Self-Supervised Image Denoising with Noise Correlation Priors

Cheuk Hei Yu (Hayden), and Hok Yin Yu (Boris)

Abstract—Although there are lots of reasearch studies working on image denoising, they mainly rely on adding synthetic Gaussian noise for supervised training, which always fail in real-world settings. Besides, creating a real-world dataset for supervised training is labour intensive and time-consuming. Thus, self-supervised image desnoising approach is becoming more popular these days. There are 2 latest self-supervised denoising approaches - Local and Global Blind-Patch Network and Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising, which perform state-of-the-art performance on real-world sRGB photographs. In this paper, we investigated pixel-wise noise correlation to regenerate the findings in the previous papers. Besides, we reworked code provided SSID into a well-design library to allow easy switching of BNN and LAN models. Apart from that, we performed additional ablation studies to identify model significance and interpretability, as well as replaced model components with other state-of-the-art parts to explore model interchangeability.

Index Terms—Image Denoising, Self-Supervised Learning, Noise Correlation

1 Introduction

Image denoising is one of the many tasks in the Image Signal Processing Pipeline that removes noise while retaining visual details of an image. Due to its importance and widespread application, it has been a fundamental research area in Computer Vision. For example, when capturing astrophotographic images, low light together with long exposure time unavoidably introduces lots of noise, which is not ideal.

Currently, numerous machine-learning-based denoising algorithms have been proposed. Most of these methods rely on additive white Gaussian noise to synthesize noisy images from clean ones for supervised training. However, this is not realistic in certain ways. First, real-world noise comes from multiple sources, such as Gaussian-distributed read noise, Poisson-distributed shot noise, and uniformly distributed Quantization noise. Second, images shot under different lighting and environments can have different noises. While several attempts [1] [2] have later been made to create real-world datasets, they are costly, time-consuming, and labor-intensive.

To overcome the above-mentioned limitations, self-supervised learning [3] [4] [5] is introduced to approach image-denoising problems. While traditional supervised techniques rely on noisy and clean image pairs for training, self-supervised learning relaxes this requirement and uses only the noisy image. The intuition behind this is to leverage intrinsic data distributions and priors derived directly from the noisy images.

In this project, we start by reproducing state-of-the-art self-supervised image-denoising works and proceed to modify their works for improvement. Although we fail to do better than these works, our contributions can be summarized in 3-folds. First, we extend the investigation of pixel-wise noise correlation as image priors from a size of 9x9 to 21x21. Second, we reworked the code provided by SSID [5] into a well-designed library for easy component switching. Third, we explored model interchangeability by replacing model components across research works.

2 Related Work

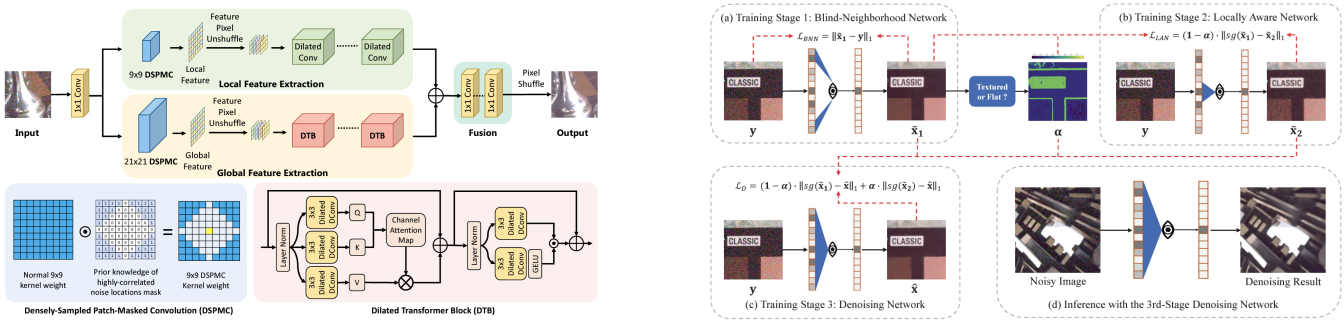
Due to the absence of appropriate training data, self-supervised image-denoising techniques have been introduced in recent years. One of the fundamental techniques in literature for self-supervised denoising is Blind-spot network (BSN) [6], which learns the masked center pixel by referring to its receptive field. This method exhibits the capability to effectively eliminate pixel-wise independent noise when trained on identical noisy images serving as both input and target. Based on this technique, several state-of-the-art self-supervised denoising models are being introduced.

