An objective comparison and analysis of the performance of the BM3D and DnCNN image processing algorithms on denoising

Erin Shappell
BioEngineering

Robert Pritchard
BioEngineering

***Abstract*—This project presents a detailed performance comparison between block-matching 3D filtering and a denoising convolutional neural network. The denoising methods were compared across 7 image quality assessment metrics: PSNR, SSIM, CW-SSIM, UNIQUE, MSUNIQUE, CSV, and SUMMER. These metrics were used to evaluate performance across 5 datasets: Set12, CURE-OR, CURE-TSR, CURE-TSD, and SIDD. While of these datasets include 3 channel colored images, we assessed denoising performance on gray scale images only.**

***Index Terms*—BM3D, DnCNN, Denoising, IQA**

I. Introduction

For this project, we compare the performance of two different denoising algorithms: Block Matching and 3D Filtering (BM3D) and the Denoising Convolutional Neural Network (DnCNN) on five datasets containing a rich variety of noise sources and noise levels. There is no one denoising method that is a gold standard for all noise types; with this in mind, we chose to compare denoising methods that are different in their filtering approach (i.e., collaborative filtering in BM3D versus deep learning in DnCNN) to determine which method is more suitable for specific noise types and/or noise levels.

Equally important to our selection of denoising methods is the data to which the methods were applied. Five datasets were provided for this project: CURE-OR, CURE-TSR, CURE-TSD, SIDD, and Set12. The first four datasets were chosen for their extensive representation of challenging noise types, while Set12 was chosen as a smaller baseline dataset to which we could apply our own noise for additional testing and comparison. CURE-OR contains 1,000,000 noisy images of 100 different objects taken at various orientations [1]. CURE-TSR contains over 2,000,000 noisy images of traffic signs in both real-world and simulated environments [2]. Similar to CURE-TSR, CURE-TSD contains noisy videos taken from the POV of a moving vehicle, where multiple traffic signs are present at different timestamps of each video [3]. SIDD contains over 30,000 noisy images taken from a variety of different smartphone cameras under three different lighting conditions [4]. Finally, the Set12 dataset consists of 12 challenging images that were introduced for testing image denoising and enhancement algorithms [5].

We expected DcNN to outperform BM3D across the board on Set12 based on the results presented in [5]. Zhang et al. reported that even the DnCNN trained for general gaussian denoising could outperform BM3D tuned for the particular noise level. Our findings did not support this hypothesis. We also expected that DnCNN would generalize to the outer noise types across the datasets better than BM3D. However, we did not expect either of them to work well on large area noise types such as dirty lenses or the weather conditions in CURE-TSD and CURE-TSR. Along similar lines we did not expect either method to correct for poor contrast or lighting issues since they were both designed originally for gaussian denoising. While these trends loosely held, which denoising method performed best on a type of noise often was not consistent across datasets.

II. Methods

A. Data Selection and Augmentation

We summarize the results of denoising images and videos from five datasets (CURE-OR, CURE-TSR, CURE-TSD, SIDD, and Set12) in Tables I-V. The discussion in this paper is limited to the images and videos containing only extreme levels of noise, 1 (weakest noise) and 5 (strongest noise). Levels 2-4 are intermediate levels of noise, which we felt would not add much to our final comparison of denoising success given the limits on project timeframe and report length; therefore, we have omitted these levels from the discussions in this report. Additionally, while CURE-OR provides a rich variety of both different camera sources (e.g., DSLR, HTC, iPhone, LG, and Logitech) and levels of background information within images (3D, texture, and white), we will only discuss the results from the HTC 3D images for each provided noise source. Similarly for SIDD, which hosts images taken by five different smartphone cameras, we limit our discussion to the results from denoising Samsung Galaxy S6 images.

We report data from each set as follows. For CURE-OR we report the results of denoising levels 1 and 5 of the underexposure, overexposure, contrast, dirty lens, and salt and pepper noise types (Table II). For CURE-TSR we report the results of denoising levels 1 and 5 of the codec error, darkening, dirty lens, exposure, gaussian blur, haze, lens blur, noise, rain, shadow, and snow noise types (Table III). For CURE-TSD we report the results of denoising a selection of videos from the test set, which include levels 1 and 5 of the lens blur, codec error, darkening, dirty lens, exposure, gaussian blur, noise, rain, shadow, snow, and haze noise types (Table IV). Finally, for SIDD we report the results of denoising images from the three provided exposure levels (low light, normal brightness, and high exposure) across eight of the provided scenes (Table V).

Set12 is different from the other datasets in this project because they are provided without any significant noise. We selected to add 5 noise conditions to these images: additive gaussian noise with a sigma of 15, additive gaussian noise with a sigma of 30, poisson noise, salt and pepper noise, and speckle noise. These noise types were selected because they are all very common noise types, which were discussed in class. The variances for the gaussian noise were selected from the literature. Most studies such as [5] report performance on gaussian denoising with σ around 15 and 30. Examples of the noisy Set12 images may be viewed in Figure 1, and the denoising results for Set12 may be viewed in Table I.

We initially attempted to process the frames of the CURE-TSD videos and images from SIDD in their raw form, but we quickly realized that the runtime of our selected denoising techniques was too long given the project timeframe. For reference, one folder of 300 1628x1236 CURE-TSD frames took over 3 hours to process using BM3D. To mediate this runtime issue while maximizing processing throughput, we chose to resize the frames and images from CURE-TSD and SIDD by factors of ½ and ¼, respectively using MATLAB's imresize method.

B. BM3D

Block matching and 3D filtering (BM3D) is a denoising method that has been proven successful in many applications including video denoising and superresolution [6], [7]. We selected this method based on (1) its prevalence as a baseline denoising method [8] and (2) its ease of use: a Python implementation is freely available [9], [10]. The main hallmark of BM3D is how it selects the groups of image pixels to denoise together; image patches are groups based on similarity, but unlike previous methods, these groups may overlap. Each patch group is then transformed into the 3D wavelet domain, stacked, and then denoised by thresholding or filtering the wavelet coefficients. The denoised patch groups are transformed back and weighted appropriately to account for patch overlaps to obtain the final denoised image.

The main parameter for BM3D is the noise power spectral density (PSD). While we do have access to the ground truth images for each of the five datasets, we chose to estimate this parameter without using ground truth as a more authentic challenge. We applied a fast noise variance estimation method [11] to the first image in each noise type and noise level, and then applied this estimate to the BM3D denoising of all images containing that respect noise type and level within the dataset. The distribution of the estimated noise PSD values for each dataset may be viewed in Figure 2.

