IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 11, 2022, accepted March 20, 2022, date of publication March 28, 2022, date of current version April 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162608

Gradient and Multi Scale Feature Inspired Deep Blind Gaussian Denoiser

RAMESH KUMAR THAKUR, (Member, IEEE), AND SUMAN KUMAR MAJI, (Member, IEEE)
Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801103, India
Corresponding author: Ramesh Kumar Thakur (ramesh.pcs17@iitp.ac.in)

ABSTRACT In this paper, a novel deep blind Gaussian denoising network is proposed utilizing the concepts of gradient information, multi-scale feature information and feature denoising for removing additive white Gaussian noise (AWGN) from images. The proposed network consists of two modules where in the first module generates an intermediate image whose gradient information is concatenated with the features of second module to generate the final residual image. Subtracting this residual image with the noisy image gives the desired denoised image. The feature denoising block used in the middle of the first module enhances the feature information of the intermediate image. The usage of gradient information of this intermediate denoised image, together with the multi scale feature information block, in the second module further contributes to the quality of the final denoised image. Experimental results show superior denoising performance of the proposed method in comparison to several state of the art classical and learning based blind denoising methods like EPLL, BM3D, WNNM, DnCNN, MemNet, BUIFD, Self2Self and ComplexNet by a decent margin (an improvement of up to 2.4dB in PSNR, 0.07 in SSIM and 0.03 in FOM index with the second best performing model) when experimented over BSD68, Set5, Set14, SunHays80 and Manga109 image databases.

INDEX TERMS Gradient information, multi-scale feature information, additive white Gaussian noise, residual image, feature denoising block.

I. INTRODUCTION

The advent of multimedia and its ever increasing utility devices have created an enormous content of data mostly realized in the form of images and videos. Processing of this data forms the core of many computer vision tasks. The presence of noise, inherent in the data due to several factors related to the acquisition procedure, poses a limitation on their processing and further applications. Noise removal or denoising therefore forms an intrinsic post-processing task for image related applications. Over the years, denoising of images have evolved from the conventional filtering based methods [1] to model-driven optimization techniques [2] to the more recent and highly researched learning based techniques. Learning based techniques have in fact proved to be a trend setter in the area of image denoising with multiple deep learning based methods seen to outperform (and at times outclass) [3], both in terms of visual quality as well as quantitative metrics, their filtering and model-driven based counterparts. The traditional limitations of manual parameter setting for filtering based methods and regularization parameter setting for model-driven techniques, for a fixed scenario, is overcome in the learning based scenario. Also much of the filtering and optimization based denoisers fail to denoise properly in the case of various noise level [4], which is also overcome in today's deep learning based methods.

In the sphere of learning based techniques, convolutional neural network (CNN) based methods are highly popular and widely investigated frameworks for image denoising. The concept of CNN based denoisers was first investigated in [5] where in the authors proposed a convolutional network for image denoising which produced better output than wavelet based methods as well as hidden Markov models. This CNN based algorithm did not suffer from the computational difficulty in probability learning which was faced by Markov based models. Thereafter some more methods were proposed [4], [6] before the advent of the highly popular denoising convolutional neural network or DnCNN [7] framework that was designed to restore images corrupted with AWGN of unknown variance (blind denoising). Neural networks have

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

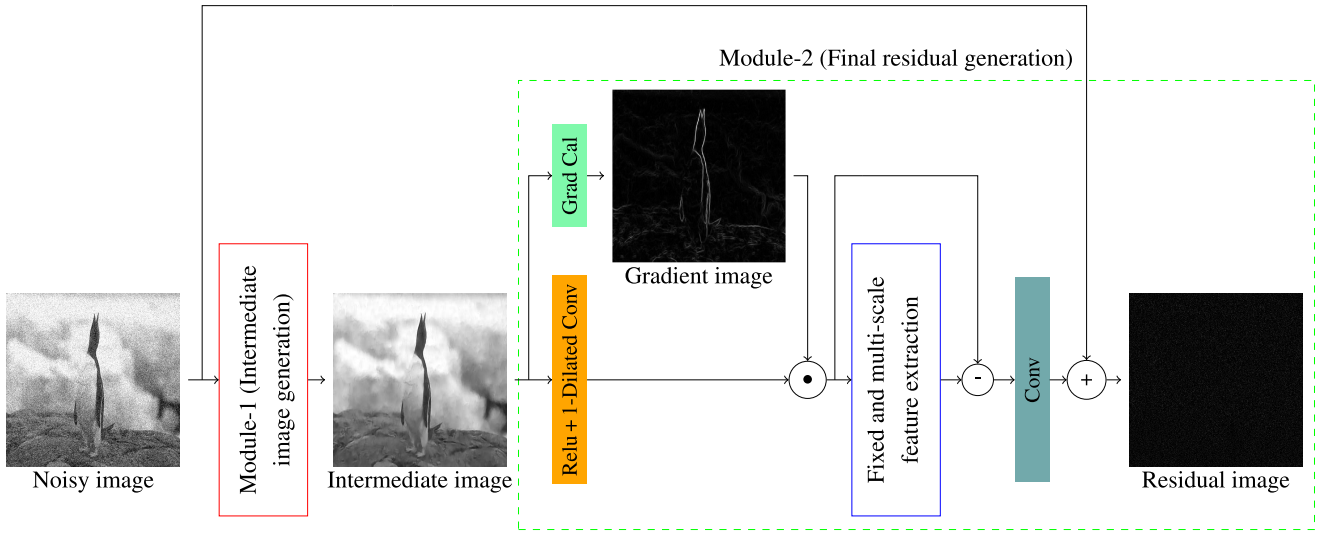


FIGURE 1. Block diagram of proposed network (⊙ denotes concatenation).

the capacity to model complex functions [8] by using the concept of batch normalization [9] and residual learning [10]. Sometimes complex function modelling is not appropriate and it becomes difficult to design a general CNN model which deals with multiple levels of noise. This leads to the necessity of developing blind image denoisers that do not require any prior information of the noise levels for denoising. This particular property of blind denoisers is very useful for real time image denoising also as such cases one do not have accurate (sometimes nil) information of the noise present in the image. Current blind denoising models, although beneficial, suffer from certain drawbacks. Some of the earlier deep learning based networks [4], [7], [11], [12], that deal with blind image denoising, follow a brute force technique in order to learn the weights of the network. In order to increase the performance of these blind denoisers, more weight layers have to be added which leads to the vanishing gradients problem [13].

In this paper, we propose a blind Gaussian denoising network (refer to Fig. 1) by using the concepts of gradient information, multi-scale feature information and feature denoising for removing additive white Gaussian noise (AWGN) from images. The proposed network takes noisy image as input and produces residual noise as output which is then subtracted from the noisy image to get the final denoised output. The main contributions of proposed network are as follows: -

1) Simultaneous incorporation of gradient information, feature denoising block and multi-scale feature information block in a deep neural framework for superior denoising.
2) Explicit introduction of gradient information into the deep neural network by Sobel filter to enhance information content (such as contours etc.) of the feature map.
3) A multi-scale feature information block is proposed to incorporate features of intermediate image in higher resolution as well as in normal resolution to further enhance the feature extraction capability.
4) Superior denoising in comparison to several state of the art classical and learning methods like EPLL, BM3D, WNNM, DnCNN, MemNet, BUIFD, Self2Self and ComplexNet by a decent margin (an improvement of up to 2.4dB in PSNR, 0.07 in SSIM and 0.03 in FOM index with the second best performing model) when experimented over BSD68, Set5, Set14, SunHays80 and Manga109 image databases.

II. RELATED WORK

Non learning based denoising methods can be divided into spatial domain denoising approach and frequency domain denoising approach. KSVD [14] is a dictionary based spatial domain denoising technique where in the authors proposed a K-means clustering based training algorithm on an over-complete dictionary that best suits a set of given signals. BM3D [15] is the most popular spatial domain denoising approach where in the authors used collaborative filtering technique applied on blocks to denoise Gaussian noise corrupted images. In EPLL [16], authors proposed a generic framework for image denoising using any patch based prior for which a MAP estimate can be calculated. Chen *et al.* proposed a noise level estimation method [17] by establishing a relationship between the noise level and the eigenvalues of covariance matrix of patches. They proposed a non-parametric algorithm to estimate the noise variance from the eigenvalues in polynomial time. With the help of this noise level estimation method, denoising performance of tradition statistical method can be improved. In frequency domain approaches, WNNM [18] is one of the most popular approach where in the authors proposed a weighted nuclear norm minimization-based image denoising approach by exploiting the nonlocal self-similarity of images.