2.1 Self-Supervised Denoising for Real-World Images via Asymmetric PD and Blind-Spot Network (APBSN)

Apart from BSN, Pixel Downsampling (PD) is also a good technique to remove spatial correlation of real-world noise. APBSN [3] proposes to integrate both PD and BSN by using an Asymmetric Pixel-shuffle downsampling framework. By using asymmetric PD stride factors for training and inference, this avoids the problem of breaking spatial correlation and aliasing artifacts during downsampling. In addition to the asymmetric framework, the authors propose PD refinement to minimize the loss of visual features during inference.

2.2 Local and Global Blind-Patch Network (LGBPN)

While AP-BSN demonstrated promising results, using BSN focuses on recovering center pixels based on all neighbors. This is not ideal since neighborhood pixels are also highly



(a) Model Architecture of LGBPN

(b) Model Architecture of SSID

Fig. 1: A side-by-side comparison of the two recent works from CVPR 2023

correlated in noise. Following AP-BSN's findings where noise is most correlated in a 9x9 region, they replace the BSN with a Blind Neighbourhood Network (BNN) by masking additional pixels during the learning process. In this way, they estimate the center pixel using farther neighborhood pixels that are less correlated in noise. Their works consist of two branches, local feature extraction, and global feature extraction. Figure 1a shows the model architecture of LG-BPN [4]. The major difference between the branches is to capture information with different receptive fields.

Although they manage to produce quality results, we notice several issues with their work. First, there is a lack of explanation as to why a receptive field of 21x21 is used in the global branch. Second, there is a long inference time due to high model complexity. In particular, the Dilated Transformer Block in the global branch is expensive in practice. Third, the separation between local and global is not intuitive. The receptive fields between the two overlap and it is hard to come up with a meaningful interpretation of what information they are trying to capture.

2.3 Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising (SSID)

SSID [5] is a 3-stage training procedure consisting of a Blind Neighborhood Network (BNN), a Locally Aware Network (LAN), and a Denoising Network. Unlike LGBPN which learns local and global features in two branches simultaneously, this work splits features according to characteristics and learns them at different stages. The first stage uses a BNN to learn denoising in flat areas. The second stage uses the results from the first stage for the supervision of texture areas such as edges and texts. Finally, the third stage balances the flat and texture features using a spatially adaptive coefficient. As a result, this strikes a perfect balance between the over-smoothing results of flat regions given by BNN and the noisy results in texture areas captured by LAN.

In SSID, we see a clear separation of features and the use of a three-stage method in fusing the flat and texture features. We found such techniques inspiring and would like to extend their works.

3 Proposed Method

3.1 Motivation

Understanding features is critical to improving model performance. As we would like to understand the role of each component, we perform a variety of ablation studies on top of the works of LG-BPN [4] and SSID [5]. Further, we notice a trend of using BSNs or BNNs in self-supervised image-denoising in recent years. Since they use different BNN architectures, we would like to interchange them to observe any differences. Ideally, we expect similar results as they share the same purpose.

To support easy switching between components in the three-stage network. we reworked the code provided by SSID into a well-designed library to allow easy switching of BNN and LAN models. For example, if we would like to switch to another BNN, we can simply replace Line 1 in Figure 2 with your custom BNN model.

```
1 model_bnn = SSID_BNN(args.bnn_cfg_path)
2 model_lan = SSID_LAN(args.lan_cfg_path, model_bnn)
3 model_unet = SSID_UNet(args.unet_cfg_path, model_bnn, model_lan)
4 model_unet.train()
```

Fig. 2: Sample call to our redesigned code.

3.2 Noise Correlation Analysis

In this section, we analyze the real-world noise correlation between each center pixel and their neighborhood pixels across all images to produce a correlation map. With a map size of 21x21, we try to search for reasons why LGBPN decided to use a receptive field of 21x21 for the global branch. Specifically, we let X be the noise obtained by subtracting the noisy image from the clean image, X_{ij} then refers to the noise at row i and column j. The noise correlation map can be computed as follows:

$$r_{kl} = \sum_{i,j} r(X_{i+k,j+l}, X_{i,j}),$$

where $k, l \in [-10, 10] \times [-10, 10]$ to produce a correlation map of size 21, and r is the pearson correlation coefficient:

$$r(x,y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

In this context, the values for the data points (x_i, y_i) will be the RGB values. This method only applies to SRGB images but not RAW ones since we cannot compute the correlation coefficient similarly without RGB noise values.

