February 2023
Master's Degree Thesis

Aggregated multiscale self-supervised denoising

Graduate School of Chosun University
Department of Computer Engineering
Shafkat Khan Siam

Aggregated multiscale self-supervised denoising

멀티스케일 집합을 사용한 자기지도 디노이즈 기법

February 24, 2023

Graduate School of Chosun University

Department of Computer Engineering

Shafkat Khan Siam

Aggregated multiscale self-supervised denoising

Advisor: Prof. Jung, Ho Yub, Ph.D.

A thesis submitted in partial fulfillment of the requirements for a master's degree

October 2022

Graduate School of Chosun University

Department of Computer Engineering

Shafkat Khan Siam

This is to certify that the master's thesis of

Shafkat Khan Siam

has been approved by examining committee for
the thesis requirement for the master's degree.

샤프카트 칸 시암의
석사 논문 승인

위원장 조선대학교 교수 강문수 (인)

위 원 조선대학교 교수 정호엽 (인)

위 원 조선대학교 교수 김판구 (인)

2022년 12월

조선대학교 대학원

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Aggregated multiscale self supervised denoising

Shafkat Khan Siam
Advisor: Prof. Jung, Ho Yub, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

In typical image denoising approaches, both supervised and unsupervised learning methods does not take account of individual image's particular image prior, the noise statistics, or both. The networks learned from external images inherently suffer from a domain gap problem as the image priors and noise statistics can be significantly different from the training and test images. So, it is difficult if the methods primarily requires clean images to train denoising. Furthermore, some images inherently generate significant noise (satellite images of distant galaxies, medical images like MRI images, CT scans, X-Ray images, etc.), and there are no clean images for training. Here the problems dominantly lie with the data delivery system. Our approach takes the noisy images and creates a new version of them with specific pre-processing; by doing so, we make the target pseudo clear image for the deep neural network. We generate multiple versions of these noisy images using interpolation of arrays and train the network to the extent where the network can learn the information on the images without the noises. In practice, the noisy pictures are blurred on three different scales. These blurred versions and the original noisy images are combined together to create

a single set. This set captures all the necessary information from all of the four groups. The network architecture uses the concatenation of the module concept to learn the clear images from a versatile perspective. Then we trained the model using the main noisy set as input and the newly created set as the target to predict a much cleaner image from a regular noisy image. This method creates an output image where the structural integrity image is sustained and the noise component is removed from the image.

한 글 요 약

멀티스케일 집합을 사용한 자기지도 디노이즈 기법

시암 샤프캇 칸
지도교수: 정호엽
컴퓨터공학과
조선대학교 대학원

일반적인 지도 또는 비지도 학습 이미지 노이즈 제거 접근법에서는 개별 이미지의 특정 prior이나 이미지 특정 노이즈 통계를 고려하지 않는다. 노이즈 없는 이미지로 학습된 네트워크에서는 학습 이미지에서의 노이즈 통계가 테스트 이미지와 크게 상이할 수 있기 때문에 본질적인 domain 격차 문제로 어려움을 겪는다. 따라서, 깨끗한 이미지만을 가지고 노이즈 제거를 훈련하는 것에는 다양한 어려움이 따른다. 게다가, 일부 이미지는 내재적으로 상당한 노이즈(먼 은하의 위성, 의학에서의 MRI, CT, X-ray 사진 등)을 발생시키며, 훈련하기 위한 깨끗한 이미지가 존재하지 않는다. 그리고 이런 노이즈는 주로 데이터 전송 시스템에 의해 발생한다. 제안하는 접근법은 노이즈가 많은 이미지를 취하여 특정 사전 처리를 통해 학습에 필요한 새로운 psudo-clean 이미지를 만드는 것이다. 우리는 집합체를 사용하여 이러한 노이즈가 많은 이미지의 여러 psudo-clean버전을 생성하고 네트워크가 이미지 노이즈 정보를 학습할 수 있는 범위까지 네트워크를 학습한다. 구체적으로는 노이즈가 많은 사진들은 세 가지 다른 척도로 smoothing을 하고. 이처럼 smooth 버전과 원래 노이즈가 많은 이미지가 결합하여 하나의 샘플 세트를 만들어낸다. 이 집합은 네 개의 그룹 모두에서 필수적인 정보를 캡처한다. 네트워크 아키텍처는 모듈 개념의 연결을 사용함으로써 다목적 관점에서 선명한 이미지를 학습한다. 그 이후 일반적인

노이즈 이미지에서 훨씬 깨끗한 이미지를 예측하기 위해 메인 노이즈 세트를 입력으로 사용하여 새로 생성된 세트를 대상으로 모델을 교육시켰다. 이 방법은 구조 무결성 이미지가 유지되고 노이즈 요소가 이미지에서 제거되는 이미지를 생성한다.

I. Introduction

The task of denoising an image is a well-studied topic in computer vision. Researchers have developed different types of methods to solve this problem. The significant challenges for image denoising are to make the all part of the image free of noisy components, keep the edges intact, preserve the textures of the various parts of an image, and not introduce new artifacts in the image. It is difficult to maintain balance among these targets in all cases. There are many cases like as images of sequencing tens of thousands of DNA, images of molecules taken with an electron microscope, images of very low-illumination, etc. These are some of the most difficult cases to remove the generated noise from the image. The performance of denoising methods significantly effect the performance of downstream tasks. The computer vision tasks like super-resolution, semantic segmentation, and object detection's result can be drastically changed based on if the denoiser is run on the image before. Novel denoising methods have few goals, such as producing more clear and lossless imagery than previous methods'.

Traditional methods have various processes to remove different types of noise from an image. As noise is commonly presented in a higher frequency spectrum, many previous studies used spatial filters to remove the noisy components presented in the image. Variational denoising methods use image priors, and minimize energy function to obtain the denoised image. Some method worked based on removing the pixel values with the weighted average of neighboring pixel values[1]. There are some methods which developed on the prior knowledge of the signal structures. These methods are domain specific. If a different type of data is provided these methods does not work properly. There is also a

requirement of calibration for these methods. The scale of self-similarity, or the rank of the matrix have various different type of impacts on performance of the method. These non-learning based methods do not have any need for ground truths.

In a typical supervised method for denoising tasks, the main emphasis is on the neural networks. So that, the clean image (y) and the noisy image (X) were fed to the neural network as ground truth and as noisy input, respectively [2]–[12]. A neural network that is trained on noisy/clean pairs learns to predict the clean signal. There is a problem with supervised methods; they perform well in specific type of noise and data they are trained on. But if the noise type or the data type changes, these methods perform poorly. In so many cases, there is no clear ground truth available for noisy images (e.g., MRI images, satellite images, microscopic images, etc.) In these sectors of images, it is difficult to collect the appropriate number of image pairs to create a training dataset. That is why, many recent studies are not using clear images as ground truth to avoid this obstacle[13]–[19]. Most of them use different versions of the noisy image as ground truth, which increases the methods' ability to remove noises from various types of images and, at the same time, keep the information as intact as possible. Some of these methods do not provide any target image and apply window based masking to generate the target image.

For the self-supervised methods, Noise2Noise applies basic statistical reasoning to image reconstruction using deep learning. This study's main goal was to "learn bad images into good images by only looking at bad images" [13]. The Self2Self[15] method of denoising trains the neural network using Bernoulli-sampled instances of the input image. The result of the neural network is estimated by averaging the prediction generated from multiple instances of the

trained model with dropout. Using a self-prediction loss and a blind spot strategy, Noise2Void[16] avoids identity mapping. The Noise2Void and Noise2Self[14] methods use processed noisy images with noisy images for ground truth and input. As blind spots in inputs contains a large area, the predicted pixels' receptive field losses much valuable context. This degrades the performance of denoising. For the motion scenarios or medical imaging, heavy computational burden and artifact generation in denoised image limits the application. Neighbor2Neighbor[18], Blind2Unblind[17], and Recorrupted2Recorrupted[19] use window based masking process to remove the clean image from the training procedure. In the Neighbor2Neighbor method, they have trained with using sub-sampled pair images. As the sub-sampled pairs are used for training, the training leads to over smoothing the predicted image.

In our proposed method for self-supervised denoising, we have developed a technique to tackle previously mentioned problems differently. We have created a training procedure with the basic idea of image pair generation. Our framework takes noisy images as input and generate pseudo-clean images as target for the model to train. We used the interpolation of array technique on noisy images to create multiple pseudo-clean (blurry) versions of the noisy image. We have used scaling to make different sets of pseudo-clean images. In this way, we have removed the clean ground truth from the training procedure. If we assume that noise of different pixels in image are a different layer of values, then downscaling and repopulating the image with neighborhood pixel's value will somewhat remove that noisy layer from the image. This process helps our method to remove any over-fitting problem presented with clean ground truth training procedure. This process of generating target images has done before the training period started. Then we applied different augmentation on these images to increase

the number of training images. We have used a custom U-Net[20] architecture for training these noisy images as input and pseudo-clean images as the target. Our training procedure is similar to Noise2Noise[13]. Here the noisy images are the input and pseudo-clean images work as the ground truth. This process removes the noisy high-frequency components from a noisy image. Our training procedure only takes noisy images as input. This reconstruction network also performs well in extracting the information presented in the image. We have applied our custom loss function, which works by combining the pixel-to-pixel distance between two images, keeping a check on the signal-to-noise ratio, and keeping the structural integrity of the image intact. As our pseudo-clean image generation process introduces the changing values of pixels in the image pairs. The pixel-to-pixel based loss functions such as mean-square-error, peak-signal-to-noise-ratio loss works well to remove the noisy component from the image. The structural-similarity index loss helps to remove blurriness, and maintain the structural integrity in the predicted image as much as possible. Combining all of these in our training procedure helps us generate output images where the structural integrity of images is kept intact, the signal-to-noise ratio is high, and no new artifacts are added.

The main difference of our proposed method to other self-supervised methods is how we have trained the neural network. Some method developed data for training by using different levels of noisy images. Other method have created a mask to generate block of noisy images. A few method have developed sub-sampled images for training procedure. Similar to different noise level, different versions of corrupted noisy images are also as training procedure. The blind spot in the receptive field of the network has also changed by some method to increase the performance of the neural network for denoising. Our method has used a

single set of noisy images and create pseudo-clean images as target by using scaling and interpolation. Most of these previously described methods have used regularization in their loss function to improve the performance of the method. We have developed our own combination of losses to remove the noise from the image, keep the information and quality of the image intact in the output of the model. These are the main differences among our proposed method and other self-supervised methods.

We have evaluated our proposed method in different contexts. We have performed a series of different experiments on both synthetic and real noisy images. We have used AWGN and Poisson noise for synthetic noise experiment. Different experiments on different datasets shows that our method perform very well among self-supervised methods. These results shows the effectiveness of our method in different types of situations.