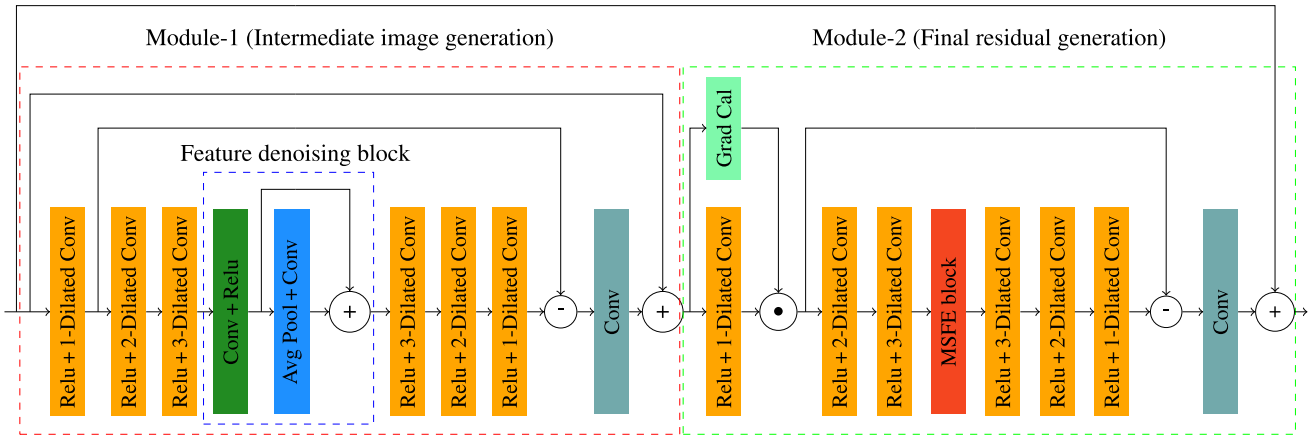


FIGURE 2. Proposed CNN architecture ($\odot$ denotes concatenation).

Khmag *et al.* proposed an adaptive denoising technique [19] based on second-generation wavelet domain using hidden Markov models (SGWD-HMMs). They utilize the fact that the images are non-stationary with singularities and some smooth areas that can be considered as stationary. In another work, Khmag et al proposed a nonlocal means and hidden Markov model based denoising approach [20]. In this work, hidden Markov model is proposed to capture the dependency between additive white Gaussian noise pixel and its neighbors on the wavelet transform. Both the spatial domain and frequency domain denoising techniques have some inherent limitations. Let aside the rigorous statistical analysis, both the classes suffer from the drawback of manual parameter setting for a fixed noise level (which may not work for other noise levels) to get the desired denoising. With the advancement in deep learning methods and availability of large-scale database, the CNN based image denoising techniques came into existence and became need of the hour.

Learning based denoising methods [21]–[30] can be broadly classified into two categories - blind and non-blind. Non-blind denoising techniques work on the principle of having the noise level parameters (associated with the noisy image) known before hand, where as in the case of blind denoising such parameters have to be estimated during the denoising process. Based on these two basic categories, we will summarize the previous works on image denoising that has been done using neural networks.

In the category of non-blind denoising techniques, a fast and flexible CNN based network called FFDNet was proposed in [31]. The network considers a tunable noise level map as its input, whereby it is able to handle noise of different levels in images for their denoising thereby offering a much needed flexibility to the problem of denoising images of different noise levels through a single network. A set of fast and effective CNN denoiser for non-blind AWGN denoising is proposed in [11] where the denoising network is trained multiple times for different level of Gaussian noise. A class-aware fully convolutional non-blind Gaussian denoising method is proposed by Remez *et al.* [32], where the network require class information of noisy image before denoising. Here the denoising network is trained separately for each of the five fixed class of images. A universal non-blind denoising network which deals with multiple degradation is proposed by Zhang *et al.* [33] in which noisy image along with degradation maps is needed for denoising task. These degradation maps are constructed by stretching the blur kernel and noise level. A non-blind CNN network for denoising AWGN corrupted image is proposed by Uchida *et al.* [34]. Here the denoising network take noisy image and degradation attribute channels as input and gives residual noise as output which is then further subtracted from noisy image to produce denoised output. An adaptive non-blind denoising network called ATDNet is proposed in [35] where a noise level indicator network is implemented to denoise an AWGN corrupted image. Here a baseline denoising network is used along with gate-weight generating network for effectively denoising the image. A recent non-blind denoising method named as attention residual convolutional neural network (ARCNN) is proposed in [36] where the network first try to learn the underlying noise expectation using an attention - residual mechanism followed by the denoising process. Non-blind denoising methods performs well, but the prior information of the input noise level parameters that it needs for denoising are not always available in practice thereby posing a serious limitation to their usage.

This limitation of the non-blind denoising techniques is being overcome in blind denoising networks, where in the noise level parameters associated to the noisy image are estimated as a part of the denoising procedure. This makes the non-blind techniques more popular and also flexible in terms of usage. One of the most popular and highly used state-of-the-art blind denoiser for Gaussian denoising is the denoising convolutional neural networks or DnCNN [7]. In this model, a deep CNN network is trained on randomly-sampled patches of multiple noise levels to develop a general denoiser for noisy image of particular range of noise level. Many blind

and non-blind [37] recent methods has been outperformed by DnCNN. A residual learning based blind Wavelet denoising CNN method (WDnCNN) is proposed in [38] where the architecture is same as DnCNN and the training of network is done using four decomposed wavelet sub-bands. A residual learning blind denoising network (SCNN) is proposed in [39] where different levels of noise in input image is tackled by using soft shrinkage activation function. A blind residual network having same architecture as DnCNN without batch normalization (IDCNN) is proposed in [40]. IDCNN faces the problem of non-convergence with stochastic gradient descent due to gradient explosion. To fix this problem, gradient is clipped in pre-defined interval. IDCNN uses a non-fixed noise mask for handling different noise levels by a single model. DnCNN based denoising networks face problem in the case of unpaired noisy images i.e. when the noisy image does not have corresponding ground truth image for training purpose. To solve this problem of unpaired noisy image, a generative adversarial network (GAN) based blind CNN denoiser [41] is proposed where the ground truth image is generated first and then ground truth is given input to the GAN for training the denoiser. A residual learning based blind denoising network (ECNDNet) having same architecture and loss function as that of DnCNN is proposed in [42]. ECNDNet has difference with DnCNN in the sense that it used dilated convolution to enhances the receptive field of network. It also extracts more context information and decrease the computational cost. A noise estimation based blind CNN denoiser(CBDNet) is proposed in [43] where two sub-networks are used for denoising. One sub-network estimate the noise information whereas the second sub-network provides the denoised image. MemNet [12] is one recent blind gaussian denoising CNN network which performs slightly better than DnCNN and uses adaptive learning process. To solve the problem of internal co-variate shift for extracting more number of features, batch-renormalization denoising network (BRDNet) is proposed in [44]. In BRDNet, dilated convolution, batch-renormalization and residual learning is used to tackle the problem of internal co-variate shift. A receptive field size variation based deep iterative down-up network (DIDN) was proposed in [45]. DIDN is a blind denoising network which comprises of four stages: initial feature extraction block, down-up block, reconstruction block, and enhancement block. Here convolution operation is applied for initial feature maps extraction then these feature maps are fed into down-up block for iterative down and up-sampling. The down and up-sampled feature maps are given as input to the reconstruction block which is made of convolutional and parametric rectified linear units. At last all the outputs of reconstruction block are being concatenated and fed into the enhancement block which finally produce a denoised image. A UNet [46] architecture based blind denoising network (DHDN) is proposed in [47] which tackle vanishing gradient problem by using residual network and dense connectivity to convolution layers. Another UNet based blind denoising architecture (MWCNN) is proposed in [48] where combination of UNet architecture and multi wavelet transform is used to enhance the receptive field size by decreasing the feature map's resolution. To solve vanishing gradient problem, a GAN based blind denoising network is proposed in [49]. Here DenseNet architecture [50] is used as the generator network which predict the ground truth and the discriminator network reduces the difference between original ground-truth and the output of generator network. An attention-guided blind denoising network (ADNet)is proposed in [51] which tackle the problem of increase in network length. Fully convolutional encoder-decoder architecture with skip connections is used in ADNet for blind Gaussian denoising. One of the latest blind gaussian denoiser is BUIFD [52] which outperformed state-of-the-art DnCNN, MemNet and the best classic non-learning denoiser BM3D [15]. BUIFD method is based on fusion denoising and theoretically derived from the assumption of Gaussian image prior. Self2Self [53] is a self-supervised learning based denoising method that uses only the input noisy image for training purpose. Here the network is trained with dropout on the pairs of Bernoulli-sampled instances of the input image, and the result is estimated by averaging the predictions generated from multiple instances of the trained model with dropout. In ComplexNet [54] method, a complex-valued CNN for image denoising is proposed. Here the authors define several basic operations in the complex number field to exploit possible advantages over their counterparts in the real number field. SwinIR [29] is a Swin Transformer-based image denoising method where the authors proposed a denoising network consisting of three parts: shallow feature extraction, deep feature extraction and high-quality image reconstruction. This method used several residual Swin Transformer layers in deep feature extraction module for local attention and cross-window interaction. In [30], authors proposed a model-agnostic method for reducing epistemic uncertainty in Gaussian denoising while using a single pretrained network. DBDNet [27] is a residual denoising network in which the network first generates a noise map from noisy image and then updates the noise map gradually with the help of a boosting function.