C. DnCNN

Deep learning approaches have become very popular for image denoising due to their versatility and speed. One of the simplest architectures for deep learning are convolutional neural networks (CNNs). The model we used in this project is simply called the Denoising Convolutional Neural Network, originally proposed by Zhang et al. [5]. This model is provided pre-trained in the MATLAB deep learning toolbox. The provided model consists of 20 convolutional layers each using 64 3x3 filters. There are batch normalization and ReLU activation layers after each convolutional layer. The batch normalization layers speed up model training, increase the performance, and reduce its sensitivity to the initialization during training. Matlab does not provide details on the exact training paradigm used for their model, but Zhang et al. trained models for individual levels of gaussian noise ($\sigma = 15, 25, 50$) and one more model for general gaussian denoising. For training the general model the training images contained a range of noise levels of 0 to 55. The authors reported that DnCNN could outperform BM3D even when trained for gaussian noise in general, however they did not explore how the models performed on non gaussian noise and only used PSNR values for the comparison. Based on the performance we see from the model in MATLAB.

The trained model has no parameters to change during inference, however there is a minimum and maximum image size that can be used. Convolutional layers can generally handle images of different sizes, however if the images are too small there are not enough values to pass to the next layer. Only the CURE-TSR dataset had images too small for the model. This was solved by simply upsampling the images to pass into the model and then downsampling the output back to the original size. The images in the CURE-TSD and SIDD datasets had images too large to fit the model into GPU memory. This was solved in two different ways: downsampling the images to half and one quarter the size respectively using imresize, and block processing the images in 256x256 patches. After preliminary results from both methods were seen to be roughly equivalent based on the IQA metrics. However, the block processing took multiple times as long to run. Due to time constraints and the desire to match the processing of the BM3D method only the results from downsampling the images are reported here.

D. IQA Metrics

To measure how well each algorithm denoised the provided image and video sets in an objective and thorough manner, we will use the following Image Quality Assessment (IQA) metrics: PSNR, SSIM, CW-SSIM, UNIQUE, MS-UNIQUE, CSV, and SUMMER. PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity) are two common IQA metrics that measure the ratio of signal power to noise power and change in local image structure, respectively. CW-SSIM (Complex Wavelet SSIM) measures structural information change while accounting for consistent phase shifts that do not necessarily disrupt image structure [17]. We used a publically available implementation of CW-SSIM found at [16] UNIQUE, MS-UNIQUE, and CSV are loosely modeled after the human visual system: UNIQUE and MS-UNIQUE each utilize an unsupervised approach to determine image quality [13], [14], while CSV's structure is modeled after retinal ganglion cells using a Laplacian of Gaussian method [12]. Finally, SUMMER estimates image quality by comparing the error signal spectra of a noisy and reference image [15]. Taken together, the results from all of these IQA metrics will provide a rich, objective measure of how well each denoising algorithm restored the original quality of each image we analyzed.

III. Results

A. Set12

In every noise condition other than speckle BM3D consistently outperformed DnCNN Table I. While the difference in each metric alone was very small, BM3D performed better on every single metric for both levels of gaussian noise and poisson noise. In contrast, on speckle noise DnCNN was more effective across the IQA metrics. This was a somewhat surprising result as we expected poisson or salt and pepper to be the most difficult for BM3D to adjust to since it is designed for Gaussian denoising.

B. CURE-OR

As shown in Table II the results for the CURE-OR dataset was less consistent than Set12. The results for UNIQUE and MS-UNIQUE are all extremely small and difficult to interpret. CSV also shows almost no change across all of the noise conditions. SUMMER was able to distinguish between the performance of the methods in the salt and pepper noise, but does not provide significant insight in the other conditions. For salt and pepper noise BM3D has much higher SUMMER values for both low (BM3D: 4.806;DnCNN: 1.573) and high (BM3D: 4.75; DnCNN: 2.222) levels. For underexposure, overexposure, and contrast only PSNR, SSIM, and CW-SSIM showed much of a difference between the denoising methods. DnCNN performed slightly better than BM3D on the low-noise conditions, but BM3D performed significantly better on the high-noise conditions. Neither approach performed better on all 3 metrics in any condition. BM3D tended to perform better on PSNR, while DnCNN performed better on SSIM in general. CW-SSIM peformance was more variable. Overall, the differences in performance were small across conditions. Neither method performed very well on dirty lens noise. However, based on only PSNR, and SSIM DnCNN performed slightly better.

C. CURE-TSR

The full IQA results for CURE-TSR are provided in Table III. Across the many noise conditions a common pattern is DnCNN performing better on the low noise level, but BM3D performs better on the images with higher noise. This pattern is seen in darkening, exposure, gaussian blur, haze, and lens blur. Within these noise categories two data points stand out: BM3D's SSIM value for high lens blur (0.739) compared to DnCNN (0.158), and BM3D's PSNR value for level 5 exposure noise (40.466) compared to DnCNN (27.780). DnCNN performed better in both low and high noise conditions for codec error, noise, and shadow. The other common performance pattern in the CURE-TSR set is DnCNN performs better on all metrics except PSNR. For images with rain, snow, and dirty lenses DnCNN performed better across all the metrics except PSNR.

D. CURE-TSD

The results for the CURE-TSD dataset shown in Table IV show very similar performance as the CURE-TSR dataset. DnCNN again performs better on both low and high codec error, noise, and rain. DnCNN also performs better on images with shadow in PSNR and SSIM, but BM3D achieved a higher CW-SSIM value. The motif of DnCNN performing better on low levels of the noise and BM3D performing better on high noise levels is seen again in this datset. This trend is seen for images with darkening, dirty lenses, poor exposure, and haze. For the snowy images DnCNN performed better on the low level but the two denoising methods perform about equally on the high level.

E. SIDD

Table V shows the full results of denoising the SIDD dataset. Overall, all of the IQA metrics were higher on thise dataset. For most of the scenes and noise types BM3D performs better for PSNR and SSIM but DnCNN peforms better for CW-SSIM, UNIQUE, MS-UNIQUE, and SUMMER. Only three conditions break this trend: Scene 8 low light, scene 5 low light, and scene 4 normal light. DnCNN performed better across all metrics for scenes 8 and 5, while BM3D performed better across all metrics for scene 4.