A. Contributions

In our proposed thesis, we have developed and implemented a new technique of self-supervised image denoising. The contributions of proposed denoising scheme are described as follows:

- We have generated pair of images from a single set of noisy images applying interpolation. Using different scales we have created multiple sets of pairs. In this process we can generate pair of images in an optimal way, which is independent of the neural network.

- We have created a better performing loss function combining commonly used loss functions like mean-square-error, PSNR values, and SSIM values. This combined loss function monitors the quality based parameter like the

signal-to-noise-ratio, structural integrity and removes the noise presented from the image.

- We have developed a training procedure where the combination of pseudo-clean data generation in different scales from noisy data and combining different types of loss functions together to extract the noise-less image have improved performance of self-supervised denoising method regardless of the network it is trained on. Our training procedure takes noisy images as input of the model and pseudo-clean images generated from these noisy images as target. With the help of the loss functions our model learn to remove noise from the image. By this procedure, our model also learn to keep the information presented on the image intact.

B. Thesis Layout

The thesis is organized as follows. In Chapter II, we present previous works done in image denoising. Then in chapter III, we describe in details the problem statement, our proposed solution. Next in Chapter IV, we describe the different variation of experiments done based on our proposed solution, results of the experiments, and ablation study based on the proposed solution. And finally, we have concluded our thesis in Chapter V.

II. Related Work

A. Traditional image denoising methods

Many studies have been done based on filtering and transformation in the early stages of image denoising. Linear and non-linear-based filters are developed to remove noise by calculating the value of each pixel based on the correlation between pixels or image patches in the original image[21]. These filters have successfully removed most of the noise from an image. But the early linear filters tend to over smooth the image texture.

Mean filtering[22] was developed to remove the Gaussian noise from an image. But similar to early linear filter, if the image's noise level was high it can over smooth the image. In order to overcome this over smoothing problem, Wiener filtering[23] was applied. But it can easily blur the sharp edges in the image. Then non-linear median filter[22], [24] and weighted median filter[25] are used to suppress noise from image. Bilateral filter[1] is another filter used for image denoising. It is developed as noise-reducing, non-linear, and edge-preserving smoothing filter. In bilateral filtering the value of a pixel is replaced by the average values of the neighboring pixels in a specific window. Bilateral filter applied on images work as a brute force method as it has to calculate all the values in the window, average it and replace the targeted pixel's value. This process of pixel-wise calculation is not always efficient. If the kernel size of the window rises it also increases the time needs to change the pixel values.

Various image patch-based denoising methods have low time complexities. But when the noise level is high these methods performances drops. The weighted filtering of non-local self-similarity (NSS) prior to the non-local mean creates a point-wise image estimation. Each pixel is the weighted average of pixels

centered at the estimated pixel. Improvement was made by learning from image patches and low-rank property weighted nuclear norm minimization.[26]

The idea for the k- singular value decomposition (k-SVD) algorithm[27], [28] is to learn the dictionary of sparse image patches following a joint optimization problem. This method doesn't follow the correlation among different image patches in cases of high noise. The differences in local information can be seriously distort the output image. Low-rank minimization formats the image as a matrix of patches. Every column of this matrix is considered as a stretched patch vector. By exploiting the low-rank prior of the matrix, this method can remove most of the noise in an image[29], [30]. This type of iterative boosting also has a high computational cost and also time consuming. Independent component analysis (ICA) and Principal component analysis (PCA) are two data-driven methods to remove Gaussian noise. As they use sliding windows, the drawback is highly computational cost.

CBM3D[31] is an extension of the NLM approach. Similar patches are stacked into 3D groups by block matching, then transformed into the wavelet domain. Applying to filter and inversing the image, noise is removed in this procedure. But when there is high noise in the image, CBM3D introduces new artifacts in the image, especially in the flat areas.[26] These patch-based methods filter the noisy images properly based on patches and create a clear image. Similarly, NLM[32] works with patches. There were many variations of this patch-based method developed using CBM3D and NLM as the basis. Such as SADCT[33], SAPCA[34], NLB[35], and INLM[36]. These methods look for self-similar patches in various transformed domains.

There are a few algorithms worked on statistical prior for denoise an image. These algorithms have demonstrated that by using a clean external database,

these methods can remove most of the noise from the image[37]–[40]. These algorithms are also class-specific.

B. Supervised image denoising methods

After the development of neural network, the next step for denoising was to use deep learning methods to optimize the difference between clear and noisy images. These deep learning methods can be categorized as MLP model-based and convolutional neural network (CNN)-based methods. Usually, MLP model-based optimization schemes have time-consuming iterative inferences. But these feed-forward based MLP methods can work well because they has fewer complicated calculation in parameters. On the other hand, CNN-based methods try to learn by mapping the features by optimizing loss functions onto the training set. The use of CNN in image denoising started with[41] a five layer neural network.

In MLP based denoising method, auto-encoders developed by Vincent et al.[42], and Xie et al.[43] are few. In MLP based method the optimization algorithms[44] can generate different type of noise specific architecture. They can interpret the noise of the image better. It also has a downside, which is that it restricts the learned priors and the inference process. So, the model tends to have over-fitting behaviour.

Trainable Nonlinear Reaction-Diffusion (TNRD)[10] is another deep neural network created for image denoising. In this algorithm, several inference steps extend the non-linear diffusion parameters into a set of trainable linear parameters. The main problem with this method is it requires a lot of data to train this method properly. Also, a considerable number of tuning in hyper-parameters needs to be done to train this network.

Most of these neural networks were generated using noisy and clear dataset pairs. Using these datasets, the model learns what a clean counterpart of a noisy image looks like. Some examples of these methods will be[2]–[8]. Among these methods, DnCNN[6] is one of the most prominent ones where the residual learning method is used for denoising an image. It uses a mapping function, combining with batch normalization. The residual learning method and the batch normalization help each other to reach their potential. This integration helps the training procedure. It can also help with the compression of the image or the interpolation error. But a DnCNN model trained in images with Gaussian noise is not suitable for images with Poisson noise or real-life noise. So if the model is trained specifically with image generated from Gaussian noise, it will perform very well in testing scenario where there are only images with Gaussian noises are provided.

For the unknown noise level scenario, the model should be able to adaptively choose between suppressing the noise and protecting the texture of the original image. In order achieve this, fast and flexible denoising convolutional neural network (FFDNet)[8] was developed. FFDNet model is usually trained on down-sampled sub-images. In this process, the training procedure works much faster than DnCNN's training time. IrCNN[9] is another network-based algorithm for denoising developed based on DnCNN. In this method, the residue of a noisy image uses the ground truth as target according to the loss function. But both DnCNN and IrCNN are developed without considering the underlying structure of the noisy image itself. IrCNN's multiple denoising neural networks have been developed based on DnCNN. Jiao et al.[11] has created a neural network with two sub-networks. They are called "FormatingNet" and "DiffResNet." The difference between these two networks is in the loss layers. The "FormatingNet"

uses the variational loss, and the "DiffResNet" uses the l_2 loss as the primary loss function. Combining the work of these two sub-network the method can generate good results.

C. Self-supervised image denoising methods

In recent years, a few neural networks for denoising have been created, particularly targeted for blind denoising. The DnCNN[6], IrCNN[9], TNRD[10], FormatingNet[11], DiffResNet[12] these models can perform well enough in regular synthetic noise. But if the image is a real noisy image, these methods cannot properly remove the noise from the image. Deep neural networks trained with noisy-clean pair also have a problem where it fails to remove real-life noises from images because of the data domain gap[18].

Researchers of computer vision created some methods with self-supervision in mind as they don't need noisy/clean pair for training. Such as Noise2Noise[13], Self2Self[15], Neighbour2Neighbour[18], Noise2Void[16], Blind2Unblind[17], Recorrupted2Recorrupted[19]. These methods are trained and prepared in specific way, so that no clean ground truth is necessary for them. They have used a set of noisy images can be used as a target for the models to train without any clean images provided to the model. These distinctions help the models be trained in a better format and create better results in the blind-noise test. In this way, Noise2Noise[13] has already reached close results to noisy/ground truth paired images. Cha et al.[45] have used the GAN (generative adversarial network) to analyze the structures of noisy images. The problem of Noise2Noise is identity mapping.

This identity mapping problem is removed in the work Noise2Void[16]. This

method only makes the noisy images' predictions by their relative pixel values. A pixel is randomly chosen from a noisy image, then another randomly chosen neighbor pixel's value is applied in the previous position. After that, the loss is calculated with each iteration. Similar work is done in Self2Self[15]. In the self-supervised method of Noise2Self[14], they have used a group of features which work best in condition where noise is independent of the original ground truth. In this way, every features helps the model to learn different kind of scenario which in turns reflects in good performance. In the same way, it is possible to apply these features onto method such as median filters, non-local means etc.

Neighbor2Neighbor[18] is a training technique where the noisy image is sub-sampled and provided to the model. They did not use any target images. They have compared the losses among sub-sampled inputs and the inferred outputs. They have created these sub-sampled images from the noisy image using randomly choosing two pixels from a four pixel window. They have used U-Net[20] architecture as their base denoising model. They have performed custom reconstruction and regularization loss on these sub-sampled input and output. Combining these losses and training technique they have achieved very good performance in removing noise from an image and keeping the original texture intact.

In Blind2Unblind[17] method, a window for masking is applied. This window for masking work in similar way a Neighbor2Neighbor window works. They have also applied re-visible loss with regression based loss similar to Neighbor2Neighbor. In place of sub-sampled noisy image, they have used a global-aware mask mapper. Which will generate mask for the noisy image and increase the number of training images. Similar to Neighbor2Neighbor they did not provide any target image to the neural network. In the place of target image,

they have measured the losses between regular inference of the model and the masked images' inferences of the model. Using a regularization parameter they have controlled the output without providing any target images.

For Recorrupted2Recorrupted[19], the main technique is based on corrupting the noisy images. It has used an unsupervised learning technique for denoising. The main idea behind this method is based on Noise2Noise[13]. For Noise2Noise the training dataset contain two version of the same image with different noise level. In similar way, Recorrupted2Recorrupted corrupts a single noisy datasets with different level of corruption. One of the corrupted set used as noisy input and the other one is used as the target of the neural network. In their training procedure they have used DnCNN as their base denoising model.

In Laine et al.[46], they have developed a blind-spot architecture based on U-Net. In this model, they have created four denoiser network branches, where every one of them has their own respective field fixed to a different direction. So, there image is rotated in four directions and then put through the network. Here a single pixel offset in every branch differentiate each branches. This offset is done by a window technique where the center pixel dictates the offsetting function. Another architecture was developed by them where they rotated the image in four direction and put them as input of a single receptive field-restricted branch. Here the four different rotated output comes as inverse image. In this way Lain et al.[46] has solved the problem with blind spots in the receptive field of the neural network. All of them are aligned together and generated a single denoised image.