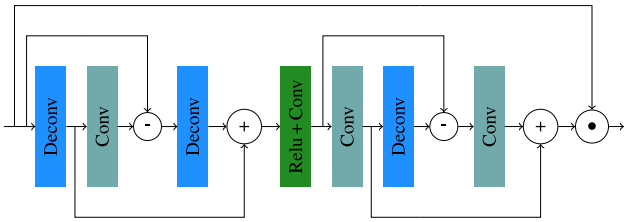


FIGURE 3. MSFE block architecture (⊙ denotes concatenation).

III. METHODOLOGY

A. NETWORK ARCHITECTURE

The architecture of proposed Gradient based Blind Gaussian Noise Removal Network(GBGNRNet) is demonstrated

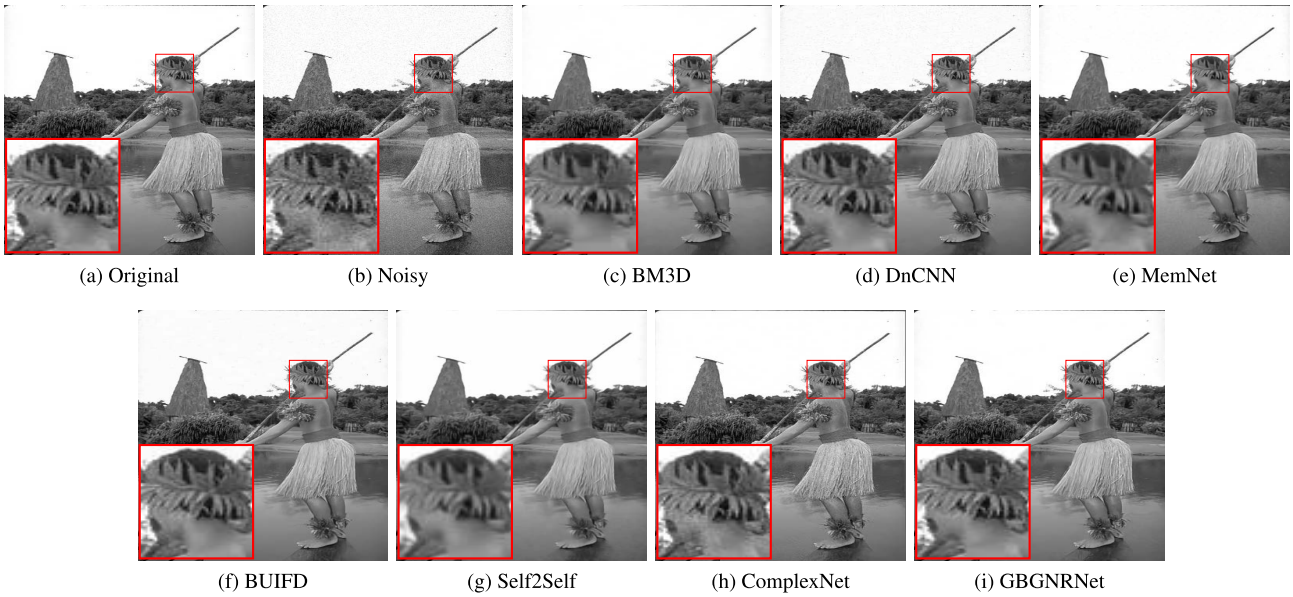


FIGURE 4. Visual result of denoising of grayscale image ID 01 from the (ordered) BSD68 for gaussian noise (*level* = 10).

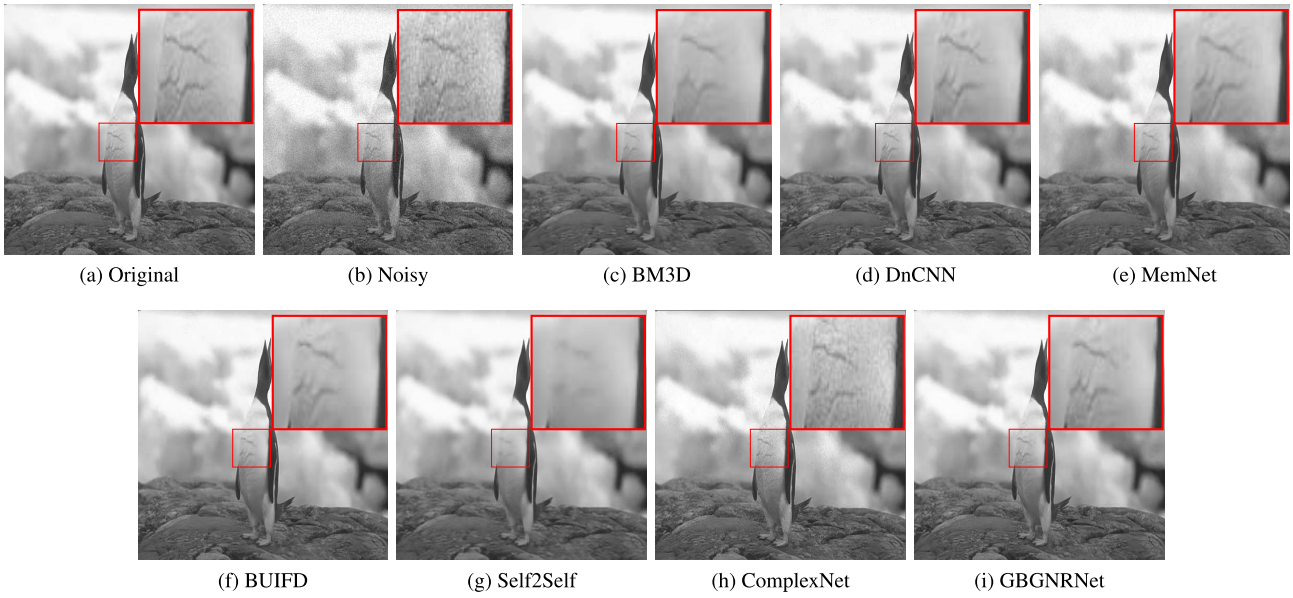


FIGURE 5. Visual result of denoising of grayscale image ID 05 from the (ordered) BSD68 for gaussian noise (*level* = 10).

in Fig. 2. The proposed GBGNRNet is composed of two modules: the first one is the intermediate image generation module and the second one is the final residual generation module. Combining these two modules, GBGNRNet can be defined as:

$$\text{GBGNRNet}_{out} = \text{GBGNRNet}_{in} - [m_{frg}(m_{iig}(\text{GBGNRNet}_{in}))] \quad (1)$$

where GBGNRNet_{in} and GBGNRNet_{out} represents the input and output of proposed GBGNRNet while m_{iig} and m_{frg} denotes the operation of the first and second module respectively. GBGNRNet_{in} represents an AWGN corrupted noisy image while GBGNRNet_{out} denotes the final residual noise which should be subtracted from input noisy image to get desired denoised image.

B. MODULE-1 (INTERMEDIATE IMAGE GENERATION)

Here a combination of ReLU layer followed by a dilated convolution layer(ReLU + Dilated CONV) is used to extract features of noisy image which is enhanced by a feature denoising block. This feature denoising block [55] mainly consist of a mean filter for denoising noisy image features. There are

three main reasons for choosing mean filter for denoising features: (1) adversarial robustness can be improved by mean filter [56], (2) the number of parameters are not increased by using mean filter and (3) with the help of average pooling operation, mean filter can be implemented smoothly. After performing feature denoising operation, ReLU + Dilated CONV layer is used again to generate intermediate image. Here successive increase and decrease in dilation rate of convolutional layer is applied to enhance the feature extraction capability [11]. Here skip connections are used such that neurons in the last layer can observe the full spatial receptive field of the neurons of first convolution layer.

C. MODULE-2 (FINAL RESIDUAL GENERATION)

Here, the intermediate image generated as output from first module is taken as input whose gradient is calculated by Grad Cal block which is discussed in subsection III-D. The feature information of intermediate image is calculated by ReLU + Dilated CONV block and is concatenated with gradient of intermediate image for further feature extraction. The position of gradient concatenation is suggested and explained in [57]. The extracted features are sent to MSFE block for multi-scale feature extraction whose detailed information is discussed in subsection III-E. After performing multi-scale feature extraction, a final residual image is generated by applying again ReLU + Dilated CONV block followed by a convolution layer. Here also several skip connections are used to preserve image details which is useful in reconstruction. Here residual learning strategy is applied for obtaining higher accuracy. The final output of GBGNRNet is the residual noise i.e. the difference between noisy input image GBGNRNet_{in} and the desired denoised image which has to be subtracted from input noisy image to get the final denoised image.