F. CURE-TSR and CURE-TSD CNN Recognition

Using existing code as a reference [18], we trained a CNN classifier on the German Traffic Sign Recognition dataset to use to evaluate the recognition of the denoised images and videos. Given our limited time after debugging denoising methods and downloading and processing the large datasets, we decided to keep the original code's training set rather than train a model from scratch on the CURE-TSR and CURE-TSD sets. After preliminary testing, we noted that the classification accuracy of this CNN did not improve with either denoising algorithm. Due to this lack of improvement and time constraints, we focused on our discussion of the IQA metrics instead of classification performance. We expect that a CNN classifier trained on the CURE-TSR and CURE-TSD datasets may show some difference in classification accuracy for the denoised images.

IV. Discussion

Overall DnCNN and BM3D were able to achieve similar performance across datasets and IQA metrics. CSV is the only IQA metric that was not able to provide any interpretable information. We suspect this is due to our images being in gray scale. While UNIQUE, MS-UNIQUE, and SUMMER also use color information, converting the gray scale images to RGB seems to have been enough for them to still be somewhat meaningful. We focus our discussion on the metrics that appear to be the most meaningful for our gray scale images: PSNR, SSIM, and CW-SSIM. These metrics were selected because a relatively large range of values appear across datasets and noise conditions and larger differences are seen between the two denoising methods.

A. BM3D

The BM3D algorithm performed best on noise types that were uniformly sparse throughout (e.g., speckle and salt and pepper noise), but struggled to denoise the more extreme noise types (e.g., dirty lens, rain). This result was expected, as the BM3D architecture is designed to match together similar blocks and denoise them as a stack; if the noise is sparsely distributed, then it is likely that the ratio of noise to information within any given block is low. For noise types in which the noise contaminates more of the original image data per block, it is less likely that the noise will be removed.

One potential source of BM3D's performance issues on some datasets may be with the method in which we assigned a noise PSD value to each noise source and level. As mentioned previously, to save computation time, we estimated noise PSD by only calculating an estimate for the first image in each noise type and noise level and using that estimate for all images within that folder. Had we estimated noise on a per-image basis, we would expect the performance of BM3D to increase at the cost of a higher runtime.

B. DnCNN

DnCNN performed well on gaussian noise across datasets as we expected. It's performance on Set12 images with speckle noise added was slightly surprising since speckle noise was not included in its training set. However, since speckle noise is multiplicative Gaussian noise it seems the model was able to correct it to some degree based on its training for various levels of additive gaussian noise. DnCNN did not translate to salt and pepper noise as well. DnCNN was able to acheive decent values for the IQA metrics for salt and pepper noise, but as shown in Figure 2, there are still large visual artifacts. DnCNN struggled with the extreme noise types such as dirty lens and rain much like BM3D. The IQA metrics still showed DnCNN had some positive effect on these kinds of noise, however most of these improvements just came from a smoothing effect. A more powerful technique would be needed to remove these occlusions from the images.

C. Comparison

The trend of DnCNN performing better thant BM3D on low levels of most noise types but worse on high levels is difficult to interpret. The simplest explanation is that DnCNN is better able to generalize to noise types but the noise level parameter in BM3D allows it to better handle the extreme cases. DnCNN does not provide any way to tune it to new noise without training the model more on new data.

A more detailed comparison between the two metrics is shown in Figure 3. The SSIM images show whiter pixels where the denoised image was closer to the clean image corresponding to where the denoising methods performed the best. Surprisingly both methods performed best around edges in the images. BM3D's performance is very focused around the edges of the table and boat in the examples images. DnCNN shows high values around the edges as well, but overall there are a lot more white areas across the rest of the image as well. These SSIM images also help explain why the denoising methods performed better on the salt and pepper noise in the Set12 dataset compared to the CURE-OR dataset. The amount of texture and edges in the Set12 images increased performance. The CURE-OR dataset contains only images of that couch and table, the only difference image to image is what is on the table. This means that every image in the CURE-OR dataset has a lot of low frequency areas which make hiding the salt and pepper noise more difficult.

D. Limitations

The sheer size of all of the datasets required in this project put a strain on computing resources and time. Retraining the DnCNN network for each specific type of noise would have been the most effective way to use CNNs for denoising all the datasets. There are also more powerful deep learning architectures that have been proposed. Transformers are currently very popular in machine learning, but they take a long time to train and validate. Code and training data was set up to train SUNET a U-NET based architecture that includes Swin-transformer blocks [19], but took too long to train on a 4GB GPU.

A project just focused on one CURE-OR could also have attempted to extend the denoising methodologies more to handle dirty lens noise and other extreme types. This would have involved incorporating much more complicated algorithms resulting in longer runtimes per image.

V. Conclusion

We were able to compare the performance of BM3D and DnCNN across all the noise types present in CURE-OR, CURE-TSR, CURE-TSD, and SIDD. We also added 5 different noise conditions to Set12 including poisson noise which is not specifically present in any of the other datasets. Both metrics worked best on simpler wide spread noise such as gaussian, speckle, and salt and pepper, while struggling with more drastic changes to the images such as dirty lenses.

VI. Code

All code used in this project made be found at the following GitHub repository: https://github.com/erinshappell/ECE6258_Project_Fall2023

REFERENCES

[1] D. Temel, J. Lee, and G. AlRegib, “CURE-OR: Challenging unreal and real environments for object recognition,” 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, Florida, USA, 2018.

[2] D. Temel, G. Kwon*, M. Prabhushankar*, and G. AlRegib, “CURE-TSR: Challenging unreal and real environments for traffic sign recognition,” in Neural Information Processing Systems (NIPS) Workshop on Machine Learning for Intelligent Transportation Systems (MLITS), Long Beach, U.S., December 2017.

[3] D. Temel, M-H. Chen, and G. AlRegib, ”Traffic Sign Detection under Challenging Conditions: A Deeper Look Into Performance Variations and Spectral Characteristics,” IEEE Transactions on Intelligent Transportation Systems, 2019.

[4] A. Abdelhamed, S. Lin, and M. S. Brown, “A high-quality denoising dataset for Smartphone cameras,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. doi:10.1109/cvpr.2018.00182

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, 2017. doi:10.1109/tip.2017.2662206

[6] A. Buades, B. Coll, and J.-M. Morel, “Non-local means denoising,” Image Processing On Line, vol. 1, pp. 208–212, 2011. doi:10.5201/ipol.2011.bcm_nlm.

[7] M. Lebrun, “An analysis and implementation of the BM3D image denoising method,” Image Processing On Line, vol. 2, pp. 175–213, 2012. doi:10.5201/ipol.2012.l-bm3d

[8] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” Visual Computing for Industry, Biomedicine, and Art, vol. 2, no. 1, 2019. doi:10.1186/s42492-019-0016-7

[9] Y. Mäkinen, L. Azzari and A. Foi, ”Collaborative Filtering of Correlated Noise: Exact Transform-Domain Variance for Improved Shrinkage and Patch Matching,” in IEEE Transactions on Image Processing, vol. 29, pp. 8339-8354, 2020, doi: 10.1109/TIP.2020.3014721.