III. Methodology

A. Dataset Creation

The basic idea of self-supervised denoising developed from Noise2Noise[13] and Noisier2Noise[47]. In the Noise2Noise method, authors used a clean set of image and applied two different level of noise to it. Then used the lower noise level images as target and the higher noise level images as input of the training network. The main idea behind it similar to "learn to turn bad images into good images by only looking at bad images". In supervised methods, we know the clean target. In the Noisier2Noise method, the authors have used a noisy image set and applied more noise to it. Then use those two noisy variations as input and target of the neural network. According to Fig. 1, we can assume it's in the center if all the distribution of the variants of that specific image. Now if we select another position from this distribution we will achieve noisy variation of that image. In self-supervised technique we don't know the center or the clean variation of the image. But we can generate different variation of the noisy image from this distribution. Then we can use these multiple variants as target of the neural network to train. Convolutional neural network tends to reach towards the average point of the target distribution. In Fig. 1 we can see that the clean distribution lies in the middle where the neural network tends to go. Based on this theory, we have developed our self-supervised method.

The main part of our work is to create the necessary data to work as a target for self-supervised learning techniques. In supervised methods, they have used clean ground truth as their target for deep learning. In our method we do not provide any clean images. The necessary noise for training images is made by additive white gaussian noise (AWGN), which can be distributed evenly on an image.

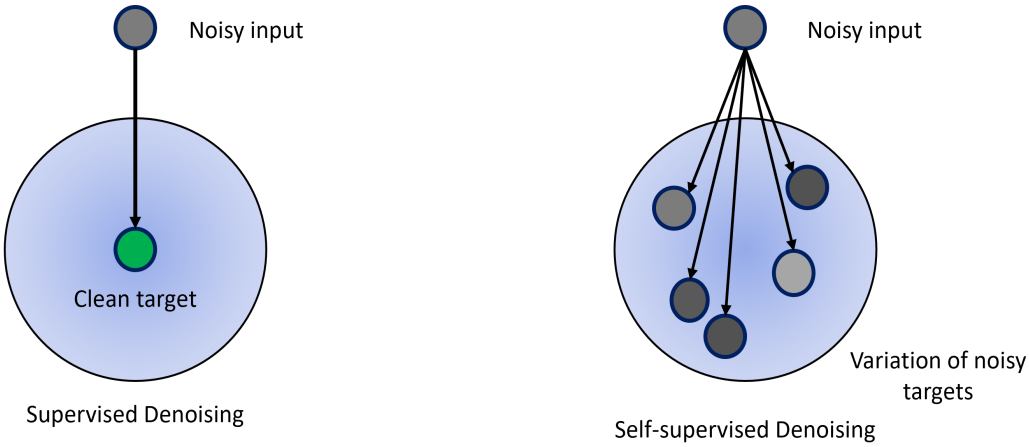

Figure 1: Basic idea for supervised denoising and self-supervised denoising.

We have added AWGN to the clean training data to create the noisy input for the model. For the testing of images with Poisson noise, we have also generated training images using Poisson noise. Even if both of them are synthetic noises, they work well enough to be used as an alternative to real-life noise scenarios. In a real noisy image, we do not know how much actual noise is presented in an image. The noise level of a real-life noisy image is always dependent on the perspective, and vision of the observer.

A noisy image can be described as,

$$X' = X + n. \tag{1}$$

The AWGN we have used is a normal distribution applied to the image at random variation somewhat uniformly. AWGN follows the Gaussian distribution, which is independent of the pixel values of the image. In a similar way, we have applied Poisson noise to create the dataset for Poisson noise removal. We have used these noisy images as the input for the denoising model.

Figure 2: Visual representation of a sample of created dataset containing three sets of same images with AWGN noise level $\sigma = 50$.

The next part is the creation of the pseudo-clean or blurry dataset, which we have used as the target for training. We have applied bicubic interpolation in order to create this target dataset. Using bicubic interpolation, we can estimate the updated pixel values when we upscale the image from the downscaled version. These pixel values can be considered by using Eq. 2, 3, 4, and 5. Using these four equations as the pivotal point, the image we generate can be smoother and blurry. These images contain much less noisy components on the image.

Bicubic interpolation is a process applied to a 2D array to rescale or reshape the array using cubic interpolation or other polynomial techniques. This interpolation process takes neighboring 16 pixels (4×4) into account when it upscales or downscales an array. As there are more values generated from a limited number of values, bicubic interpolation fills up the gap much for efficiently and smoothly. These estimations of values are the process done by any kind of interpolation. Bilinear interpolation uses a single equation to populate these values. But in bicubic interpolation, sets of neighboring pixels' values are considered to generate these unknown values. If we consider the noise of the image as a different layer, when the image is downscaled, most of the noise components get removed. But, the upscaling using bicubic interpolation creates

new values from the neighboring pixels. That is why the output of bicubic interpolation can create a smoother low-frequency-based array or a little blurry image containing less noise than the input image.

If a 4 pixels (2×2) unit square is interpolated using bicubic interpolation the equation for every pixels will be,

$$p(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j. \tag{2}$$

As this 4 pixels will be interpolated to be 16 pixels, there has to be 16 coefficients a_{ij} for every $p(x,y)$ based on the function f, and it's respective derivatives are f_x, f_y, and f_{xy}. The following function provides the $p(x,y)$ for the derivatives as,

$$p_x(x,y) = \sum_{i=1}^{3} \sum_{j=0}^{3} a_{ij} i x^{i-1} y^j. \tag{3}$$

$$p_y(x,y) = \sum_{i=0}^{3} \sum_{j=1}^{3} a_{ij} x^i j y^{j-1}. \tag{4}$$

$$p_{xy}(x,y) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij} i x^{i-1} j y^{j-1}. \tag{5}$$

As an image consists of a matrix, these(Eq. 2, 3, 4, 5) equations will work on height and width of this matrix. When the images are downscaled, every channel's pixel values are affected. In this way, the noisy component is mostly removed. Then the image is upscaled where these equations generate the values necessary. Following this method, the image becomes blurry and somewhat noiseless. The sharpness and blurriness of images can be controlled by the co-efficient a_{ij}. All these co-efficient can be confined in a single vector α. The functions for achieving these pixel values can be considered into a vector x. The relation between these

functions and coefficients can be comprised in a matrix $A\alpha = x$. This matrix can easily extract the values of α and, in terms, can clearly estimate the pixel values of the changed shape of the array. Applying this method, we have created pseudo-clean images into three different sets. These images are then randomized to create a more generalized and varied dataset.

We can see a clear image in (Fig. 2). Then AWGN of level 50 was added onto it. We have scaled the image down with a 1.2 ratio and upscaled it back to the previous size again. In the same way, we have downscaled the same noisy image to 1.5, and 1.8 scales and upscaled it to the original shape. For scaling coefficient, we have used different values to make these pseudo-clean images more diversified. From visual representation, we can see the scaled versions are much less noisy and more blurry than the real noisy image. We have used this way to create our dataset. For our training dataset, we have added the noise in random values in levels 50 to 70 for AWGN and 40 to 60 for Poisson noise.

B. Training Procedure

As described in the dataset creation section, we can generate different versions of noisy images. Using bicubic interpolation, we can generate the target images. First, we have downscaled the noisy images. When the images are downscaled, it changes the pixel values as only a single values of the window is considered. As the noise layer are mixed with pixel values, many of the noise components get lost by downscaling. Then applying bicubic interpolation on these downscaled images, we upscaled these images. For returning to the previous size there are gaps that needed to be full-filled. In times of upscaling, we have generated the remaining gap values. We have used the pixel-wise equation of bicubic

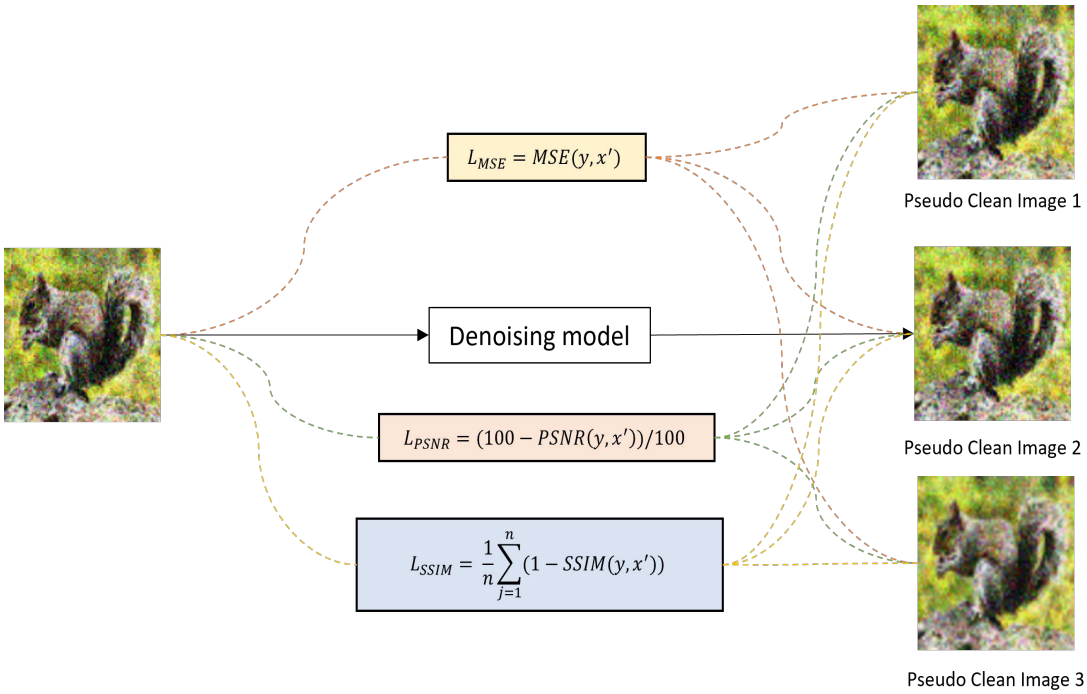

Figure 3: Overview of proposed multiscale target denoising framework.

interpolation to generate these pixel values. These values are generated using neighborhood pixels' values. Considering the neighboring pixels' these images become blurry and less noisy. The blurriness of the image can be controlled by managing the a_{ij} coefficient of new pixel value generation.

We have generated three sets of pseudo-clean images for training using a single set of noisy images. If the noisy image is y then we can call the pseudo-clean image of this noisy image as $f_{scaled}(y)$. As the y has gone through the interpolation, most of the high-frequency noise components are removed from these images. The output images are also become a little blurry. So these images contains more information that was presented in the clean version. In this way, the generated images can be used as the target part of the noisy/pseudo-clean

pair to train the neural network. In the Fig. 3, we have presented our whole training procedure in a simple format. There we can see the denoising model is an interchangeable part of a self-supervised training procedure. Here we calculate the losses for a single noisy image with three different target images.