D. GRADIENT CALCULATION (GRAD CAL) BLOCK

For fusing the gradient information [57] into the second module of proposed network, Sobel filter [58] is used to calculate the gradient of intermediate image. Before calculating the gradient, one horizontal filter S_H and one vertical filter S_V is defined as below:

$$S_H = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad S_V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

A convolution operation is then performed between intermediate image $\boldsymbol{r}$ with S_H and S_V to get horizontal gradient $\boldsymbol{G}_H$ and vertical gradient $\boldsymbol{G}_V$ as shown below:

$$\boldsymbol{G}_H(\boldsymbol{r}) = \boldsymbol{S}_H * \boldsymbol{r}, \quad \boldsymbol{G}_V(\boldsymbol{r}) = \boldsymbol{S}_V * \boldsymbol{r}$$

where * represents the convolution. Finally the image gradient $\boldsymbol{G}(\boldsymbol{r})$ is calculated as:

$$\boldsymbol{G}(\boldsymbol{r}) = \sqrt{\boldsymbol{G}_H(\boldsymbol{r})^2 + \boldsymbol{G}_V(\boldsymbol{r})^2}$$

where the square and square root are element-wise matrix operations.

E. MULTI-SCALE FEATURE EXTRACTION (MSFE) BLOCK

For utilising the features of multiple scales, a multi-scale feature extraction (MSFE) block is applied in the middle of second module whose detailed architecture is shown in Fig. 3. Earlier methods [59] which have used multi-scale feature extraction have many layers for multi-scale feature extraction which can result into vanishing gradient problem but we designed a 7 layer small MSFE block which does not let the network to suffer from vanishing gradient problem. Because of symmetric up and down sampling structures present in the MSFE block, the network see more image context information at training phase. Such symmetric up sampling and down-sampling structure was first proposed in [60], but we have incorporated a Relu+Conv layer over it for increasing the feature content at higher scale. A self-correcting mechanism is applied by using Up and Down-Projection unit which feed a projection error to the sampling layer and the solution is updated iteratively with the help of projection error as feedback. By applying Up and Down Block-CNN, MSFE block extracts features present in the higher resolution. The fixed scale features, which are calculated before this block, are concatenated with the features of higher resolution and leads to realization of multi-scale feature extraction. The output of MSFE block is the concatenation of features in different scales.

F. OBJECTIVE FUNCTION

Here image patches are used to train the proposed network. There are two main reasons for choosing image patches for training instead of the whole image. First reason is the random sampling of patches taken from different locations of the training image. These randomly shuffled training patches stabilize the training process in deep CNN networks. So, batch training of these patches is preferred with a random mixture of shapes, patterns and local structures. Second reason behind choosing image patches is the impressive denoising results achieved [16] by approaches which use image patches for training.

Suppose the input-output pairs to train the proposed network are $(x_j, y_j)_{j=1}^N$ where x and y are related as $y_j = x_j + n_j$. Here x_j denotes ground truth image patch and y_j represents noisy image patch. If G represent the gradient filter then the learning objective of the proposed network is defined as minimizing the given loss function:

$$\mathcal{L} \triangleq \frac{1}{2N} \sum_{j=1}^{N} \left\| \text{GBGNRNet}\left(y_j, G\right) - (y_j - x_j) \right\|^2 \qquad (2)$$

If training is performed on big dataset, then the loss function is minimized using mini-batches of training samples. The detailed training information is explained in Section IV-A.

IV. EXPERIMENTS

A. GBGNRNET EXPERIMENTAL SETTINGS

For training the proposed network, 400 images from BSD400 [61] database is used. In each training step,

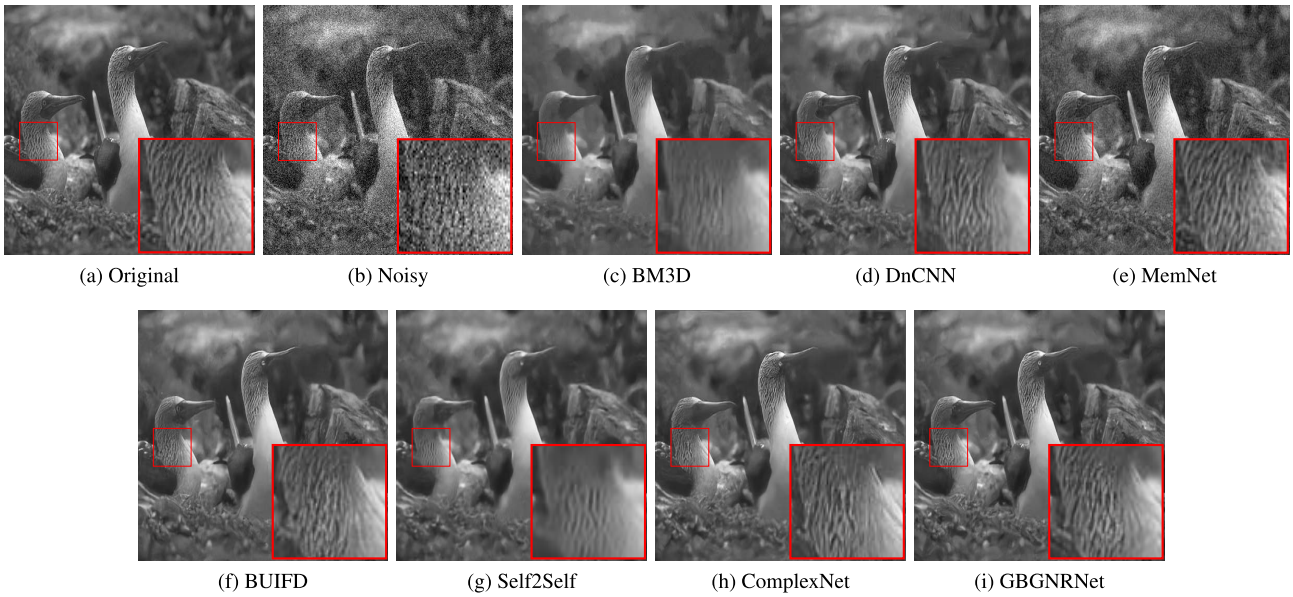


FIGURE 6. Visual result of denoising of grayscale image ID 03 from the (ordered) BSD68 for gaussian noise (*level* = 25).

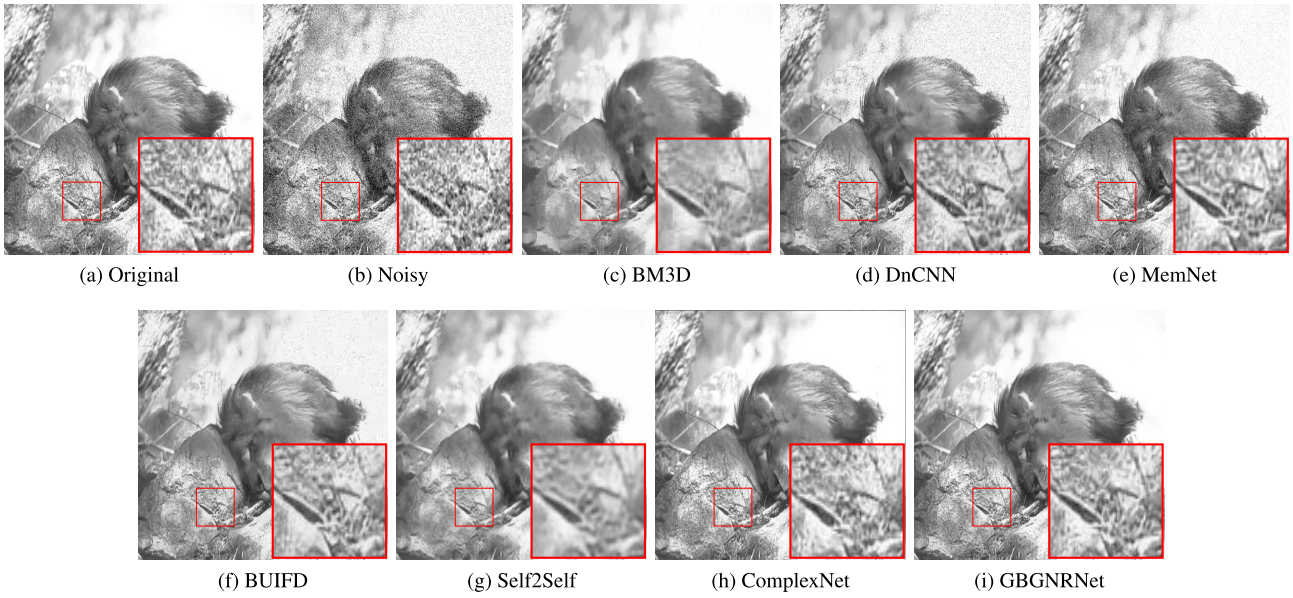


FIGURE 7. Visual result of denoising of grayscale image ID 04 from the (ordered) BSD68 for gaussian noise (*level* = 25).

image patches of size 30 × 30 is randomly cropped and augmented by applying flip and rotation operation to increase the number of training patches. After applying all augmentation operation, we get 266032 patches for training the proposed denoising network. Normalization of pixel intensity of training data is also performed by dividing it by 255 to restrict the intensity value in the range [0, 1].