[10] “BM3D,” PyPI, https://pypi.org/project/bm3d/ (accessed Dec. 2, 2023).

[11] J. Immerkær, “Fast noise variance estimation,” Computer Vision and Image Understanding, vol. 64, no. 2, pp. 300–302, 1996. doi:10.1006/cviu.1996.0060

[12] D.Temel and G. AlRegib, “CSV: Image Quality Assessment Based on Color, Structure, and Visual System”, the Signal Processing: Image Communication Journal, vol.48, pp.92-103, October 2016

[13] D. Temel, M. Prabhushankar and G. AlRegib, ”UNIQUE: Unsupervised Image Quality Estimation,” in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1414-1418, Oct. 2016.

[14] M. Prabhushankar, D. Temel, and G. AlRegib, MS-UNIQUE: Multi-model and Sharpness-weighted Unsupervised Image Quality Estimation, the Electronic Imaging, Image Quality and System Performance XIV, Burlingame, California, USA, Jan. 29 Feb. 2, 2017.

[15] D. Temel and G. AlRegib, “Perceptual Image Quality Assessment through Spectral Analysis of Error Representations ,” Signal Processing: Image Communication, Volume 70, 2019, Pages 37-46.

[16] ”pyssim”, Jeff Terrace, https://github.com/jterrace/pyssim

[17] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik and M. K. Markey, ”Complex Wavelet Structural Similarity: A New Image Similarity Index,” in IEEE Transactions on Image Processing, vol. 18, no. 11, pp. 2385-2401, Nov. 2009, doi: 10.1109/TIP.2009.2025923.

[18] “Traffic Sign Classifier,” Google Colab, https://colab.research.google.com/github/CleanPegasus/Traffic-Sign-Classifier/blob/master/Traffic_Sign_Classifier.ipynb (accessed Nov. 15, 2023).

[19] C. -M. Fan, T. -J. Liu and K. -H. Liu, ”SUNet: Swin Transformer UNet for Image Denoising,” 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 2333-2337, doi: 10.1109/ISCAS48785.2022.9937486.

APPENDIX

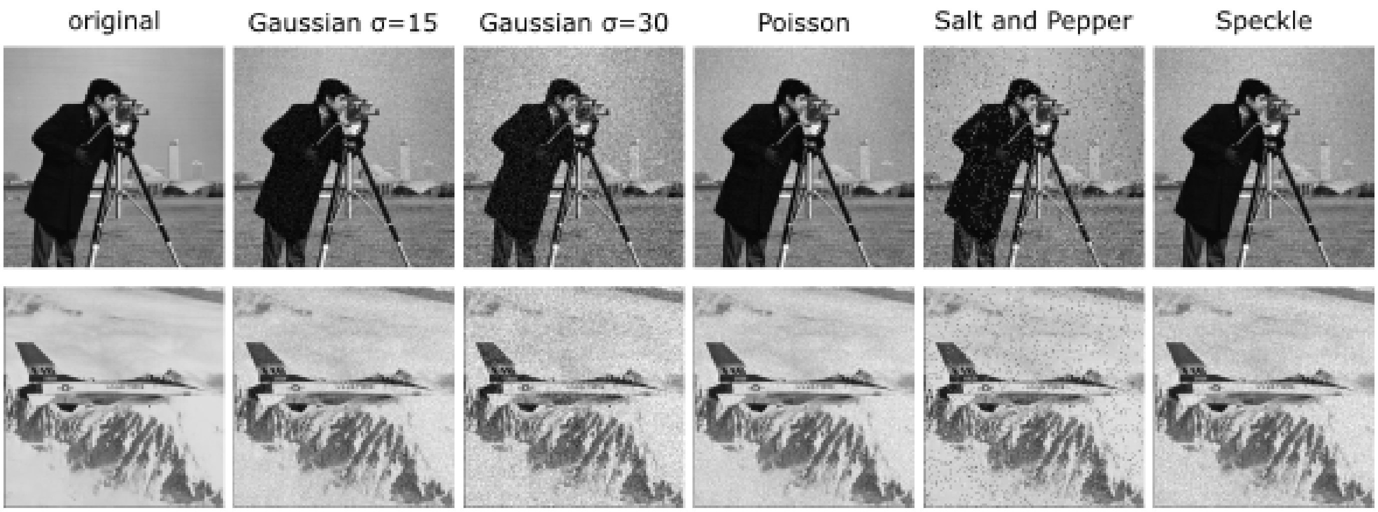

Fig. 1. Examples of the noise added to Set12

Fig. 2. Example of the visual artifacts left after denoising salt and peper noise with DnCNN.

