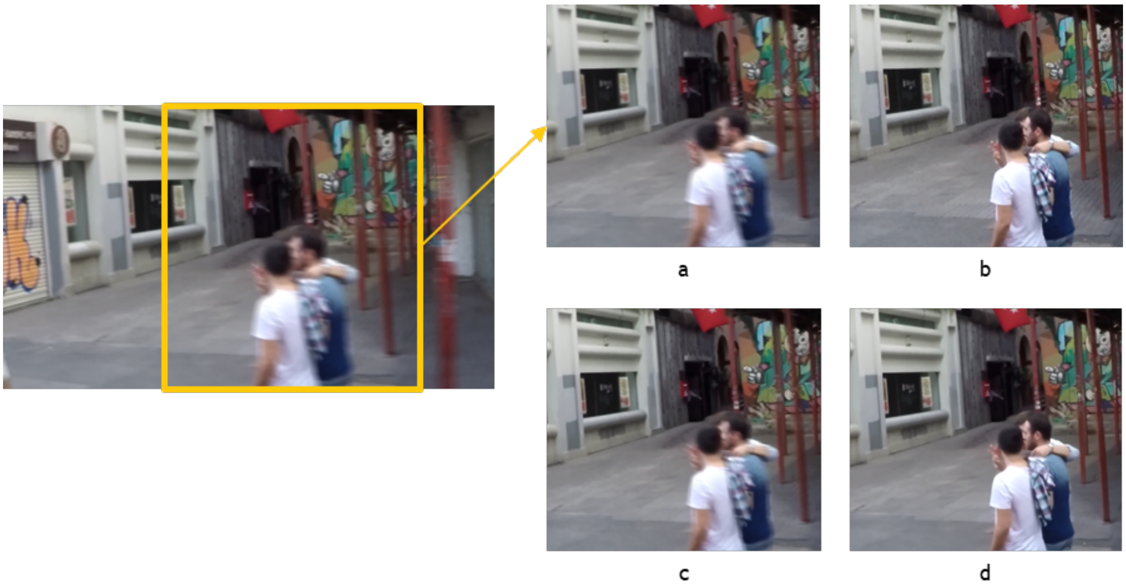

Figure 9. Another closer comparison of Restormer and BANet deblurring result on GOPRO-NU test set. (a) shows the blurred image, (b) shows the ground truth image, (c) shows the deblurred result by using Restormer, while (d) shows the deblurred result by using BANet.

and μ_2 respectively. The t-test results can thus be seen in Table 7.

Table 7. T-test result of Restormer and BANet.

Metrics	**p-value**	**Hypothesis**
PSNR	20.02e-78	Rejected null hypothesis
SSIM	1.23e-73	Rejected null hypothesis

Table 7 shows that both PSNR and SSIM null hypothesis results were rejected, which means that Restormer performances were worse than BANet.

6. Conclusion

Through the three levels of testing and significance test conducted in our experiment, all results confirm that in all testing scenarios, BANet can perform better consistently compared to Restormer.

While Restormer can perform various image restoration including image deblurring, its performance is still lacking when handling specific non-uniform deblurring. As shown in the result of conducted experiments and significance test, Restormer still has problems with deblurring an image that has non-uniform blur, while BANet can handle the problem better, which is shown on better PSNR and SSIM on GOPRO-NU dataset.

We suspect that this result is due to BANet using MKSP, which focused on capturing regional blur and directional artifacts, making the methods more robust in deblurring the non-uniform blur on the images.

7. Acknowledgements

The authors express our gratitude to the Tokopedia-UI Artificial Intelligence Center of Excellence for the provision of the NVIDIA ® DGX-1 used in this experiment.

References

[1] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan, “Blur-Invariant Deep Learning for Blind-Deblurring,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 4762–4770, 2017, doi: 10.1109/ICCV.2017.509.

[2] T. Zeng and C. Diao, “Single Image Motion Deblurring Based on Modified DenseNet,” *Proc. - 2020 2nd Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2020*, pp. 521–524, 2020, doi: 10.1109/MLBDBI51377.2020.00109.

[3] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, “Image and video restorations via nonlocal kernel regression,” *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1035–1046, 2013, doi: 10.1109/TSMCB.2012.2222375.

[4] K. Zhang *et al.*, “Deep Image Deblurring: A Survey,” *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2103–2130, 2022, doi: 10.1007/s11263-022-01633-5.

[5] R. Wang and D. Tao, “Recent Progress in Image Deblurring,” pp. 1–53, 2014, [Online]. Available: http://arxiv.org/abs/1409.6838

[6] P. Wang *et al.*, “Non-uniform motion deblurring with blurry component divided guidance,” *Pattern Recognit.*, vol. 120, p. 108082, 2021, doi: 10.1016/j.patcog.2021.108082.

[7] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. H. Yang, “Restormer: Efficient Transformer for High-Resolution Image Restoration,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2022-June, pp. 5718–5729, 2022, doi: 10.1109/CVPR52688.2022.00564.

[8] F. J. Tsai, Y. T. Peng, C. C. Tsai, Y. Y. Lin, and C. W. Lin, “BANet: A Blur-Aware Attention Network for Dynamic Scene Deblurring,” *IEEE Trans. Image Process.*, vol. 31, pp. 6789–6799, 2022, doi: 10.1109/TIP.2022.3216216.

[9] S. Nah, T. H. Kim, and K. M. Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 257–265, 2017, doi: 10.1109/CVPR.2017.35.

[10] Z. Shen *et al.*, “Human-aware motion deblurring,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 5571–5580, 2019, doi: 10.1109/ICCV.2019.00567.

[11] A. M. Ali, B. Benjdira, A. Koubaa, W. El-Shafai, Z. Khan, and W. Boulila, “Vision Transformers in Image Restoration: A Survey,” *Sensors*, vol. 23, no. 5, 2023, doi: 10.3390/s23052385.

[12] H. Ren, “Restormer,” 2022. https://github.com/leftthomas/Restormer (accessed Mar. 13, 2024).

[13] F. J. Tsai, “BANet-TIP-2022,” 2023. https://github.com/pp00704831/BANet-TIP-2022 (accessed Mar. 13, 2024).

[14] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for

Biomedical Image Segmentation," *Med. Image Comput. Comput. Interv. 2015 18th Int. Conf.*, vol. 9351, no. Cvd, pp. 12–20, 2015, doi: 10.1007/978-3-319-24574-4.