Finally, we average out the correlation values across all images from the dataset to obtain the desired correlation map.

3.3 Blind Neighbourhood Network (BNN)

The works LGBPN [4] and SSID [5] each feature a different strategy for BNN despite having the same goal. We adopt both strategies in our analysis and interchange them to observe model behavior.

LGBPN [4] implements the BNN by masking out the values in the center. This is done by multiplying a mask on top of the convolution kernel K_{Conv2D}:

$$K = K_{Conv2D} * Mask_{Corr}$$

Referring to figure 3, $Mask_{Corr}$ is the prior knowledge of highly correlated noise locations obtained in Section 3.2 that enables self-supervised learning. We are trying to predict the yellow pixel in the center by using far neighbors colored in blue. This avoids learning from the strongly correlated neighbor pixels (in white) while extracting as much local information as possible during training.

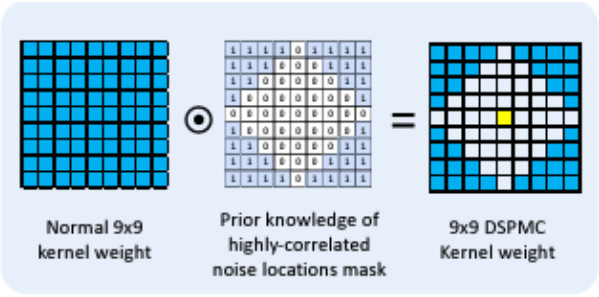


Fig. 3: Patch-Masked Convolution used by LGBPN

To further preserve high-frequency local details, especially during inference, the authors introduce a kernel shift strategy based on deformable convolution [7]. Given input and output features x and y, we denote the position of the feature at p as $x(p)$ or $y(p)$. The strategy shifts the kernels to the center with the equation

$$y(p_0) = \sum_{k=1}^{K} w_k \cdot x(p_0 + p_k + \alpha * (p_k - p_0)),$$

where α is some predefined offsets denoting the extent of the kernel shift and K is the kernel sampling locations. In other words, for each kernel pixel k, we shift it from position $p_0 + p_k + \alpha * (p_k - p_0)$ of the input feature to position p_0 of the output feature. Figure 4 illustrates the shifting procedure during inference.

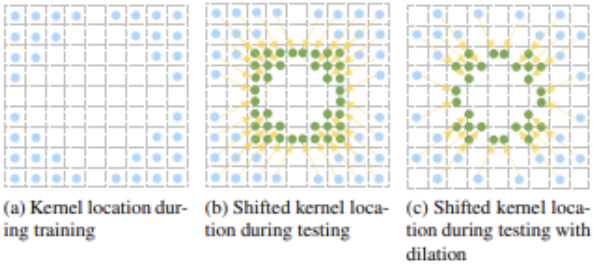


Fig. 4: Kernel Shift Strategy Illustrated

SSID [5], on the other hand, creates the Blind Neighborhood Mask by shifting the kernel to exclude the center pixel at each layer. By stacking k 3x3 shifted convolution layers, this produces a $(2k+1) \times (2k+1)$ blind neighborhood. This differs from that of LGBPN [4] since they are not multiplying the kernel by the noise correlation mask directly. Figure 5 demonstrates the shift at each layer.

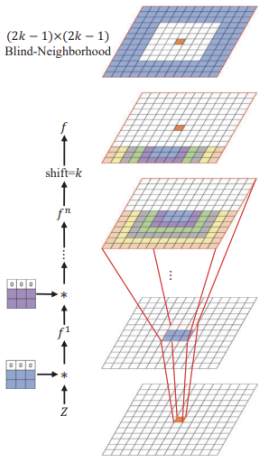


Fig. 5: BNN used by SSID

3.4 Model Component Analysis

In this section, we present the list of ablation studies that we had, the intuition, and what we are expecting to see from each of them. These ablation studies can be separated into three groups, LGBPN-based, SSID-based, and a mixture of the two.

3.4.1 LGBPN-based Ablation Studies

The first study we did was to replace concatenation fusion with average fusion when combining the local and global branches (L-2). In the original LGBPN paper [4], concatenation fusion was employed to combine the local and global branches, primarily to facilitate the subsequent convolution calculations. However, we decided

to investigate an alternative fusion method, opting for average fusion in our initial study. Our motivation for this change stems from SSID [5] having a spatially adaptive weight, but instead, we fix this weight to 0.5 since it is hard to tell whether local or global features are more important by their names. While concatenation fusion may not pose significant computational demands, it introduces additional parameters and intricacies to the model architecture.