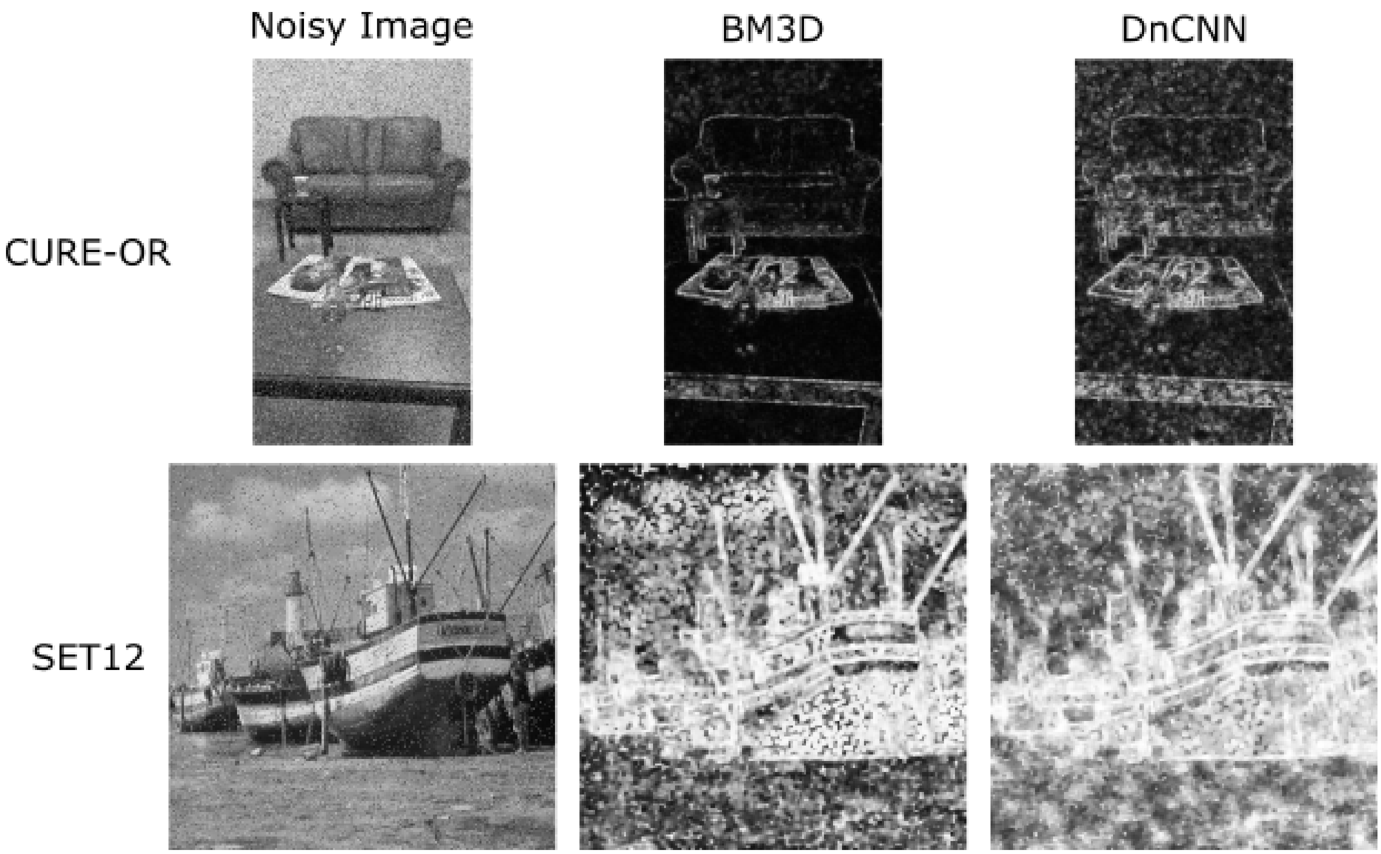

Fig. 3. Comparison of SSIM images for denoised images from the Salt and pepper condition of CURE-OR and SET12. The image from CURE-OR is from the level 1 salt and pepper noise set. White regions indicate high local SSIM values and therefore areas where the denoising methods performed well.

		PSNR	SSIM	CW-SSIM	UNIQUE	MS-UNIQUE	CSV	SUMMER
Gaussian ($\sigma = 15$)	BM3D	**33.637**	**0.897**	**0.975**	**0.700**	**0.745**	0.999	**4.977**
	DnCNN	33.090	0.880	0.973	0.685	0.732	0.999	4.976
Gaussian ($\sigma = 30$)	BM3D	**31.887**	**0.833**	**0.954**	**0.516**	**0.584**	0.999	**4.961**
	DnCNN	31.358	0.795	0.795	0.477	0.550	0.999	4.957
Poisson	BM3D	**34.430**	**0.914**	**0.980**	**0.759**	**0.794**	0.999	4.981
	DnCNN	33.946	0.904	0.978	0.757	0.793	0.999	4.981
Salt & Pepper	BM3D	**32.971**	0.602	**0.930**	**0.442**	**0.526**	0.999	4.951
	DnCNN	32.378	**0.616**	0.926	0.417	0.493	0.999	**4.945**
Speckle	BM3D	33.324	0.868	0.973	0.715	0.757	0.999	4.977
	DnCNN	**33.536**	**0.891**	**0.974**	**0.720**	**0.761**	0.999	**4.978**