Adam optimizer [18] with default parameters is used to train the proposed network. Here learning rate is initialized with 10^{-3} and it is scheduled such that it becomes half in every 5^{th} epoch. The proposed network is trained for 30 epochs by using mini-batches of size 32. The training of proposed network is performed by adding AWGN of multiple standard deviation, where the value of standard deviation is selected randomly from the range [0,55]. For implementation purpose, we used Keras and Tensorflow on a system having 11GB NVIDIA GeForce GTX 1080 Ti GPU. It takes 5 hours 20 minutes to train the proposed denoiser for gray image denoising.

B. IMAGE DENOISING EVALUATION

We have used nine state-of-the-art blind denoising techniques for comparing the proposed model both quantitatively as

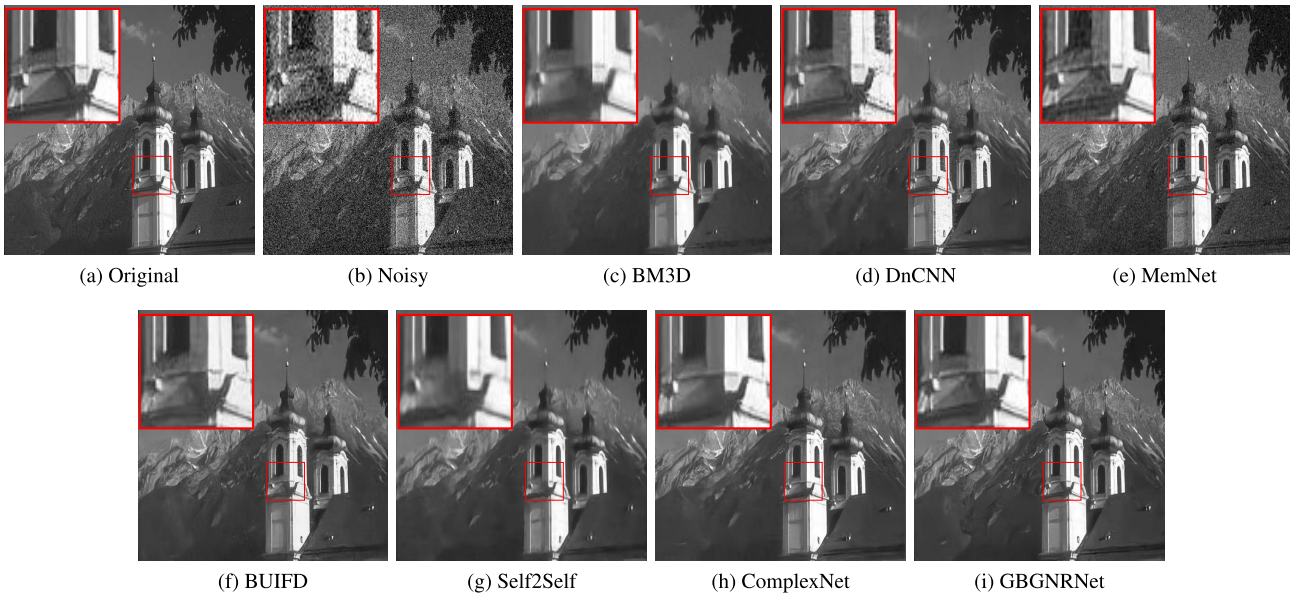


FIGURE 8. Visual result of denoising of grayscale image ID 13 from the (ordered) BSD68 for gaussian noise (*level* = 25).

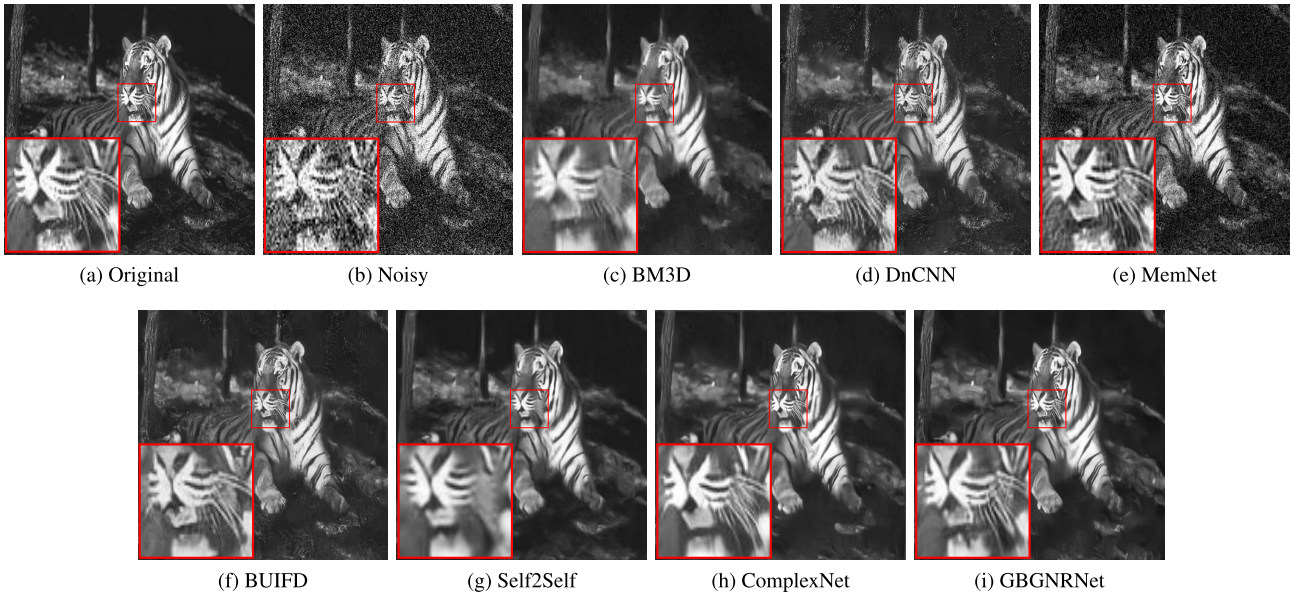


FIGURE 9. Visual result of denoising of grayscale image ID 08 from the (ordered) BSD68 for gaussian noise (*level* = 50).

well as qualitatively. The names of comparing techniques are:- KSVD [14], EPLL [16], BM3D [15], WNNM [18], DnCNN [7], MemNet [12], BUIFD [52], Self2Self [53] and ComplexNet [54]. A brief discussion about the comparing techniques are given in section II. PSNR, SSIM [67] and FOM metrics are used for quantitative comparison of denoised outputs by different denoising techniques performed on three synthetic and two real image databases. The quantitative results in terms of PSNR and SSIM are shown in Table 1 whereas the FOM values are shown in Table 2. By considering practical scenario, the intensity range of images are restricted to $[0, 255]$. The proposed network outperforms all the comparing methods in terms of PSNR, SSIM and FOM which can be seen in Table 1 and Table 2. While comparing with other techniques for unseen noise level (60), then also the proposed method outperforms them by a decent margin in PSNR, SSIM and FOM. The proposed method is also compared with learning-based methods using maximum GPU memory uses, number of parameters and depth of network to show the lightweight nature of proposed network. The code of proposed method is uploaded at https://github.com/RTSIR/GBGNRNet.