The second approach involves substituting Dilated Convolution Layers (DCL) with Dilated Transformer Block (DTB) and reducing the number of DTBs from 6 to 3 (L-3). The idea is to investigate the capabilities of DTB in capturing global interactions compared to DCL. According to the authors, DTB is designed to avoid information exchange between spatially adjacent pixels, thereby satisfying the pixel-wise independent requirement of the blindspots [4]. We would like to determine DCL's ability to extract local connectivity from images and assess its necessity in the LGBPN model. The decision to assign only 3 DTBs to each branch is rooted in the substantial computational power required for each DTB. To ensure the feasibility of training our model on limited GPU resources, we had to limit the number of blocks to 3 blocks per branch.

The third one is to remove one of the branches from the model, which is to train the LGBPN model with only the local branch or the global branch (L-4, L-5). This deviation from the original architecture was motivated by the observation that the terms "local" and "global" in LGBPN [4] lack a clear demarcation, unlike the more distinct separation of flat and texture features in SSID [5]. Since there are overlapping receptive fields between the local and global branches, we raise the concern of potential redundancy in maintaining both branches. Thus, we trained the local branch and global branch separately to understand what each branch contributes individually and determine if their collaboration is crucial for the LGBPN model structure. Additionally, the image outputs from these two models can assist us in defining the characteristics associated with "local" and "global" features as mentioned in the context of the LGBPN model.

3.4.2 SSID-based Ablation Studies

For SSID, we propose one simple experiment by replacing the loss function in the second-stage model LAN to supervise using the noisy image instead of the output from the BNN in Stage 1 (S-2). Mathematically, we replace the equation:

$$L_{LAN} = ||BNN(x) - y||_1, \text{ with } L_{LAN} = ||x - y||_1,$$

where x is the input noisy image, and y is the output of LAN.

The authors propose to supervise LAN using the outputs from BNN [5]. This is because BNN learns to recover the flat areas, which provide some clean signal for LAN to train with. By modifying the loss function to supervise the noisy image, we are guiding the model to predict the center pixel using highly correlated noisy pixels. If things go on the right track, we should expect LAN to produce noisy outputs instead of capturing texture areas as expected. This also verifies the reason why we need BNNs in masking out center neighborhoods.

3.4.3 Interchanging components from LGBPN and SSID

Further, we attempt to interchange the BNN component of LGBPN and SSID (L-1, S-1). That is, for LGBPN, we replace its local branch by SSID's BNN, and for SSID, we replace its Stage 1 BNN learning process with LGBPN's local branch. As described in Section 3.3, both BNN components share the same objective even though they feature different strategies. If the results produced after interchanging the components are similar, we can divide the research work into parts and focus on improving individual components. For example, building a better BNN will eventually improve the results for all the three stages, and hence improving denoising results.

4 Experiments

4.1 Experimental Settings

To obtain a comparative analysis between methods, we follow previous works and work on SRGB images. For comparative analysis, we try to follow as many hyperparameters from previous works, which include learning rate, number of gradient steps, and loss functions.

4.1.1 Dataset

We trained and evaluated our method based on Smartphone Image Denoising Dataset (SIDD) [1]. The authors created this dataset by collecting real-world raw images from five smartphone cameras and manually cleaned the images to obtain the ground truth. We use the SIDD-Medium dataset, with 320 images of dimension 4032x3024 for training, 1280 256x256 patches for validation, and 1280 256x256 patches for testing. During training, we perform simple data augmentation procedures such as random crop to size 256x256, random flip, and transpose to increase model robustness.

4.1.2 Training Details

The noise correlation map is computed on Intel i9-13900K CPU with all 24 cores enabled. As for the machine learning models, they are trained on either NVIDIA GeForce RTX 4070, 4090, or RTX A4500 depending on the machine availability of UofT GPU clusters. For LGBPN-based methods, we set the learning rate to 1e-4 and train all networks for 20 epochs with batch size 8. As for SSID-based methods, the learning rate is set to 3e-4 with a cosine annealing scheduler trained for 400k gradient steps in each stage. To evaluate our results, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate the performance.