TABLE I
IQA RESULTS ON THE SET12 DATASET

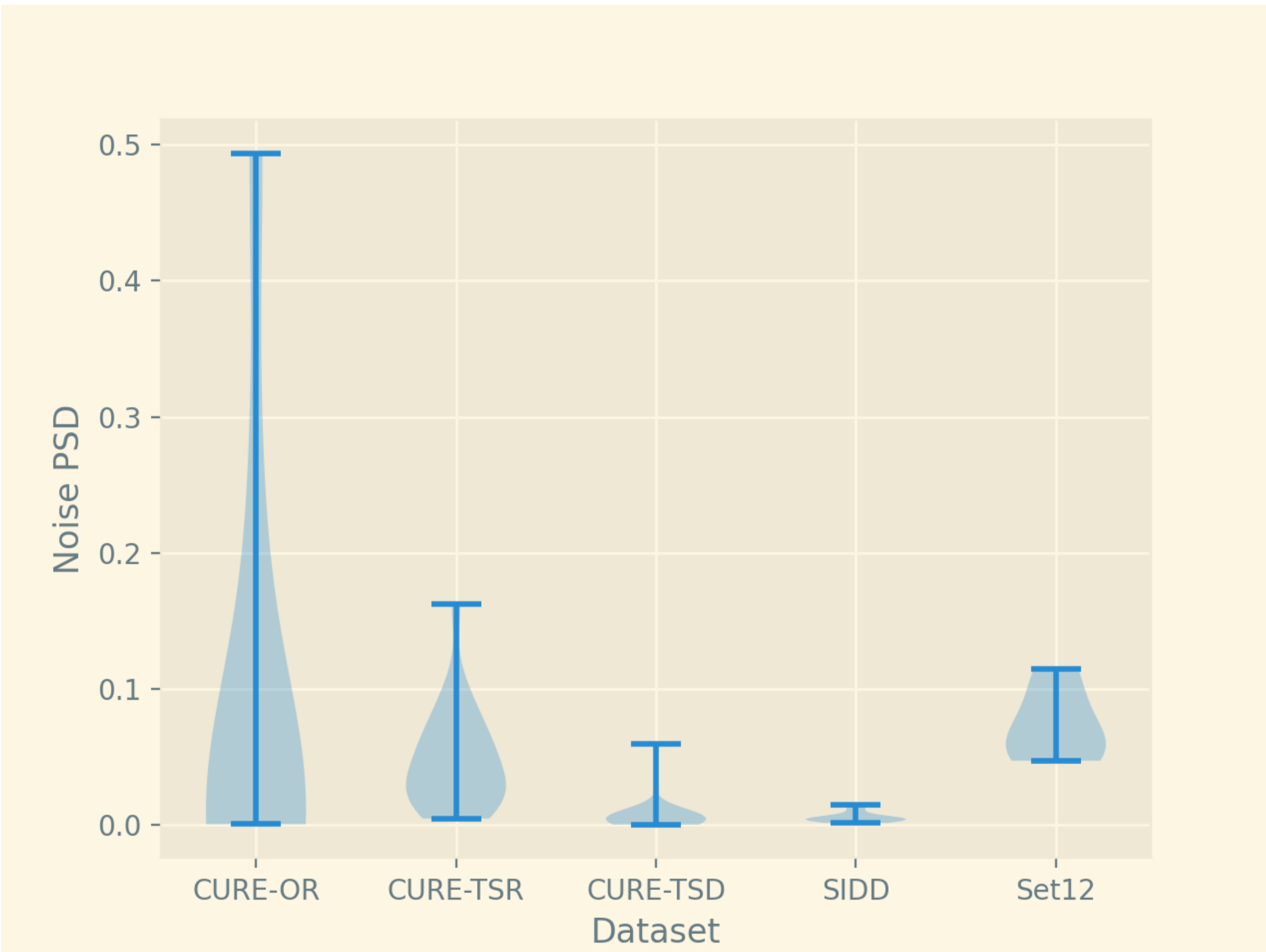

Fig. 4. **Distribution of the noise PSD values estimated by the method in [11] for the BM3D denoising method**. Noise PSD was estimated for each noise type and noise level for each dataset. Higher levels of reported noise did not necessarily result in a higher noise PSD, suggesting that this method may require improvement (see text).

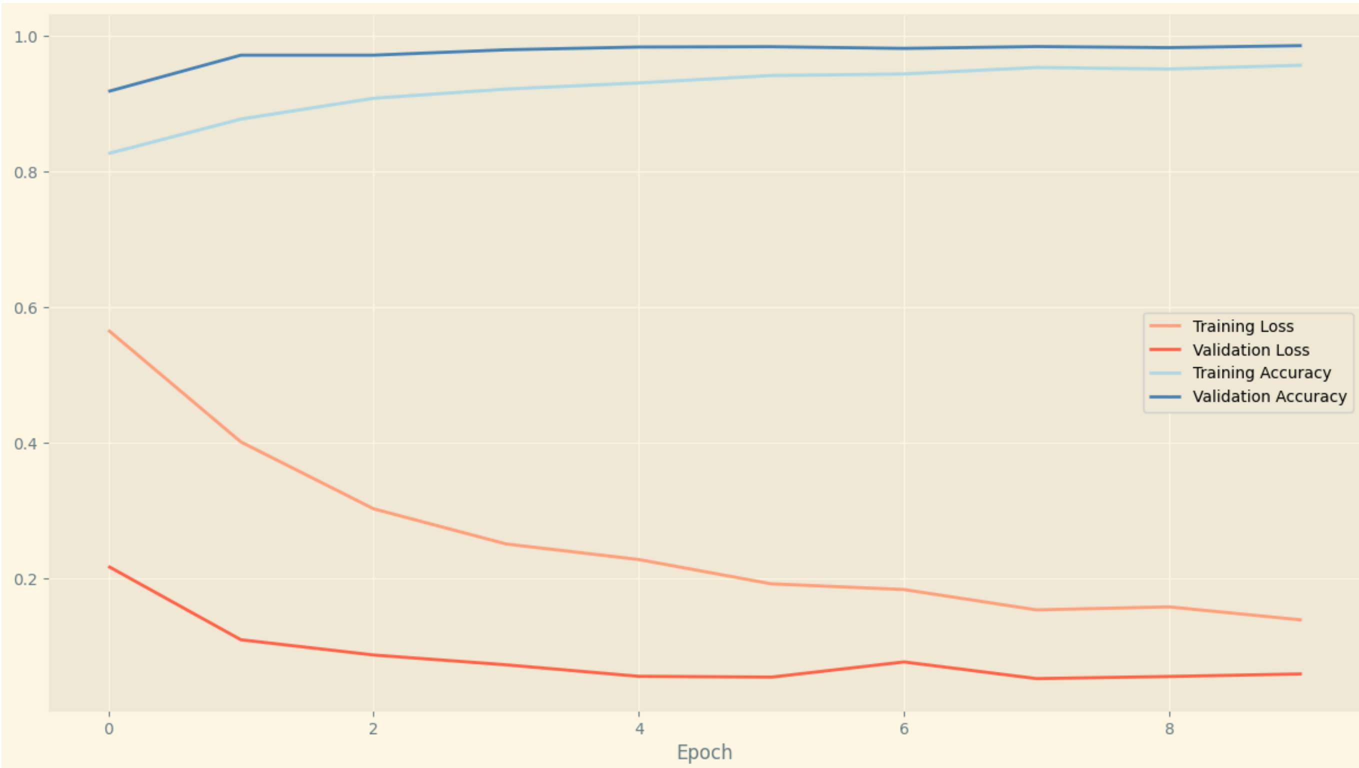

Fig. 5. **Loss and accuracy results from training a Traffic Sign Classifier using Keras**. The model was trained using the German traffic signs dataset, with a train/validation/test split of 67:9:24.

			PSNR	SSIM	CW-SSIM	UNIQUE	MS-UNIQUE	CSV	SUMMER
Underexposure	level 1	BM3D	27.511	0.265	**0.655**	**1.51E-09**	7.90E-04	0.995	2.378
		DnCNN	**27.646**	**0.547**	0.654	7.77E-10	**8.70E-04**	0.995	**2.395**
	level 5	BM3D	**30.218**	0.048	**0.165**	1.35E-04	**1.03E-04**	0.996	0
		DnCNN	27.749	**0.089**	0.16	**2.58E-04**	9.91E-05	0.996	0
Overexposure	level 1	BM3D	**28.343**	0.62	0.779	**2.60E-03**	**0.041**	0.996	4.666
		DnCNN	27.782	**0.873**	**0.879**	1.33E-03	0.03	0.996	**4.676**
	level 5	BM3D	**37.42**	**0.9**	0.171	6.13E-09	**5.00E-03**	0.995	**2.621**
		DnCNN	27.913	0.413	**0.179**	**1.99E-08**	4.60E-03	0.995	2.605
Contrast	level 1	BM3D	**28.746**	0.38	0.649	1.00E-04	**0.012**	0.996	4.697
		DnCNN	28.023	**0.704**	**0.758**	**8.00E-03**	8.65E-03	**0.997**	**4.745**
	level 5	BM3D	**30.051**	0.339	**0.511**	3.82E-06	6.10E-03	0.995	**4.74**
		DnCNN	27.667	**0.432**	0.498	**1.05E-02**	**1.05E-02**	**0.996**	4.321
Dirty Lens	level 1	BM3D	27.813	0.517	**0.691**	2.21E-02	0.117	0.997	4.476
		DnCNN	**29.025**	**0.742**	0.236	**1.16E-01**	**0.236**	**0.998**	**4.803**
	level 5	BM3D	27.79	0.125	0.435	8.17E-06	**6.59E-03**	0.995	**4.351**
		DnCNN	**27.93**	**0.256**	0.435	**8.39E-06**	6.50E-03	0.995	4.345
Salt & Pepper	level 1	BM3D	27.815	**0.499**	**0.745**	**0.208**	**0.347**	**0.998**	**4.806**
		DnCNN	**29.626**	0.474	0.379	1.11E-05	0.012	0.995	1.573
	level 5	BM3D	**29.289**	**0.469**	**0.683**	0.113	**0.212**	**0.998**	**4.75**
		DnCNN	27.917	0.042	0.353	**1.468**	2.10E-03	0.995	2.222