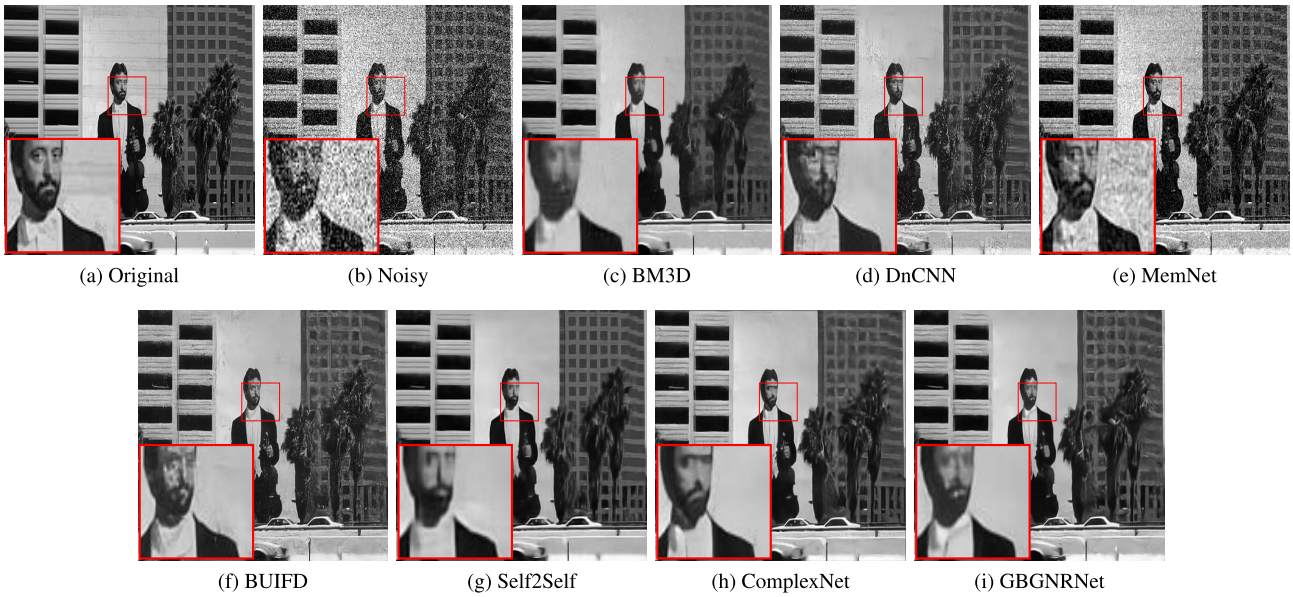
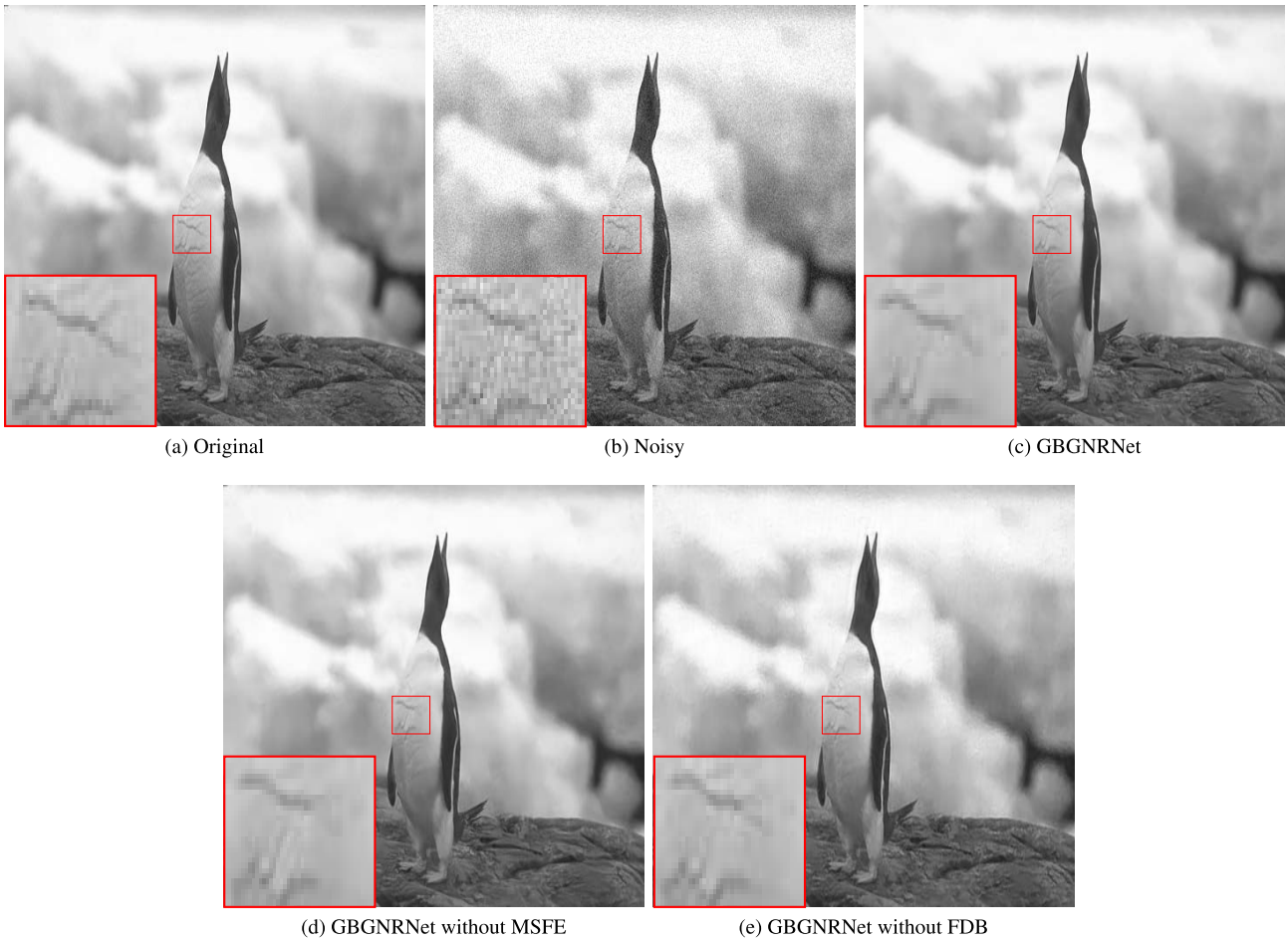


FIGURE 10. Visual result of denoising of grayscale image ID 10 from the (ordered) BSD68 for gaussian noise (*level* = 50).

The visual comparison of denoising of seven gray images from BSD68 [61] database is shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10. By analyzing the output of proposed technique with the comparing methods, it can be seen that it performs better denoising by preserving finer details in the high-frequency region while removing the noise in the low frequency regions. Other comparing methods give blurry outputs with noisy edge because of smudging effect which is visible in the low-frequency regions. The proposed technique does not have this blur problem and supersede all the comparing techniques. By closely observing Fig. 4, it can be seen from the zoomed red box that only the proposed method retains the sharp edges of the leaf structure of the hair band while other approaches blur the leaf structures. In Fig. 5, it can be seen from the zoomed red box highlighting the stomach of penguin that only the proposed method retains the fine structure without artefacts. In Fig. 6, it can be observed that the comparing techniques either blur the output image or introduce artefacts thereby failing to preserve the fine fur structure on the neck of the bird whereas the proposed technique retains the minute fur details which can be clearly seen in zoomed portion of red box. In Fig. 7, the minute structures on rock is properly reconstructed by proposed method while other techniques fail to retain these patterns and it is shown in zoomed portion. If we take a closer look in Fig. 8, it can be seen that all the comparing approaches blur the output image which can be clearly observed in their zoomed section. The corners and edges of the tower are properly reconstructed in the output of the proposed technique only. In Fig. 9, it is demonstrated clearly that the moustache hair of the tiger is not properly reconstructed in other comparing approach while the proposed method reconstructs it properly and is clearly visible in zoomed portion. In Fig. 10, it is observed that the face and tie of the man are better reconstructed by the proposed approach while the other techniques blur the output or add artefacts. By closely observing all the visual comparison figures, it can be deduced that the proposed method provides a non-blurred, artefact free clean output whereas the output of comparing techniques suffer from blur or artefact or both.

C. EXTENDED BENCHMARK COMPARISON

Multiple experiment has been performed on the proposed network by using several benchmark datasets and comparison done with existing state-of-the-art techniques. We have used BSD68, Set5, Set14, SunHays80, and Manga109 datasets corrupted with noise levels 10 to 60 (with a step size of 10). Set5 and Set14 contains 5 and 14 traditionally used images respectively, which are widely used for comparing image denoising approaches. Most of the images in Set5 and Set14 are smaller than 512×512. The SunHays80 dataset comprises of 80 high-resolution images having size smaller than 1024×1024. The Manga109 dataset is made up of 109 drawings which are drawn by professional artists. Each image in this database is of size of 827×1170. The denoising results of various blind non-learning techniques like KSVD, BM3D, WNNM and EPLL are given in Table 1 and Table 2. These methods are commonly used for Gaussian denoising. For comparing with state-of-the-art learning- based blind denoising methods, DnCNN, MemNet, BUIFD, Self2Self and ComplexNet are used. Every learning-based method, used for comparison, has been trained to denoise in the noise level [0,55]. The PSNR, SSIM and FOM values of all the techniques are presented in Table 1 and Table 2 where in the best values has been shown in boldface. Proposed method shows an improvement of up to 2.4dB in PSNR, 0.07 in SSIM and 0.03 in FOM index from second best denoising method.