4.2 Experimental Results

4.2.1 Noise Correlation Analysis

We present the noise correlation map that we compute across the SIDD dataset [1] in Figure 6. As to our expectation, the region within a distance of 4 from the center contains most correlated noise, which matches previous analysis from APBSN [3]. We still see a minimum amount of

noise outside this region but they are insignificant relative to the center region. Surprisingly, we observe checker patterns of noise correlation from outer regions. This might be due to the wave-particle duality of the photons hitting the camera sensor, or defects during demosaicking the RAW into SRGB images when preparing the dataset.

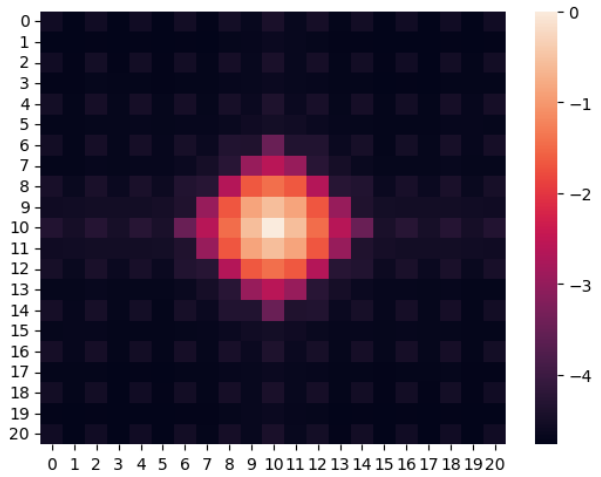


Fig. 6: Noise Correlation Map (in log-scale)

Since we compute the correlation map by averaging out results across all images from the dataset, we would also like to investigate the noise correlation values of individual images. From this, we randomly selected 4 pixels and plotted the distribution of the values for that pixel as in Figure 7. We see that not all images have low correlation values in outer regions. In particular, there are a few outliers having correlation values of more than 0.2. We manually looked into these images but failed to identify characteristics leading to such results. However, we should keep in mind inconsistent noise distributions even under the same camera. This could happen when there are different lighting or camera settings when shooting the images.

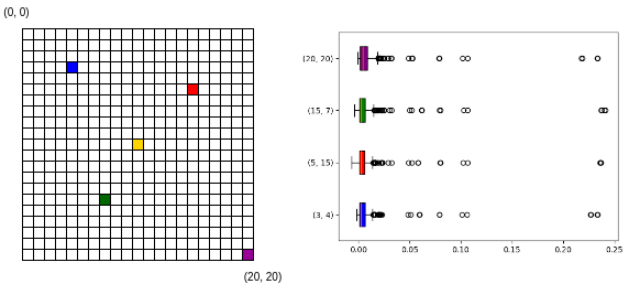


Fig. 7: Noise distribution for certain pixel

4.2.2 *Model Component Analysis*

We assess the effectiveness of each method in real-world image denoising using the SSID validation dataset.

From Table 1, unfortunately, none of our approaches manage to reach the PNSR and SSIM value from LGBPN and SSID papers. This is somehow disappointing but expected since they are all state-of-the-art works published in CVPR 2023 few months ago. Yet, most of our models achieve PSNR values in the range of 36 to 37, indicating commendable performance in minimizing pixel-wise differences during denoising tasks. However, the variability observed in SSIM values, ranging from 0.85 to 0.93, underscores the differences among the models in their ability to effectively recover and preserve structural information within the denoised images. Besides, the approach having the most similar result as the original papers is the average fusion approach (L-2), but still its result reveals that the concatenation fusion is better way for LGBPN model to combine the output from local and global branches.

TABLE 1: Experiment Result

	PSNR	SSIM
LGBPN (Original)	37.280	0.9360
LGBPN (L-1)	33.546	0.8446
LGBPN (L-2)	37.138	0.8858
LGBPN (L-3)	36.798	0.8810
LGBPN (L-4)	36.038	0.8629
LGBPN (L-5)	36.212	0.8614
SSID (Original)	37.390	0.9340
SSID (S-1)	36.015	0.8732
SSID (S-2)	25.320	0.4047

Apart from that, by comparing the images generated from the local-branch-only (L-4) and global-branch-only (L-5) models in Figure 8, local-branch-only model can retain more details of the image and the edges are sharper than the global-branch-only model. However, when comparing the overall PSNR and SSIM value, both models are having a similar result but still cannot exceed the original LGBPN model, so we cannot conclude which branch is redundant for LGBPN model.

Both approaches of interchanging BNN component of LGBPN and SSID (L-1, S-1) underperform in terms of PSNR and SSIM. The result did not align with our interchangability hypothesis and both BNN components are customized for each model, so we are unable to divide the research work into parts.