TABLE II
IQA RESULTS ON THE CURE-OR DATASET

			PSNR	SSIM	CW-SSIM	UNIQUE	MS-UNIQUE	CSV	SUMMER
Codec Error	Level 1	BM3D	27.762	0.169	0.689	0.304	0.369	0.996	4.241
		DnCNN	**29.016**	**0.348**	**0.690**	**0.313**	**0.377**	0.996	**4.765**
	Level 5	BM3D	27.820	**0.240**	0.593	0.268	0.306	0.995	3.625
		DnCNN	**28.455**	0.090	**0.603**	**0.278**	**0.316**	0.995	**4.575**
Darkening	Level 1	BM3D	27.599	0.110	0.929	0.270	0.374	0.996	4.335
		DnCNN	**27.753**	**0.890**	**0.942**	**0.275**	**0.383**	**0.997**	**4.679**
	Level 5	BM3D	**30.517**	**0.049**	0.107	0.261	0.263	0.996	0
		DnCNN	27.582	0.029	**0.117**	**0.263**	**0.264**	0.996	0
Dirty Lens	Level 1	BM3D	27.747	0.106	0.972	0.766	0.832	0.999	4.752
		DnCNN	**33.958**	**0.972**	**0.982**	**0.817**	**0.870**	0.999	**4.956**
	Level 5	BM3D	**28.282**	0.303	0.693	0.327	0.442	0.996	3.595
		DnCNN	27.679	**0.545**	**0.697**	**0.329**	**0.445**	0.996	**3.646**
Exposure	Level 1	BM3D	**28.731**	0.148	0.894	0.293	0.400	0.996	4.916
		DnCNN	27.670	**0.782**	**0.897**	**0.296**	**0.405**	0.996	**4.939**
	Level 5	BM3D	**40.466**	**0.797**	0.358	0.246	0.292	0.995	1.216
		DnCNN	27.780	0.113	**0.363**	0.246	**0.293**	0.995	**1.266**
Gaussian Blur	Level 1	BM3D	27.721	0.201	0.841	0.361	0.478	0.997	4.461
		DnCNN	**28.994**	**0.600**	**0.842**	**0.362**	**0.479**	0.997	**4.510**
	Level 5	BM3D	**28.214**	**0.498**	0.599	0.267	0.331	0.996	**2.083**
		DnCNN	27.948	0.178	0.599	0.267	0.331	0.996	2.071
Haze	Level 1	BM3D	27.750	0.217	0.880	0.494	0.604	0.998	4.113
		DnCNN	**28.059**	**0.836**	**0.896**	**0.502**	**0.612**	0.998	**4.297**
	Level 5	BM3D	**28.113**	**0.790**	0.441	0.274	0.362	0.996	0.273
		DnCNN	28.079	0.240	**0.452**	0.276	**0.368**	0.995	0.213
Lens Blur	Level 1	BM3D	27.721	0.158	0.909	**0.462**	0.558	0.998	4.541
		DnCNN	**29.569**	**0.739**	**0.910**	0.460	**0.586**	0.998	**4.592**
	Level 5	BM3D	**28.098**	**0.388**	0.635	0.266	**0.332**	0.996	2.764
		DnCNN	28.041	0.173	0.635	0.266	0.331	0.996	**2.951**
Noise	Level 1	BM3D	27.656	0.111	**0.967**	**0.741**	**0.816**	0.998	4.671
		DnCNN	**30.420**	**0.905**	0.963	0.731	0.804	**0.999**	**4.996**
	Level 5	BM3D	27.806	0.149	**0.736**	**0.304**	0.413	**0.997**	4.631
		DnCNN	**27.962**	**0.403**	0.732	0.301	**0.408**	0.996	**4.867**
Rain	Level 1	BM3D	**28.119**	0.215	0.722	0.369	0.496	0.997	3.646
		DnCNN	27.741	**0.597**	**0.733**	**0.379**	**0.506**	0.997	**3.888**
	Level 5	BM3D	**28.527**	0.350	0.567	0.307	0.421	0.996	2.255
		DnCNN	27.662	**0.402**	**0.578**	**0.313**	**0.431**	0.996	**2.434**
Shadow	Level 1	BM3D	27.643	0.099	0.959	0.610	0.723	0.998	4.637
		DnCNN	**31.232**	**0.958**	**0.967**	**0.642**	**0.746**	0.998	**4.915**
	Level 5	BM3D	27.634	0.091	0.694	0.431	0.457	0.997	3.492
		DnCNN	**30.609**	**0.611**	**0.699**	**0.437**	**0.461**	0.997	**3.493**
Snow	Level 1	BM3D	28.088	0.142	0.958	0.715	0.797	0.999	4.461
		DnCNN	**32.559**	**0.955**	**0.980**	**0.826**	**0.882**	0.999	**4.959**
	Level 5	BM3D	**29.492**	0.313	0.682	0.252	0.306	0.994	4.312
		DnCNN	27.989	**0.422**	**0.693**	**0.253**	**0.307**	0.994	**4.613**