(a) Original (b) Noisy (c) GBGNRNet

(d) GBGNRNet without MSFE (e) GBGNRNet without FDB

FIGURE 11. Ablation study of visual result of denoising of grayscale image ID 10 from the (ordered) BSD68 for gaussian noise (*level* = 10).

Visual comparison result has been shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10 respectively.

D. EXECUTION TIME COMPARISON

Table 3 shows the execution times (GPU) of multiple techniques for denoising images of size 256 × 256 and 512 × 512 with noise level 25. The running time of the proposed GBGNRNet method is faster than models like BM3D, MemNet, BUIFD, Self2Self and ComplexNet. The proposed technique takes more time to execute than the DnCNN technique in one case, but taking the image denoising quality and implementation framework into consideration, the proposed network is one of the fastest blind denoising method with superior reconstruction results. In Fig. 12, the graph of run time comparison of proposed method at several range of noise density ([0,60]) is shown. By observing this graph closely, it can be deduced that proposed technique requires similar time to denoise image having any level of noise density between 0 to 60.

E. MAXIMUM GPU MEMORY USES COMPARISON

Table 4 shows the comparison of maximum GPU memory consumption of proposed network with four recent CNN based denoising methods on images of size 256 × 256 and 512 × 512 having noise level 50. Here it can be deduced that the proposed GBGNRNet model takes less maximum GPU memory than other learning based methods for denoising images of size 256 × 256 and 512 × 512 whereas DnCNN gives second best result in terms of maximum GPU memory consumption.

F. NUMBER OF PARAMETERS AND DEPTH OF NETWORK COMPARISON

Table 5 represents the comparison of number of parameters and depth of network. The proposed network outperforms the recent CNN based denoising techniques except DnCNN in terms of number of parameters as well as depth of network. By taking visual quality and other metrics into consideration, one can say that proposed method is better than recent blind denoising techniques including DnCNN. The depth of proposed network is restricted to 23 so that the chance of occurring vanishing gradient problem becomes very less compared to other deeper networks.

G. ABLATION STUDY

The ablation study of proposed network is shown in table 6 by showing the impact of different blocks used in the proposed

TABLE 1. PSNR & SSIM comparison of denoised output with blind KSVD, BM3D, EPLL, WNNM, DnCNN, MemNet, BUIFD, Self2Self and ComplexNet. (Best values in the margin of 0.07 dB for PSNR and 0.005 for SSIM highlighted in bold.)

Dataset	Methods	PSNR						SSIM					
		AWGN level						AWGN level					
		10	20	30	40	50	60	10	20	30	40	50	60
BSD68	Proposed	33.46	**30.02**	**28.20**	**26.99**	**26.12**	**25.36**	**0.921**	**0.850**	**0.793**	**0.746**	**0.709**	**0.686**
	ComplexNet [54]	33.35	29.72	27.91	26.62	25.95	24.94	0.918	0.844	**0.788**	0.736	0.698	0.665
	Self2Self [53]	32.24	29.55	27.43	26.01	25.48	24.89	0.879	0.835	0.717	0.702	0.671	0.639
	BUIFD [52]	**33.58**	29.91	27.80	26.3	25.06	23.97	**0.926**	**0.852**	0.787	0.731	0.68	0.637
	MemNet [12]	33.33	29.59	27.32	25.63	24.35	23.34	**0.926**	**0.848**	0.769	0.701	0.646	0.606
	DnCNN [7]	**33.61**	29.65	27.17	25.31	23.75	22.29	**0.926**	0.838	0.754	0.682	0.616	0.546
	WNNM [18]	27.83	28.35	27.06	21.89	18.22	15.83	0.750	0.779	0.743	0.468	0.306	0.217
	EPLL [16]	29.51	29.14	26.07	20.82	17.54	15.70	0.798	0.808	0.707	0.430	0.289	0.215
	BM3D [15]	29.18	28.69	27.35	22.44	18.19	15.73	0.802	0.798	0.763	0.495	0.316	0.220
	KSVD [14]	27.10	27.71	26.56	21.48	18.12	15.85	0.713	0.750	0.715	0.444	0.297	0.212
Set5	Proposed	**35.18**	**32.12**	**30.38**	**29.16**	**28.19**	**27.22**	**0.927**	**0.886**	**0.854**	**0.827**	**0.804**	**0.774**
	ComplexNet [54]	35.05	31.88	29.61	28.51	27.61	26.82	0.926	0.879	0.835	0.81	0.785	0.751
	Self2Self [53]	33.24	31.35	28.11	26.59	25.11	23.27	0.919	0.863	0.818	0.799	0.751	0.711
	BUIFD [52]	35.08	31.56	29.18	27.28	25.68	24.17	**0.928**	0.871	0.815	0.765	0.719	0.672
	MemNet [12]	34.67	30.95	28.00	25.79	23.99	22.62	0.921	0.857	0.773	0.707	0.652	0.614
	DnCNN [7]	35.09	30.83	27.84	25.38	23.47	21.85	**0.927**	0.839	0.758	0.683	0.614	0.545
	WNNM [18]	30.16	30.50	28.39	22.51	18.69	16.29	0.840	0.839	0.765	0.476	0.305	0.216
	EPLL [16]	32.11	31.04	27.22	21.39	17.93	16.04	0.865	0.849	0.728	0.434	0.279	0.204
	BM3D [15]	32.19	30.98	28.72	23.12	18.73	16.24	0.882	0.856	0.791	0.509	0.324	0.222
	KSVD [14]	29.03	29.41	27.71	22.1	18.58	16.29	0.804	0.807	0.738	0.452	0.292	0.207
Set14	Proposed	**33.79**	**30.71**	**28.94**	**27.70**	**26.73**	**25.83**	**0.911**	**0.855**	**0.808**	**0.766**	**0.730**	**0.703**
	ComplexNet [54]	32.92	30.12	27.88	26.6	25.35	24.74	0.895	0.822	0.783	0.747	0.7	0.678
	Self2Self [53]	31.75	29.81	27.12	26.21	24.75	23.89	0.892	0.801	0.772	0.729	0.679	0.624
	BUIFD [52]	33.73	30.34	28.18	26.55	25.23	23.98	**0.914**	**0.852**	0.795	0.742	0.694	0.650
	MemNet [12]	33.45	29.91	27.43	25.56	24.17	23.06	**0.912**	0.842	0.767	0.701	0.649	0.609
	DnCNN [7]	**33.81**	29.98	27.39	25.40	23.66	22.09	**0.914**	0.832	0.757	0.688	0.625	0.553
	WNNM [18]	28.89	29.24	27.49	22.00	18.28	15.92	0.777	0.796	0.748	0.476	0.316	0.227
	EPLL [16]	30.45	29.70	26.28	20.89	17.58	15.76	0.814	0.815	0.707	0.435	0.298	0.221
	BM3D [15]	30.58	29.68	27.85	22.55	18.24	15.82	0.832	0.818	0.772	0.502	0.326	0.231
	KSVD [14]	27.73	28.19	26.74	21.50	18.12	15.90	0.729	0.755	0.711	0.447	0.304	0.219
SunHays80	Proposed	**34.92**	**31.60**	**29.84**	**28.65**	**27.75**	**26.92**	**0.932**	**0.873**	**0.825**	**0.785**	**0.752**	**0.727**
	ComplexNet [54]	34.45	31.47	29.65	28.18	27.35	26.28	0.925	0.871	0.818	0.778	0.738	0.702
	Self2Self [53]	33.81	30.97	29.11	27.71	26.92	25.81	0.885	0.853	0.805	0.739	0.719	0.688
	BUIFD [52]	**34.99**	31.44	29.36	27.77	26.41	25.14	**0.935**	**0.871**	0.814	0.763	0.716	0.674
	MemNet [12]	34.65	31.07	28.74	26.88	25.39	24.17	**0.932**	0.864	0.792	0.726	0.670	0.629
	DnCNN [7]	**34.94**	31.08	28.48	26.24	24.33	22.55	**0.933**	0.853	0.771	0.689	0.617	0.535
	WNNM [18]	29.89	30.30	28.56	22.43	18.51	16.06	0.795	0.810	0.750	0.416	0.246	0.164
	EPLL [16]	31.09	30.76	27.16	21.12	17.68	15.79	0.826	0.833	0.710	0.380	0.236	0.167
	BM3D [15]	31.27	30.63	28.93	23.11	18.44	15.89	0.848	0.837	0.787	0.465	0.268	0.175
	KSVD [14]	28.80	29.22	27.21	21.45	18.05	15.82	0.767	0.778	0.681	0.374	0.233	0.161
Manga109	Proposed	**36.36**	**33.30**	**31.40**	**29.99**	**28.81**	**27.60**	**0.952**	**0.925**	**0.903**	**0.883**	**0.864**	**0.832**
	ComplexNet [54]	35.02	31.15	28.35	27.12	26.42	25.17	0.938	0.889	0.852	0.826	0.791	0.763
	Self2Self [53]	34.21	30.66	27.81	26.02	25.52	24.45	0.895	0.877	0.825	0.735	0.746	0.738
	BUIFD [52]	35.88	31.86	29.09	26.92	25.13	23.58	**0.947**	0.907	0.864	0.820	0.777	0.731
	MemNet [12]	34.88	30.58	27.34	24.92	23.11	21.79	0.940	0.867	0.777	0.699	0.641	0.603
	DnCNN [7]	35.57	30.50	26.78	23.98	21.82	20.03	0.936	0.831	0.725	0.638	0.569	0.493
	WNNM [18]	31.58	31.31	28.80	22.83	18.93	16.48	0.870	0.872	0.803	0.511	0.335	0.234
	EPLL [16]	33.31	31.29	27.22	21.73	18.34	16.31	0.915	0.905	0.795	0.531	0.365	0.252
	BM3D [15]	33.45	31.52	28.80	23.54	19.20	16.63	0.924	0.910	0.858	0.607	0.428	0.296
	KSVD [14]	29.91	29.69	27.08	22.02	18.70	16.42	0.871	0.868	0.763	0.519	0.355	0.245