Our target is to extract structural and noiseless information from these images. After entering the model, the output of the model can be described as, $f_m(y)$. So, if we write this process in equation,

$$\lambda_1 L_{MSE}(f_m(y), f_{scaled}(y)) = 0. \tag{6}$$

$$\lambda_2 L_{PSNR}(f_m(y), f_{scaled}(y)) = 0. \tag{7}$$

$$\lambda_3 L_{SSIM}(f_m(y), f_{scaled}(y)) = 0. \tag{8}$$

Minimizing these three losses (Eq. 6, 7, and 8) in training can achieve better performance than regular self-supervised methods.

Here we use the previously generated data in random sets of noisy and pseudo-clean images as the y and $f_1(y)$, $f_2(y)$, and $f_3(y)$, respectively. The model here learns the difference between the pseudo-clean version and the noisy version of the image with respect to the typical mean-square-error (MSE), the structural similarity index (SSIM), and peak signal-to-noise ratio (PSNR) as a custom loss function. In this way, the neural network learns the best way to remove the noise from a regular image and generate a clearer version of that image without any noise, at the same time not removing its structural integrity.

We efficiently run the noisy image to the target clear image using a U-Net as our denoising model. The model trains itself to achieve the best possible outcome

for any input image. As we are using the pseudo-clean target in different scales for training a denoising model, our total procedure can be described as a self-supervised training procedure.

C. Loss Function

A regular end-to-end neural network has blind spots in their receptive fields. Because of this type of blind spots, the inference of a regular end-to-end type model can cause different artifacts on images. This can also generate pixelation effect on the output images. If the loss function is not applied properly, there can be issue how the model is learning to remove the noise. As there is no clean ground truth present in the model, the tendency of over-fitting is low. Here we describe how our combination of losses can help in self-supervised denoising. Mean-square-error loss is very common to use in image generation type model. But typical mean-square-error loss (L_{MSE}) thinks every pixel is independent. That means the changes of values in the neighboring pixel does not effect the distance of selected pixels. That is why applying only meas-square-error loss (L_{MSE}) can generate artifacts on the image. To remove that effect, we are applying a customized loss combining PSNR, SSIM, and MSE together in different ratio.

$$L_{total} = \lambda_1 L_{MSE} + \lambda_2 L_{PSNR} + \lambda_3 L_{SSIM}. \tag{9}$$

Here the L_{MSE} loss indicates the pixel-wise euclidean distances. It is a very effective loss to make the inference image to get more closer to the target image. In this loss the effect of neighboring pixel is not considered. As the L_{MSE} considers every pixel independent of each others. This loss helps to make the output identical towards the target images. In the Eq. 10 we have provided how

the pixel-wise distance is calculated.

$$L_{MSE} = \frac{1}{n}\sum_{j=1}^{n}(y_g - y_p)^2. \tag{10}$$

L_{PSNR} is the loss calculated to reduce the noise as it suggests the peak-signal-to-noise-ratio. PSNR values always depends on the presence of noise. As our prepared target image has visually lower noise than input, keeping the PSNR values in check helps the model to learn how to remove the noise from any image. The calculation of PSNR values are also dependent on the pixels similar to L_{MSE}. But the reason behind using L_{PSNR} separately is that this calculation helps the model to reduce the noise component presented in the output. As we know PSNR values are ratios, so we have formulated the Eq. 11 to integrate the PSNR value as loss into the model.

$$L_{PSNR} = (100 - PSNR(y_g, y_p))/100. \tag{11}$$

Also, the L_{SSIM} is the loss where the inter-pixel dependency is taken into account. The SSIM calculation indicates the structural integrity of a noisy image with a clear image. In SSIM calculation, the luminance, contrast, and structure these three quality of an image is considered. Using the values presented in the pixel SSIM values calculate these quality based parameters. Considering these parts of the image equations the neural network is encouraged to keep the correct luminance, contrast, and structure intact in the predicted output. As the inference image gets more and more of it's noise removed the structural integrity also increase. We have formulated the SSIM loss (Eq. 12) to keep the SSIM values in check. It helps to create visually pleasing images from end-to-end neural networks.

$$L_{SSIM} = \frac{1}{n}\sum_{j=1}^{n} 1 - SSIM(y_g, y_p). \tag{12}$$

The structural-similarity-index loss (Eq. 12) is center-heavy because of the typical structure of a convolutional neural network. So, there is a possibility of being biased toward the center pixel of the image. As SSIM loss is image quality based loss, it helps the neural network to generate more structurally sound and higher quality based images. Each of these three losses perform different operations. In our experiments we have found that each one of them performed their role in removing noise from images and generate high quality structurally sound clean image. That is why we have needed the most benefit we can get from all of these losses. We have created a custom loss by combining all three losses to perform better together.

For the customized loss function combination we empirically tuned the parameters λ_1, λ_2, and λ_3 respectively to 0.25, 0.85, and 0.3. These parameters are obtained by trial and error of various combinations. These values injects the influences of each loss functions successfully towards training the model so that the neural network can learn properly to remove noises from a noisy image. Together with our created datasets for training and loss functions our self-supervised training procedure can generate good results.

IV. Experiments

This section demonstrates the proposed method's performance compared to seven different studies. Here, five of them are self-supervised denoising methods, CBM3D is a prior-based method, and DnCNN is the supervised denoising method. We have compared AWGN and Poisson noise as synthetic noise. In the following, we present the experimental settings and then show the qualitative and quantitative evaluation of six widely used datasets in synthetic and real noisy scenario.

A. Experimental Setup

In our training process, we use a modified U-Net architecture. For optimizer we have used GCRMSprop custom optimizer based on Adam with a learning rate of 0.0001. We have implemented early stopping measurement to avoid over-fitting problem. We monitor the validation loss with the previous best loss and update if any comes better. We use the batch size of 8, where all the images are normalized between 0 and 1. We have selected 5000 images from ILSVRC2012 validation dataset. We have applied our described dataset creation on these images so for one input images there are now three different target images. For the hyper-parameters of the losses we have tuned these empirically at $\lambda_1 = 0.25$, $\lambda_2 = 0.85$, and $\lambda_3 = 0.3$. We have also applied the usual data augmentation procedure to all the training data. The images selected for training has a balance of different classes. There are different types of indoor and outdoor images of different subjects. They are high-resolution images so that the patches can be much more apparent than any typical image. We have trained our model on a server using Python 3.8.10, TensorFlow 2.8.0, and 3 NVIDIA 3090Ti GPU.

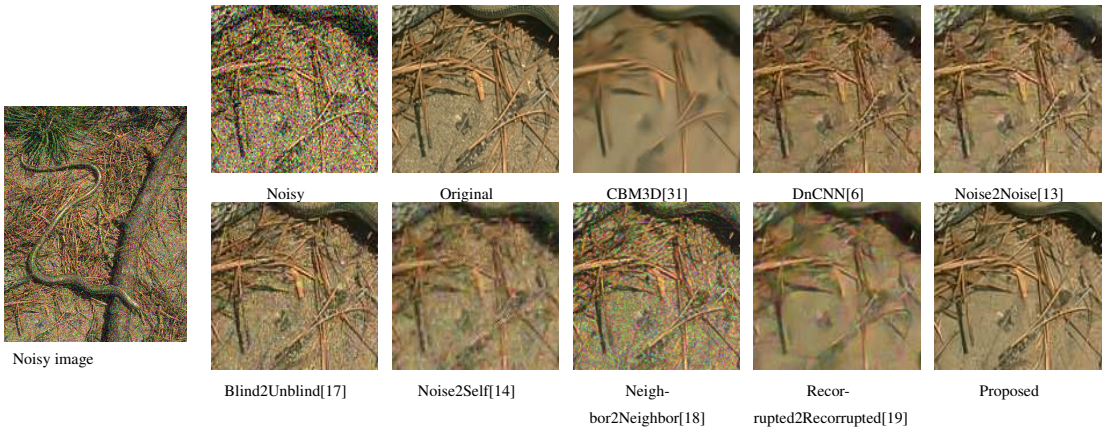

Figure 4: Visual quality comparison for " Snake " from the BSD300 dataset with AWGN noise level $\sigma = 50$.

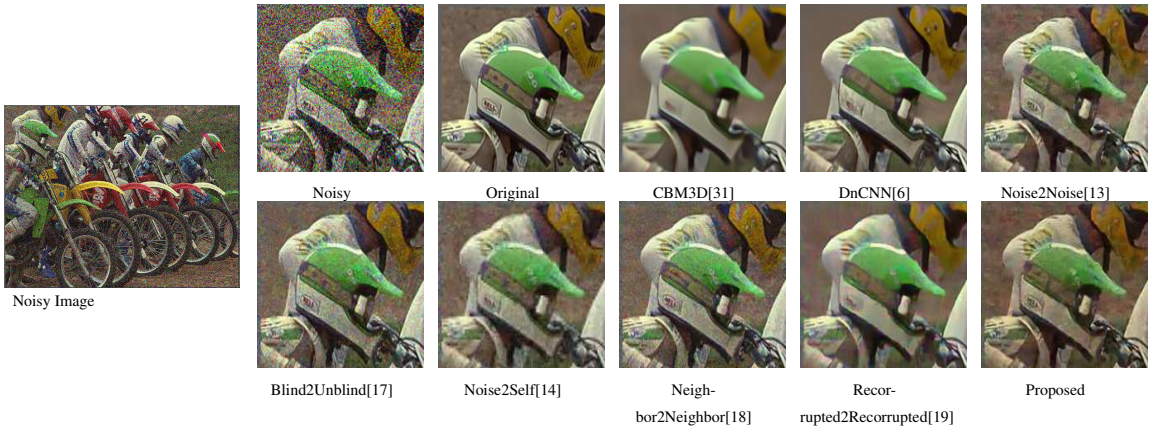

Figure 5: Visual quality comparison for " Bike " from the Kodak dataset with AWGN noise level $\sigma = 50$.

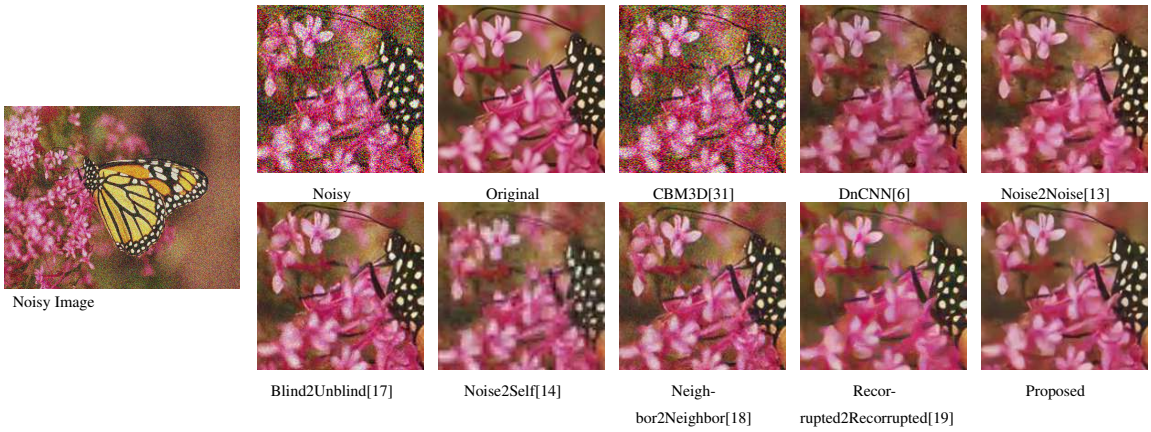

Figure 6: Visual quality comparison for " Monarch " from the Set14 dataset with AWGN noise level $\sigma = 50$.