TABLE III
IQA RESULTS ON THE CURE-TSR DATASET

			PSNR	SSIM	CW-SSIM	UNIQUE	MS-UNIQUE	CSV	SUMMER
Codec Error	Level 1	BM3D	28.126	0.403	0.715	**0.140**	0.246	0.997	4.596
		DnCNN	**31.080**	**0.632**	0.715	0.100	**0.250**	0.997	**4.597**
	Level 5	BM3D	28.116	0.414	0.633	**0.044**	0.120	0.996	4.289
		DnCNN	**30.420**	**0.543**	**0.634**	0.028	**0.121**	0.996	**4.296**
Darkening	Level 1	BM3D	27.602	0.372	0.900	0.005	0.061	0.996	4.199
		DnCNN	**27.691**	**0.850**	**0.905**	0.005	**0.065**	0.996	**4.207**
	Level 5	BM3D	**30.312**	0.050	**0.116**	0.000	0.000	0.995	0.000
		DnCNN	28.126	**0.078**	0.106	0.000	0.000	0.995	0.000
Dirty Lens	Level 1	BM3D	28.172	0.400	0.956	**0.761**	0.814	0.999	4.957
		DnCNN	**36.317**	**0.943**	**0.957**	0.693	**0.828**	0.999	**4.958**
	Level 5	BM3D	**28.164**	0.502	**0.604**	**0.001**	0.025	0.996	**3.805**
		DnCNN	27.656	**0.671**	0.603	0.000	0.025	0.996	3.801
Exposure	Level 1	BM3D	**28.142**	0.435	0.818	0.000	0.010	0.994	4.319
		DnCNN	27.694	**0.722**	**0.820**	**3.136**	0.010	**0.995**	**4.320**
	Level 5	BM3D	**39.398**	**0.935**	**0.153**	0.000	0.008	0.995	**1.486**
		DnCNN	28.207	0.461	0.148	**5.406**	0.008	0.995	1.480
Gaussian Blur	Level 1	BM3D	28.172	0.435	0.939	**0.708**	**0.766**	0.999	4.950
		DnCNN	**33.938**	**0.865**	**0.941**	0.528	0.618	0.999	4.950
	Level 5	BM3D	28.092	0.556	0.800	**0.153**	**0.261**	0.997	4.660
		DnCNN	**31.052**	**0.653**	**0.802**	0.085	0.206	0.997	4.660
Haze	Level 1	BM3D	28.083	0.403	**0.936**	**0.746**	**0.795**	**0.999**	**4.961**
		DnCNN	**28.952**	**0.812**	0.841	0.123	0.295	0.998	4.484
	Level 5	BM3D	**28.172**	0.525	**0.724**	**0.135**	**0.252**	**0.998**	**4.453**
		DnCNN	27.824	**0.608**	0.409	0.000	0.020	0.995	1.666
Noise	Level 1	BM3D	28.501	0.538	0.651	0.056	0.181	0.997	3.656
		DnCNN	**34.160**	**0.887**	**0.937**	**0.612**	**0.801**	**0.999**	**4.960**
	Level 5	BM3D	28.410	**0.641**	0.471	0.010	0.088	0.996	2.302
		DnCNN	**28.580**	0.355	**0.700**	**0.041**	**0.205**	**0.997**	**4.385**
Rain	Level 1	BM3D	28.010	0.368	**0.722**	**0.258**	**0.417**	**0.998**	**4.857**
		DnCNN	**28.528**	**0.668**	0.652	0.041	0.184	0.997	3.650
	Level 5	BM3D	27.911	0.248	0.398	0.000	0.009	0.995	**4.226**
		DnCNN	**28.203**	**0.606**	**0.469**	**0.007**	**0.090**	**0.996**	2.303
Shadow	Level 1	BM3D	28.257	0.397	**0.961**	**0.834**	**0.864**	**0.999**	**4.962**
		DnCNN	**30.700**	**0.906**	0.721	0.244	0.421	0.998	4.860
	Level 5	BM3D	30.291	0.522	**0.543**	0.000	0.003	0.994	3.829
		DnCNN	**30.319**	**0.607**	0.398	**1.030**	**0.014**	**0.995**	**4.230**
Snow	Level 1	BM3D	28.318	0.517	0.837	0.148	0.286	0.998	4.483
		DnCNN	**35.771**	**0.935**	**0.963**	**0.754**	**0.886**	**0.999**	**4.960**
	Level 5	BM3D	27.727	**0.803**	0.409	**0.001**	0.019	0.995	**1.678**
		DnCNN	**27.824**	0.608	0.409	0.000	**0.020**	0.995	1.666

TABLE IV
IQA RESULTS ON THE CURE-TSD DATASET

			PSNR	SSIM	CW-SSIM	UNIQUE	MS-UNIQUE	CSV	SUMMER
Scene 1	Low Light	BM3D	**46.411**	**0.990**	0.932	0.921	0.943	0.999	4.983
		DnCNN	44.869	0.981	**0.943**	**0.928**	**0.954**	0.999	**4.989**
	Normal Brightness	BM3D	**48.175**	**0.993**	0.952	0.907	0.933	0.999	4.984
		DnCNN	46.860	0.988	**0.969**	**0.958**	**0.970**	0.999	**4.990**
	High Exposure	BM3D	**46.410**	**0.994**	0.964	0.951	0.960	0.999	4.982
		DnCNN	45.930	0.992	**0.984**	**0.973**	**0.976**	0.999	**4.989**
Scene 2	Normal Brightness	BM3D	**45.285**	**0.993**	0.997	0.935	0.939	0.999	4.977
		DnCNN	44.722	0.990	**0.998**	**0.967**	**0.967**	0.999	**4.980**
Scene 3	Low Light	BM3D	**46.866**	**0.992**	0.970	0.907	0.927	0.999	4.981
		DnCNN	45.046	0.982	**0.981**	**0.927**	**0.946**	0.990	**4.989**
Scene 4	Low Light	BM3D	36.500	0.965	0.983	0.579	0.676	0.999	4.959
		DnCNN	**41.560**	**0.970**	**0.992**	**0.871**	**0.897**	0.999	**4.982**
	Normal Brightness	BM3D	**43.067**	**0.985**	**0.991**	**0.835**	**0.867**	0.999	**4.977**
		DnCNN	35.164	0.915	0.979	0.592	0.689	0.999	4.960
Scene 5	Low Light	BM3D	56.976	0.998	0.979	0.955	0.971	0.999	4.988
		DnCNN	**65.526**	**1**	0.**999**	**0.999**	**0.999**	0.999	**4.999**
Scene 6	Low Light	BM3D	**32.661**	**0.804**	0.886	0.429	0.608	0.999	4.876
		DnCNN	32.544	0.796	**0.887**	**0.460**	**0.632**	0.999	**4.881**
Scene 7	Low Light	BM3D	**47.165**	**0.991**	0.984	0.930	0.942	0.999	4.980
		DnCNN	45.650	0.985	**0.990**	**0.958**	**0.964**	0.999	**4.989**
Scene 8	Low Light	BM3D	42.221	0.979	0.966	0.778	0.845	0.999	4.973
		DnCNN	**44.011**	**0.989**	**0.992**	**0.917**	**0.927**	0.999	**4.986**

TABLE V
IQA RESULTS ON SIDD DATASET