5 Future Works

In the future, we plan to apply both self-supervised denoising models to RAW images instead of sRGB images so as to denoise the images from the very beginning of the image pipeline and investigate whether they can perform as good as they are with sRGB images. This is because image demosaicking adds a lot of errors to the RAW images. As there are no RGB pixels in RAW images, we cannot compute the pixel-wise noise correlation map in a similar fashion as proposed in Section 3.2. To perform such analysis, we might have to search for different priors and ways to estimate noise correlation. Besides, as we observe underperforming SSIM results in comparison to PSNR in Section 4.2.2. we believe it would be a good idea to incorporate perceptual loss [8] to the loss functions of both models. This forces

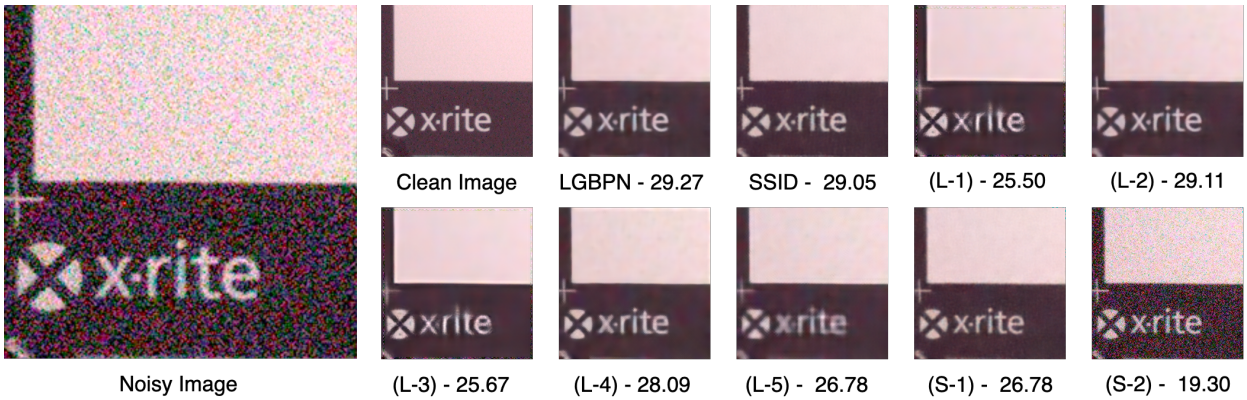


Fig. 8: Visual quality comparison of all our approaches

the model to take into account human perception, such as content and style discrepancies.

6 Conclusion

In this paper, we investigated pixel-wise noise correlation on a size of 21x21 as image prior and concluded that noise are mostly correlated within a 9x9 receptive field and there is an unexpected fluctuation pattern of noise out side the 9x9 receptive field. Besides, we reworked code provided SSID into a well-design library to allow easy switching of BNN and LAN models. Apart from that, we performed additional ablation studies to identify model significance and interpretability, as well as replaced model components with other state-of-the-art parts to explore model interchangeability. While our results did not surpass those presented in the two recent papers from the latest Conference on Computer Vision and Pattern Recognition (CVPR), this outcome aligns with our expectations considering the high standards set by those contributions.

Acknowledgments

We would like to thank Prof. David Lindell and all TAs for their support of this project. In addition, we would like to thank Dr. Michael S. Brown, whom we were fortunate enough to meet during the presentation day. He is one of the authors of the SIDD dataset [1] we use, and he provided insightful ideas on our projects. In particular, he commented that the checkboard patterns might be due to defects when demosaicking the RAW images into SRGB ones. He also recommended us to denoise images from the RAW and add perceptual loss [8]. Unfortunately, this report is due 3 days after our presentation, and we do not have sufficient time to do so, which is why we listed them in future works.

References

[1] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] B. Brummer and C. D. Vleeschouwer, "Natural image noise dataset," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2019. [Online]. Available: https://doi.org/10.1109%2Fcvprw.2019.00228

[3] W. Lee, S. Son, and K. M. Lee, "Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 725–17 734.

[4] Z. Wang, Y. Fu, J. Liu, and Y. Zhang, "Lg-bpn: Local and global blind-patch network for self-supervised real-world denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 18 156–18 165.

[5] J. Li, Z. Zhang, X. Liu, C. Feng, X. Wang, L. Lei, and W. Zuo, "Spatially adaptive self-supervised learning for real-world image denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 9914–9924.

[6] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void - learning denoising from single noisy images," November 2018.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017.

[8] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.