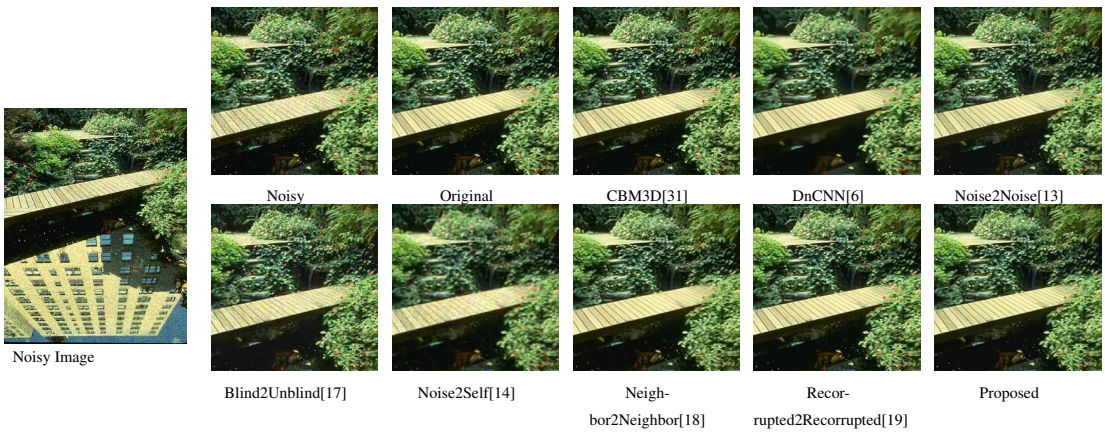

Figure 7: Visual quality comparison for " Building " from the BSD300 dataset with Poisson noise level $\lambda = 30$.

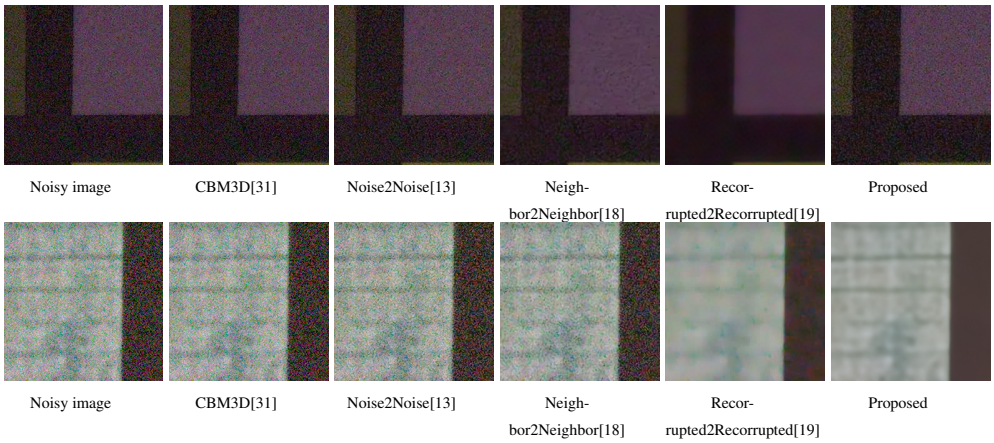

Figure 8: Visual quality comparison for SIDD dataset.

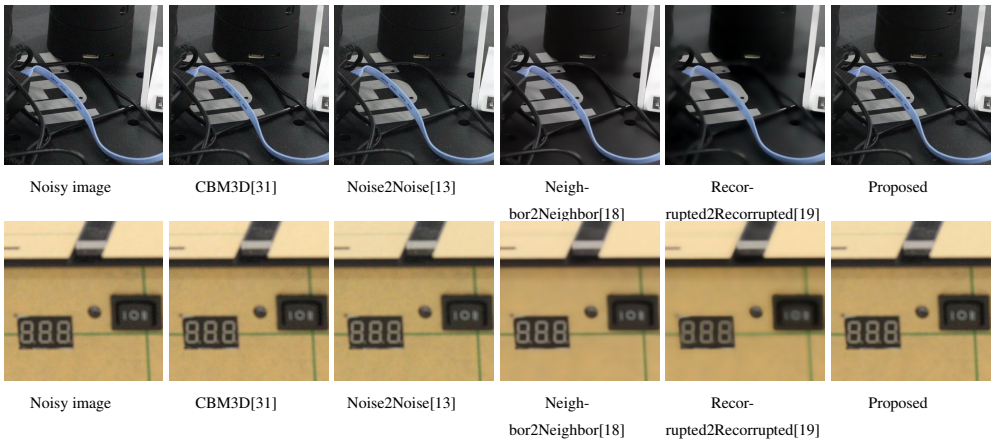

Figure 9: Visual quality comparison for the PolyU dataset.

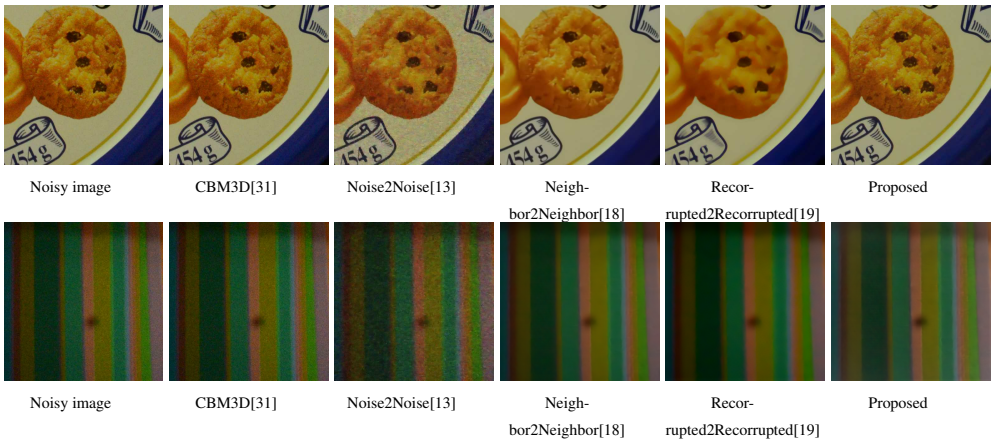

Figure 10: Visual quality comparison for CC dataset.

B. Testing Datasets

After training we have applied our method into rigorous testing. For synthetic noise we have used AWGN and Poisson noise. Using these two noises we have performed our evaluation. For synthetic noise dataset we have used BSD300[48], Kodak[49], and Set14[50]. The BSD300 dataset contain 100 images, Kodak dataset contain 24 images, and Set14 dataset contain 13 images.

For real life noise, we have used three different datasets. SIDD[51] is a dataset commonly used as a real-life noisy image dataset. It contains 40 large images for testing where every images are broken into 31 patches, containing total 1280 image patches. All of these patches are of 256×256 shape. They have provided a mat file containing all these data. PolyU[52] is another real-life noisy dataset. It contain 100 noisy images for testing. CC dataset provides 15 real-life noisy images. We have performed test on these methods and provided the results here.

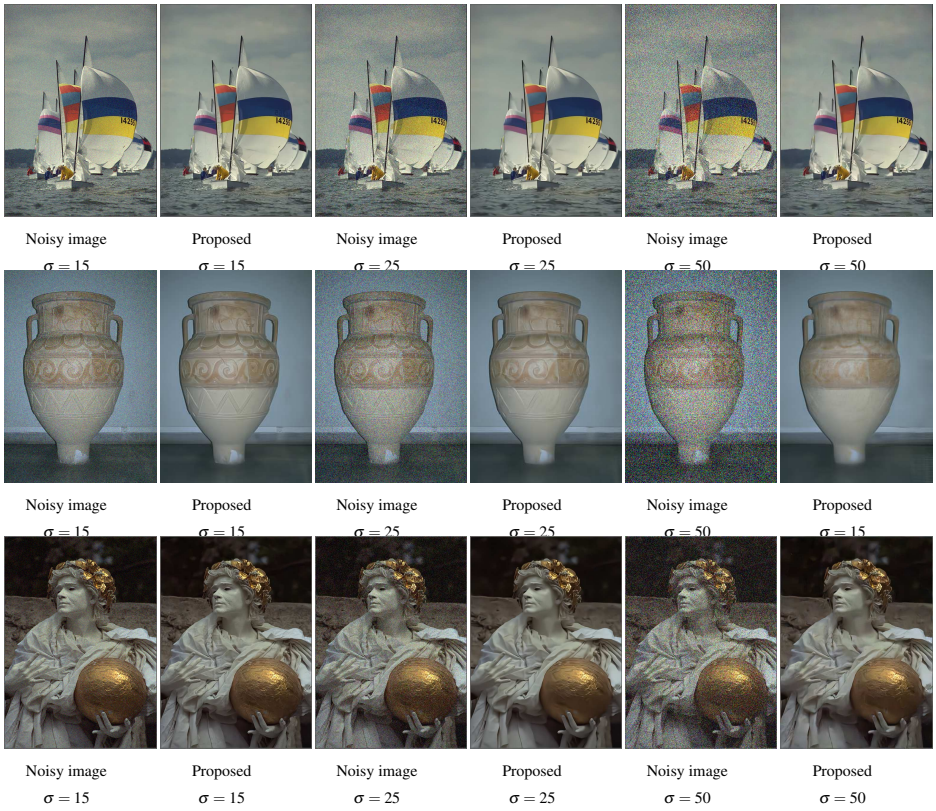

Figure 11: Visual quality comparison for BSD300, and Kodak datasets for AWGN noise level $\sigma = 15$, 25, and 50.

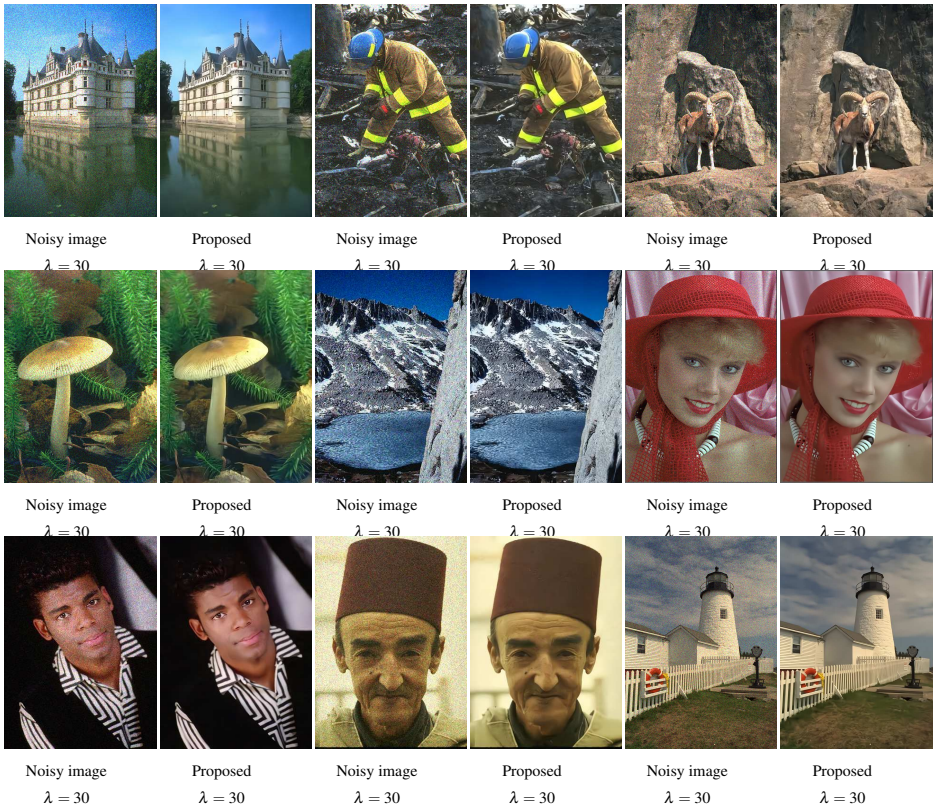

Figure 12: Visual quality comparison for BSD300, and Kodak datasets for Poisson noise level $\lambda = 30$.

C. Visual comparison analysis

For comparisons in synthetic noises, we have performed testing on following methods: CBM3D[31], DnCNN[6], Noise2Noise[13], Noise2Self[14], Blind2Unblind[17], Neighbor2Neighbor[18], and Recorrupted2Recorrupted[19]. In Fig. 4, 5, 6, and 7 we have selected a patch from the main image, and its visual representation has been presented in all of these previously mentioned methods. There is also a noisy patch and a original ground truth patch provided for visual comparison. The visual representation of multiple methods shows the strengths and weakness of these methods in separating the noises from the image. Here Fig. 4, 5, and 6 contains images with AWGN, and Fig. 7 contains images with Poisson noise.

Fig. 4 demonstrate the output generation by different methods for a image of BSD300 dataset. This image contain a snake and straws laid on the ground. The patch we are discussing is a patch containing the straws. In the CBM3D method's output is over-smoothed. The straws presented here are not clear. The patch of DnCNN contains the straws, but the ground is smoothed in some places. The output of Noise2Noise method contains some noise. Here the straws and the dirt on the ground is more present than DnCNN. In Blind2Unblind's output patch we can see the noise is more present. The patch of Noise2Self is blurry and there is artifacts of pixelation. Here we cannot properly see the straws and ground. Neighbor2Neighbor has removed the noise partially, so the straws are visible. Recorrupted2Recorrupted method's generated patch has successfully removed the noise from the image. But there are some blurriness present on the ground part of the image. Here our proposed method's output has removed most of the noise from the noisy image. But it also removed some of the straws and dirt those

are presented in the original image patch.

We have presented a image of bikers in Fig. 5. Here in the visual comparison we can see that CMB3D, DnCNN have removed the noise from the biker's helmet but these method has over-smoothed the patch. The output of Noise2Noise, Blind2Unblind, Neighbor2Neighbor, and Recorrupted2Recorrupted has partially removed noise from the helmet. But in Recorrupted2Recorrupted the output patch has become blurry. The output patch of the helmet in Noise2Self contains pixelation effect. Our proposed method's output has removed the noise partially from the image but there are some artifacts are present in the background of the helmet. In Fig. 6 a image of a monarch butterfly is presented as noisy image. The represented outputs show a part containing flowers and head of the butterfly. Here DnCNN, Noise2Noise, Blind2Unblind have removed most of the noise from the patch. The output of our proposed method is almost noise-free. Also the shapes of the flowers are clearly visible.

In these three visual comparisons we have provided results for AWGN applied on image. Here, the structure of the images' subjects and the contrast changes in the some of the methods in some cases. In our proposed method's pictures, the noise is reduced, and the structure of subjects, color, and contrasts are more similar to the original ground truth patch.

We have similarly added a sample result for poisson as synthetic noise on clean image. In Fig. 7 we have provided result for a single image in BSD300 dataset with Poisson noise which contain a shadow of a building on a pond. Here we can observe that most of the methods have successfully remove most of the noise from the presented image but Noise2Self and CBM3D have made the image smoother. We can observe the smoothness in the leaves presented here. The patch for Noise2Self contains some type of pixelation effect presented on the bridge and

the leaves. The output of Blind2Unblind has introduced some pixelation artifacts. For the method of Neighbor2Neighbor there are some blurriness on the image patch. The output of Recorrupted2Recorrupted method has generated a almost noiseless and structurally sound image patch, which is very close to the original patch of bridges and leaves of the trees. Our proposed output has removed most of the noise and kept the structure of the image more accurate to the original ground truth.

The performance of our model with real-life noisy images are done on SIDD, PolyU, CC test datasets. These visual representations are provided in Fig. 8, 9, and 10 respectively. In these figures we have provided two samples from each datasets. From these images we can see that our model removed the presented noise in the image without admonishing the image in any way. In the Fig. 10, we can see that even the texture is also presented in our image without the noise. Where as, other methods either changed the color or failed to successfully remove the noise from these presented images.

In Fig. 8 top row, it contain a patch of SIDD images. We can observe that there are some noise still available in our proposed output. We can also see that CBM3D, Noise2Noise methods also contain some noise in the output. The output of Recorrupted2Recorrupted has removed the noise from the image but also introduce the pixelation effect which distorts the structures of the image. For the second row, it is another patch provided in the SIDD images, there is noise present in the output of CBM3D. The output of Noise2Noise also contains the noise from the original input. The neighbor2Neighbor method's output has removed some noise. The output generated by Recorrupted2Recorrupted has successfully removed the noise from the image. But the output is also became blurry. The output of our proposed method in the second row has removed most the noise but

introduced a little blurriness, so the patch is not totally clear.

In Fig. 9 first row contains a image of wires on a table. CBM3D and Noise2Noise could not remove the noise properly. Neighbor2Neighbor and Recorrupted2Recorrupted methods have removed the noise completely although there are some blurring present in the image. In our proposed method's output we can observe that the noise is removed from the image and the structure of the wires in the image is intact. The second row contains a reading of a meter. Here the CBM3D and Noise2Noise method has removed the noise partially. The output of Neighbor2Neighbor has removed most of the noise from the switch of the meter in the image, but there is a blurriness to the image. Similarly in the Recorrupted2Recorrupted method's output contain almost no noise, but blurriness is present in the image of the meter. In our proposed method's output, there is still some noise left on the background, and switch parts but there was no blurriness introduced here in this image.

For the CC dataset in Fig. 10 first row, we can observe that the Recorrupted2Recorrupted and Neighbor2Neighbor methods have removed the noise from the image. Also these two methods have partially removed some information from the "Biscuit" in this image. The Noise2Noise method could not remove all the information presented in the image. Here, our proposed method have removed most of the noise also kept the information presented on the "Biscuit" intact. In the second row of the images in Fig. 10 there is a image of a cloth. In this row, the Noise2Noise method's output still contains some noise. The output of Neighbor2Neighbor method has some blurriness introduced on the cloth. Recorrupted2Recorrupted method's output contains almost no visible noise. But in the image there are more blurriness present than Neighbor2Neighbor's method. In our proposed output the cloth can be seen

almost noise-free and without new artifacts.

Here in Fig. 11 we have added a few visual demonstration of denoised image for synthetic noise. We have provided results for Gaussian noise, $\sigma = 15$, 25, and 50. Here we can see visually how our method performed in different noisy images. In the first row, there is a image of a sailboat. On this image different levels of AWGN noise has been applied. As we can see in the even column of first row the image of the boat gets a little blurry as the noise increase. Because the input to the model gets higher noise the model tends to struggle a little. That's why the denoised images gets blurry. The second row contains a image of a vase, similar to the first row, as the noise gets higher the image gets a little blurry. The design presented on the vase which we can see clearly in the noise level $\sigma = 15$ is not properly visible in the noise level $\sigma = 50$. The statue of the third row present details in the noise level $\sigma = 15$. In the output of noise level $\sigma = 25$, the image contains all the information presented. In our output most of the noises were removed. But in the output of noise level $\sigma = 50$, the image contain blurriness.

In the Fig. 12, we have provided various samples for images with Poisson noise. Here we have shown images from BSD300 and Kodak datasets. In the first image, there is a building near a pond. The reflection of the building is also present on the water. In our proposed method there are some noise present in the reflection of the image. The second image is a image of a fireman. In our generated output there are no noise present. But there are some blurriness at the debris. In our generated goat image there are some noise still visible. For the mushroom image, our generated output creates some blurriness in the background. The noise presented in the mountain image was tough for the model. As we can see there are still some noise present in the output. The output for the model image is almost noise-less. But there are some blurriness in the hair area

of the model. In the human image, our proposed method has generated a visibly noise-less output. The second human image contains some pixelation effect in the dress of the human. The last image contains a view of the sky with a watchtower. Our generated output is mostly noise-free. But the cloud in the sky contains some blurriness and some pixelation effect as artifacts.

Table 1: Quantitative comparison, in PSNR(dB)/SSIM, of different methods for AWGN removal on BSD300, Kodak24, and Set14. The compared methods are categorized according to the type of training samples.

Noise Type	Method	BSD300	Kodak24	Set14
Gaussian Noise, $\sigma \in [5,50]$	CBM3D[31]	30.56/0.847	32.02/0.860	30.94/0.849
	DnCNN[6]	**31.07**/0.866	**32.51**/0.875	**31.41**/0.863
	Noise2Noise[13]	28.72/0.815	29.67/0.899	31.37/0.868
	Blind2Unblind[17]	30.86/0.861	32.34/0.872	31.14/0.857
	Noise2Self[14]	29.79/0.832	30.56/0.809	29.92/0.822
	Neighbor2Neighbor[18]	30.73/0.861	32.10/0.870	31.05/0.858
	Recorrupted2Recorrupted[19]	28.25/0.808	29.98/**0.906**	31.32/0.865
	Proposed	30.14/**0.887**	30.19/0.830	29.67/**0.876**
Gaussian Noise, $\sigma = 50$	CBM3D[31]	24.48/0.568	27.02/0.682	26.32/0.813
	DnCNN[6]	25.92/0.718	28.56/0.763	26.08/**0.825**
	Noise2Noise[13]	25.77/0.700	25.85/0.730	**30.21**/0.763
	Blind2Unblind[17]	25.61/0.765	26.59/0.698	25.98/0.723
	Noise2Self[14]	**28.12**/0.792	**29.24/0.903**	27.96/0.759
	Neighbor2Neighbor[18]	26.13/0.709	27.12/0.849	26.03/0.813
	Recorrupted2Recorrupted[19]	26.01/0.798	26.65/0.801	26.12/0.749
	Proposed	26.59/**0.821**	26.22/0.752	28.45/0.804
Poisson Noise, $\lambda \in [5,50]$	CBM3D[31]	27.48/0.698	28.56/0.767	28.65/0.813
	DnCNN[6]	29.77/0.851	31.19/**0.861**	30.02/0.842
	Noise2Noise[13]	29.65/0.844	29.78/0.848	29.86/0.798
	Blind2Unblind[17]	29.98/0.868	29.89/0.858	29.83/0.857
	Noise2Self[14]	28.93/0.823	28.08/0.808	28.62/0.835
	Neighbor2Neighbor[18]	**30.86**/0.855	29.54/0.843	29.79/0.838
	Recorrupted2Recorrupted[19]	29.14/0.732	29.14/0.732	28.77/0.765
	Proposed	29.73/**0.877**	**31.58**/0.849	**30.93/0.895**

Table 2: Real-image denoising results of several existing methods on SIDD, PolyU, and CC dataset.

Dataset	Metrics	CBM3D[31]	Noise2Noise[13]	Neighbor2Neighbor[18]	Recorrupted2Recorrupted[19]	Proposed
SIDD [51]	PSNR	25.65	27.68	34.75	**34.78**	33.73
	SSIM	0.685	0.668	0.853	**0.898**	0.844
CC[53]	PSNR	25.19	32.77	36.43	**37.78**	36.76
	SSIM	0.658	0.7381	0.9528	**0.951**	0.936
PolyU [52]	PSNR	27.40	36.59	37.46	**38.47**	35.82
	SSIM	0.753	0.725	0.958	**0.965**	0.945

D. Performance comparison analysis

According to the table 1, in Gaussian noise, $\sigma \in [5,50]$ category, proposed method's performance is lower than DnCNN[6], CBM3D[31], Neighbor2Neighbor[18], and Blind2Unblind[17] in PSNR scores but the SSIM scores for BSD300 and Set14 is higher. Even if our PSNR scores were lower than the supervised method, they were still higher compared to Recorrupted2Recorrupted[19], Noise2Self[14], and Noise2Noise[13] these self-supervised methods. For Gaussian noise, $\sigma = 50$ category, we can observe that our method perform well in Set14 dataset. Also the SSIM scores for all three dataset was high compared to other methods. In this section, Noise2Noise performed highest score in set14 dataset. Noise2Self performed best in BSD300, and Kodak datasets. Our proposed method performed better than Blind2Unblind, Neighbor2Neighbor, and Recorrupted2Recorrupted in this section. In the Poisson noise, $\lambda \in [5,50]$ category of table 1, we can observe that our method performed highest PSNR scores in Kodak and Set14 datasets. Also the SSIM scores are highest in BSD300 and Set14 datasets. If we compare our results with Noise2Noise's results then for Gaussian noise, $\sigma \in [5,50]$ Noise2Noise achieve

PSNR and SSIM scores as 29.92 and 0.861, and our proposed method achieves 30.00 and 0.864. For Gaussian noise, $\sigma = 50$ case Noise2Noise's SSIM score is 0.731, and ours is 0.792. In Poisson noise, $\lambda \in [5, 50]$ category, our proposed method's PSNR score is 30.75 and Noise2Noise's PSNR score is 29.76. Our proposed method's SSIM score for this case is 0.874 and Noise2Noise's SSIM score is 0.830. Overall our proposed method's PSNR and SSIM scores are 29.28 and 0.843, and Noise2Noise method's PSNR and SSIM scores are 28.99 and 0.807. Here we can observe that our method based on the idea of Noise2Noise method has outperformed Noise2Noise in different cases and in overall average. We have identified the highest PSNR and SSIM scores with bold letters and the second highest scores with underlining in table 1, and 2.

In the real-life noisy comparison of table 2, we can see that the highest scores of PSNR, and SSIM are hold by Recorrupted2Recorrupted. The second highest scores for SIDD and PolyU datasets are hold by Neighbor2Neighbor method. Our proposed method holds the third highest scores for PSNR and SSIM. As we can see in the real-life noisy scenario, our method is not the best performing method. But the PSNR and SSIM scores of our method is high and the visual representation shows that our method is capable of removing real-life noise.

E. Ablation Study

Figure 13: Sample results for applying specific combination of losses.

Here in Fig. 13, we have provided demonstration of removing one loss each time. We can observe that, when MSE loss is not used there are many noise left in the image. As MSE calculate the pixel to pixel euclidean distance between images, it helps to identify the noisy elements and remove it in the training process. For removing the PSNR loss, we can observe that MSE and SSIM loss cannot properly remove all the noise from the image. As we know, the PSNR value keeps check of the amount of noise present on the image. Because of that reason, partial values of MSE loss and SSIM loss cannot effectively remove the noise from the image. The third loss SSIM mostly effect on the structure and the color contrast of the image. That's why MSE, and PSNR loss can remove most of the noise from the image. Here we have shown that demonstration with three different image of horse, house, and lake.

Table 3: Removing different losses and observing the results for BSD300 dataset.

Noise type	Metrics	MSE Loss Removed	PSNR Loss Removed	SSIM Loss Removed	Proposed
$\sigma \in [5,50]$	PSNR	26.39	25.47	<u>28.65</u>	**30.14**
	SSIM	<u>0.731</u>	0.692	0.583	**0.887**
$\sigma = 50$	PSNR	24.64	23.86	<u>25.17</u>	**26.59**
	SSIM	<u>0.715</u>	0.688	0.511	**0.821**
$\lambda \in [5,50]$	PSNR	25.65	26.33	<u>28.93</u>	**29.73**
	SSIM	0.711	<u>0.724</u>	0.572	**0.877**

We have introduced the table 3, to show the results of applying various combination of our basic losses. We have removed one loss at a time and trained the model. Then generated results using those weights.

Here we can observe that, in case of training without MSE loss, the PSNR scores dropped significantly for AWGN noise level $[5,50]$, AWGN noise level 50, and poisson noise level $[5,50]$ cases. As MSE calculate the distance between pixels, the neural network failed to remove most of the noise components from the image properly. But as SSIM loss and PSNR loss works together it can generate the structure well enough to achieve good SSIM scores.

For the case of training without PSNR loss, according to table 3, the PSNR scores drops for AWGN noise level $[5,50]$, AWGN noise level 50, and poisson noise level $[5,50]$. PSNR loss keeps the PSNR values in check that is why the PSNR scores of these images have dropped. But as SSIM loss and MSE loss works in the training, they keeps the SSIM scores relatively high.

When we have removed the SSIM loss, the scores of PSNR dropped slightly for AWGN noise level $[5,50]$, AWGN noise level 50, and poisson noise level $[5,50]$. But here the SSIM scores have dropped much more than any other cases.

The reason behind this is SSIM loss keeps the main information of the image intact. So the noise removal is not it's main priority. The loss of structural integrity in an image can create new artifacts, and sometimes distort the image. SSIM loss increases the structural integrity and helps to reconstruct the image as close to the original ground truth.

V. Conclusion

In this paper, our main target was to develop a system where the denoising can be done by self-supervised method. In our method we did not provided any clean image to the neural network. In order to achieve our target, we have created multiple set of pseudo-clean images in different scales. These images are created from various levels of noisy images. We have used these sets of images as target images. By using these images, we can train the network to extract the embedded features of the images while discarding the noise. Our combination of losses keeps check of the signal-to-noise ratio, structural integrity of the image, and pixel-wise differences. Applying all of these together the model learns to remove the noise and keep the information in the image intact without any necessity for clean ground truth. This training procedure keeps the color and the structure of the images intact. To enhance the performance of our self-supervised denoising, we have used a modified architecture built on U-Net. Finally, we compared our results in real-life noisy and synthetically noisy images with different denoising methods. Visible results with metric-based results show our method's performance in different scenarios. The results of our method might not be best in every cases, but it perform very well in multiple cases in different types of images. Also, the visual results confirm that our method is capable of removing noise from images in different types of cases.

Publications

A. Journals

1. M. A. N. I. Fahim, N. Saqib, S. K. Siam *et al.*, "Denoising single images by feature ensemble revisited," *Sensors*, **jourvol** 22, **number** 18, 2022, ISSN: 1424-8220. DOI: 10.3390/s22187080. [Online]. Available: https://www.mdpi.com/1424-8220/22/18/7080.

2. M. A. N. I. Fahim, N. Saqib, S. K. Siam *et al.*, "Rethinking gradient weight's influence over saliency map estimation," *Sensors*, **jourvol** 22, **number** 17, 2022, ISSN: 1424-8220. DOI: 10.3390/s22176516. [Online]. Available: https://www.mdpi.com/1424-8220/22/17/6516.

References

[1] C. Tomasi **and** R. Manduchi, "Bilateral filtering for gray and color images," **in***Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* 1998, **pages** 839–846. DOI: 10.1109/ICCV.1998.710815.

[2] J. Chen, J. Chen, H. Chao **and** M. Yang, "Image blind denoising with generative adversarial network based noise modeling," **in***2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2018, **pages** 3155–3164. DOI: 10.1109/CVPR.2018.00333.

[3] S. Guo, Z. Yan, K. Zhang, W. Zuo **and** L. Zhang, "Toward convolutional blind denoising of real photographs," *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] S. Lefkimmiatis, "Universal denoising networks : A novel cnn architecture for image denoising," Jun. 2018, **pages** 3204–3213. DOI: 10.1109/CVPR.2018.00338.

[5] X. Jia, S. Liu, X. Feng **and** L. Zhang, "Focnet: A fractional optimal control network for image denoising," **in***2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019, **pages** 6047–6056. DOI: 10.1109/CVPR.2019.00621.

[6] R. Vemulapalli, O. Tuzel **and** M.-Y. Liu, "Deep gaussian conditional random field network: A model-based deep network for discriminative denoising," **in***Proceedings of the IEEE conference on computer vision and pattern recognition* 2016, **pages** 4801–4809.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng **and** L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *Trans. Img. Proc.*, **journvol** 26, **number** 7, 3142–3155, 2017, ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2662206. [Online]. Available: https://doi.org/10.1109/TIP.2017.2662206.

[8] K. Zhang, W. Zuo **and** L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, **journvol** 27, **number** 9, **pages** 4608–4622, 2018. DOI: 10.1109/TIP.2018.2839891.

[9] M. Z. Alom, M. Hasan, C. Yakopcic **and** T. Taha, "Inception recurrent convolutional neural network for object recognition," *Machine Vision and Applications*, **journvol** 32, **january** 2021. DOI: 10.1007/s00138-020-01157-3.

[10] Y. Chen **and** T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *CoRR*, **journvol** abs/1508.02848, 2015. arXiv: 1508.02848. [Online]. Available: http://arxiv.org/abs/1508.02848.

[11] J. Jiao, W.-C. Tu, S. He **and** R. W. H. Lau, "Formresnet: Formatted residual learning for image restoration," **in***2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2017, **pages** 1034–1042. DOI: 10.1109/CVPRW.2017.140.

[12] T. Wang, Z. Dou, C. Bao **and** Z. Shi, "Diff-resnets for few-shot learning - an ODE perspective," *CoRR*, **journvol** abs/2105.03155, 2021. arXiv: 2105.03155. [Online]. Available: https://arxiv.org/abs/2105.03155.

[13] J. Lehtinen, J. Munkberg, J. Hasselgren *et al.*, "Noise2noise: Learning image restoration without clean data," **march** 2018.

[14] J. Batson **and** L. Royer, "Noise2self: Blind denoising by self-supervision," *CoRR*, **jourvol** abs/1901.11365, 2019. arXiv: 1901 . 11365. [Online]. Available: http://arxiv.org/abs/1901.11365.

[15] Y. Quan, M. Chen, T. Pang **and** H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," **in***2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2020, **pages** 1887–1895. DOI: 10.1109/CVPR42600.2020.00196.

[16] A. Krull, T.-O. Buchholz **and** F. Jug, "Noise2void - learning denoising from single noisy images," Jun. 2019, **pages** 2124–2132. DOI: 10.1109/CVPR.2019.00223.

[17] Z. Wang, J. Liu, G. Li **and** H. Han, "Blind2unblind: Self-supervised image denoising with visible blind spots," **in***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2022, **pages** 2027–2036.

[18] T. Huang, S. Li, X. Jia, H. Lu **and** J. Liu, "Neighbor2neighbor: Self-supervised denoising from single noisy images," **in***2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2021, **pages** 14 776–14 785. DOI: 10.1109/CVPR46437.2021.01454.

[19] T. Pang, H. Zheng, Y. Quan **and** H. Ji, "Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising," **in***2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2021, **pages** 2043–2052. DOI: 10.1109/CVPR46437.2021.00208.

[20] O. Ronneberger, P. Fischer **and** T. Brox, "U-net: Convolutional networks for biomedical image segmentation," **in***Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* N. Navab, J. Hornegger **and** A. F. Wells William M.and Frangi, Eds., Cham: Springer International Publishing, 2015, **pages** 234–241, ISBN: 978-3-319-24574-4.

[21] X. Li, Y. Hu, X. Gao, D. Tao **and** B. Ning, "A multi-frame image super-resolution method," *Signal Processing*, **jourvol** 90, **number** 2, **pages** 405–414, 2010, ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2009.05.028. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168409002618.

[22] R. C. Gonzalez **and** R. E. Woods, *Digital Image Processing*. Pearson, 2018.

[23] J. Benesty, J. Chen **and** Y. Huang, "Study of the widely linear wiener filter for noise reduction," **in***2010 IEEE International Conference on Acoustics, Speech and Signal Processing* 2010, **pages** 205–208. DOI: 10.1109/ICASSP.2010.5496033.

[24] I. Pitas **and** A. Venetsanopoulos, *Nonlinear Digital Filters: Principles and applications*. Kluwer Academic Publishers, 1990.

[25] R. Yang, L. Yin, M. Gabbouj, J. Astola **and** Y. Neuvo, "Optimal weighted median filtering under structural constraints," *IEEE Transactions on Signal Processing*, **jourvol** 43, **number** 3, **pages** 591–604, 1995. DOI: 10.1109/78.370615.

[26] L. Fan, F. Zhang, H. Fan **and** C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, **jourvol** 2, **number** 1, 2019. DOI: 10.1186/s42492-019-0016-7.

[27] M. Aharon, M. Elad **and** A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, **jourvol** 54, **number** 11, **pages** 4311–4322, 2006. DOI: 10.1109/TSP.2006.881199.

[28] M. Elad **and** M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, **jourvol** 15, **number** 12, **pages** 3736–3745, 2006. DOI: 10.1109/TIP.2006.881969.

[29] I. Markovsky, *Low-rank approximation: Algorithms, implementation, applications*. Springer, 2018.

[30] G. Liu, Z. Lin **and** Y. Yu, "Robust subspace segmentation by low-rank representation," **in***Proceedings of the 27th International Conference on International Conference on Machine Learning* **jourser** ICML'10, Haifa, Israel: Omnipress, 2010, 663–670, ISBN: 9781605589077.

[31] K. Dabov, A. Foi, V. Katkovnik **and** K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, **jourvol** 16, **number** 8, **pages** 2080–2095, 2007. DOI: 10.1109/TIP.2007.901238.

[32] A. Buades, B. Coll **and** J.-M. Morel, "A non-local algorithm for image denoising," **in***2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* **volume** 2, 2005, 60–65 vol. 2. DOI: 10.1109/CVPR.2005.38.

[33] A. Foi, K. Dabov, V. Katkovnik **and** K. Egiazarian, "Shape-adaptive dct for denoising and image reconstruction - art. no. 60640n," *Proceedings of*

SPIE - The International Society for Optical Engineering, **jourvol** 6064, **pages** 203–214, **february** 2006. DOI: 10.1117/12.642839.

[34] K. Dabov, A. Foi, V. Katkovnik **and** K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, **april** 2009.

[35] W. Feng, S.-M. Li **and** K. Zheng, "A non-local bilateral filter for image denoising," **january** 2011, **pages** 253 –257. DOI: 10.1109/ICACIA.2010.5709895.

[36] Y. Jin, W. Jiang, J. Shao **and** J. Lu, "An improved image denoising model based on nonlocal means filter," *Mathematical Problems in Engineering*, **jourvol** 2018, **pages** 1–12, Jul. 2018. DOI: 10.1155/2018/8593934.

[37] S. Anwar, C. P. Huynh **and** F. Porikli, "Combined internal and external category-specific image denoising," Sep. 2017. DOI: 10.5244/C.31.71.

[38] S. Anwar, F. Porikli **and** C. P. Huynh, "Category-specific object image denoising," *IEEE Transactions on Image Processing*, **jourvol** 26, **number** 11, **pages** 5506–5518, 2017. DOI: 10.1109/TIP.2017.2733739.

[39] E. Luo, S. H. Chan **and** T. Q. Nguyen, "Adaptive image denoising by targeted databases," *IEEE Transactions on Image Processing*, **jourvol** 24, **number** 7, **pages** 2167–2181, 2015. DOI: 10.1109/TIP.2015.2414873.

[40] H. Yue, X. Sun, J. Yang **and** F. Wu, "Image denoising by exploring external and internal correlations," *IEEE Transactions on Image Processing*,

jourvol 24, **number** 6, **pages** 1967–1982, 2015. DOI: 10.1109/TIP.2015.2412373.

[41] V. Jain **and** S. Seung, "Natural image denoising with convolutional networks," **in***Advances in Neural Information Processing Systems* D. Koller, D. Schuurmans, Y. Bengio **and** L. Bottou, Eds., **volume** 21, Curran Associates, Inc., 2008. [Online]. Available: https://proceedings.neurips.cc/paper/2008/file/c16a5320fa475530d9583c34fd356ef5-Paper.pdf.

[42] P. Vincent, H. Larochelle, Y. Bengio **and** P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," **in***Proceedings of the 25th International Conference on Machine Learning* **jourser** ICML '08, Helsinki, Finland: Association for Computing Machinery, 2008, 1096–1103, ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. [Online]. Available: https://doi.org/10.1145/1390156.1390294.

[43] J. Xie, L. Xu **and** E. Chen, "Image denoising and inpainting with deep neural networks," **in***Advances in Neural Information Processing Systems* F. Pereira, C. Burges, L. Bottou **and** K. Weinberger, Eds., **volume** 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper.pdf.

[44] B. Marcelo, *Denoising of photographic images and video fundamentals, open challenges and new trends*. Springer International Publishing, 2018.

[45] S. Cha, T. Park **and** T. Moon, "Gan2gan: Generative noise learning for blind image denoising with single noisy images," **may** 2019.

[46] S. Laine, T. Karras, J. Lehtinen **and** T. Aila, "High-quality self-supervised deep image denoising," **in***Proceedings of the 33rd International Conference on Neural Information Processing Systems* Red Hook, NY, USA: Curran Associates Inc., 2019.

[47] N. Moran, D. Schmidt, Y. Zhong **and** P. Coady, "Noisier2noise: Learning to denoise from unpaired noisy data," **in***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2020, **pages** 12 064–12 072.

[48] D. Martin, C. Fowlkes, D. Tal **and** J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," **in***Proc. 8th Int'l Conf. Computer Vision* **volume** 2, 2001, **pages** 416–423.

[49] R. W. Franzen. [Online]. Available: `http://r0k.us/graphics/kodak/`.

[50] R. Zeyde, M. Elad **and** M. Protter, "On single image scale-up using sparse-representations," **in***Curves and Surfaces* J.-D. Boissonnat, P. Chenin, A. Cohen *et al.*, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, **pages** 711–730, ISBN: 978-3-642-27413-8.

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

[52] J. Xu, H. Li, Z. Liang, D. Zhang **and** L. Zhang, "Real-world noisy image denoising: A new benchmark," **april** 2018.

[53] S. Nam, Y. Hwang, Y. Matsushita **and** S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image

denoising," **in***Proceedings of the IEEE conference on computer vision and pattern recognition* 2016, **pages** 1683–1691.

Acknowledgements

I want to express my gratefulness to almighty ALLAH and the individuals who have supported me in the process of completing my Master's degree and research. Firstly, I really would like to take this opportunity to express my gratitude to Professor Ho Yub Jung, my supervisor, for allowing me to pursue my Master's degree at Chosun University. His constant inspiration, support, and insightful recommendations have led and pushed me throughout my studies and research. His continuous supervision and direction have aided me in producing high-quality research. I will be eternally grateful to him for instilling in me the values of professionalism, organizational skills, and concentration. Furthermore, I am glad for the opportunity to work in the Department of Computer Engineering at Chosun University with such a diversified batch of students, teachers, and staff. I want to thank Computer Vision Lab for giving me such an excellent opportunity and an environment to develop academically. My lab colleagues have been a source of moral and intellectual support for me. In addition, I want to express my appreciation to all of my Bangladeshi seniors and friends at Chosun University for their compassion and cooperation in making my life in South Korea easy and joyful. Lastly, I want to express my gratitude to my wife, and parents for their constant and unwavering support throughout my difficult times. It would have been difficult for me to do anything without their motivation and direction.