TABLE 2. FOM comparison of denoised output with blind BM3D, WNNM, DnCNN, MemNet, BUIFD, Self2Self and ComplexNet. (Best values in the margin of 0.005 are highlighted in bold.)

Dataset	Methods	FOM					
		AWGN level					
		10	20	30	40	50	60
BSD68	Proposed	**0.8333**	**0.7587**	**0.7372**	**0.7297**	**0.7237**	**0.7404**
	ComplexNet [54]	0.8224	0.7521	0.7301	0.7244	0.723	0.717
	Self2Self [53]	0.7309	0.7232	0.7239	0.7164	0.7007	0.6819
	BUIFD [52]	0.8002	0.7225	0.6948	0.6815	0.668	0.7201
	MemNet [12]	0.8034	0.7491	0.7268	0.722	0.7202	0.7191
	DnCNN [7]	0.8066	0.7192	0.689	0.684	0.6803	0.7393
	WNNM [18]	0.5129	0.4728	0.458	0.4514	0.4473	0.4393
	BM3D [15]	0.7051	0.58	0.5308	0.5268	0.5533	0.6029
Set5	Proposed	0.7717	**0.7339**	**0.7223**	**0.7201**	**0.7135**	**0.7294**
	ComplexNet [54]	**0.8003**	0.7269	0.7153	0.7131	0.7104	0.7225
	Self2Self [53]	0.7301	0.7088	0.7142	0.7149	0.7032	0.7071
	BUIFD [52]	0.6767	0.6523	0.6804	0.7003	0.7013	0.7205
	MemNet [12]	0.766	0.716	0.6723	0.6672	0.6678	0.6681
	DnCNN [7]	0.7299	0.7083	0.7137	0.7147	0.7054	0.7106
	WNNM [18]	0.5027	0.4401	0.4159	0.4248	0.442	0.4761
	BM3D [15]	0.6861	0.599	0.5774	0.5863	0.6203	0.6851
Set14	Proposed	**0.7772**	**0.7756**	**0.752**	**0.7478**	**0.7447**	**0.7588**
	ComplexNet [54]	0.7758	0.7349	0.7135	0.7218	0.7249	0.7497
	Self2Self [53]	0.7274	0.6601	0.6686	0.6892	0.7122	0.7537
	BUIFD [52]	0.6986	0.6568	0.6509	0.6673	0.6664	0.7206
	MemNet [12]	0.7769	0.7735	0.7476	0.7424	0.7405	0.7391
	DnCNN [7]	0.7163	0.6594	0.6793	0.7034	0.7094	0.7525
	WNNM [18]	0.4926	0.4483	0.4329	0.4228	0.4203	0.4227
	BM3D [15]	0.7044	0.6164	0.5834	0.5801	0.5965	0.6271
SunHays80	Proposed	**0.829**	**0.7287**	**0.6836**	**0.6725**	**0.6662**	**0.7199**
	ComplexNet [54]	0.8142	0.7035	0.6701	0.482	0.6632	0.7003
	Self2Self [53]	0.7738	0.7057	0.6701	0.6669	0.6621	0.6837
	BUIFD [52]	0.7583	0.6755	0.6581	0.6646	0.6603	0.6956
	MemNet [12]	0.7788	0.6655	0.6489	0.6473	0.6468	0.6461
	DnCNN [7]	0.7818	0.7022	0.668	0.6648	0.6606	0.6795
	WNNM [18]	0.5374	0.4926	0.4788	0.4611	0.4537	0.4425
	BM3D [15]	0.6556	0.5252	0.4776	0.4799	0.5207	0.5784
Manga109	Proposed	**0.8416**	**0.7839**	**0.7506**	**0.7125**	**0.6769**	**0.7393**
	ComplexNet [54]	0.8291	0.7551	0.7312	0.7016	0.6742	0.7172
	Self2Self [53]	0.7174	0.6489	0.6351	0.6304	0.6241	0.6182
	BUIFD [52]	0.7256	0.6789	0.6596	0.6494	0.6381	0.6384
	MemNet [12]	0.7954	0.6053	0.5276	0.5207	0.521	0.5223
	DnCNN [7]	0.7391	0.6454	0.6281	0.6242	0.6199	0.6163
	WNNM [18]	0.5614	0.5255	0.4967	0.4816	0.4683	0.4591
	BM3D [15]	0.7593	0.7069	0.6883	0.6941	0.6719	0.6762

TABLE 3. GPU run time (in seconds) of different methods on images of size 256×256 and 512×512 with noise level 25. The experiments are conducted on a system having 11GB NVIDIA GeForce GTX 1080 Ti GPU.(Best and second best value is denoted by red and blue respectively).

Methods	BM3D [15]	DnCNN [7]	MemNet [12]	BUIFD [52]	Self2Self [53]	ComplexNet [54]	GBGNRNet (Proposed)
256 x 256	2.37	0.17	0.13	0.22	5589	0.77	0.12
512 x 512	8.97	0.19	0.30	0.26	16963	2.57	0.26

TABLE 4. Max GPU memory (in GB) of different methods on images of size 256×256 and 512×512 with noise level 50. The experiments are conducted on 11GB NVIDIA GeForce GTX 1080 Ti GPU.(Best and second best value is denoted by red and blue respectively).

Methods	DnCNN [7]	RNAN [62]	RNAN* [62]	DRUNet [63]	GBGNRNet (Proposed)
256 x 256	0.0339	0.3525	2.9373	0.2143	0.0152
512 x 512	0.1284	0.4240	3.2826	0.4911	0.0429

TABLE 5. Number of parameters and depth of network comparison.(Best and second best value is denoted by red and blue respectively).

Methods	RED [64]	DnCNN [7]	DIDN [65]	RDN [66]	GBGNRNet (Proposed)
Parameters	4.13M	558 K	165M	22M	1M
Depth	30	20	111	133	23

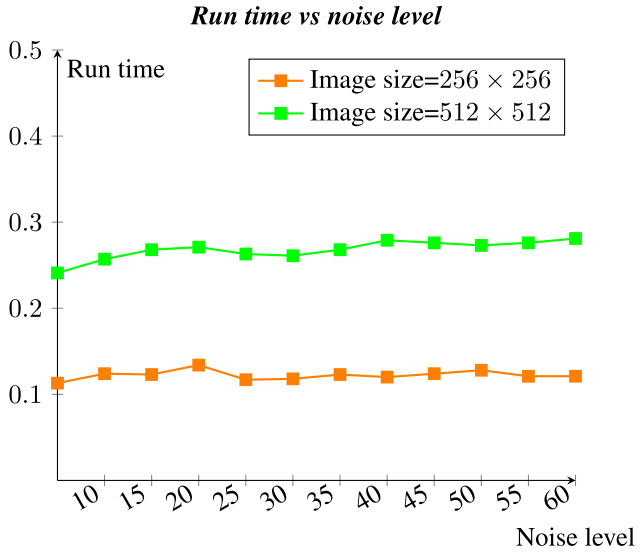


FIGURE 12. GPU Run time (in seconds) comparison of proposed method at different noise levels. *x* axis denotes the different noise levels and *y* axis denotes run time in seconds.

network. It can be clearly seen in the ablation study table that the impact of feature denoising block is highest followed by the MSFE block. After considering the scenario of inclusion and exclusion of different blocks from the proposed network, it can be deduced that GBGNRNet with all the blocks included gives the best value of PSNR as well as SSIM. The visual comparison of noise reduction results

TABLE 6. Ablation study on the BSD68 benchmark dataset.(Best values are highlighted in bold.)

Method	PSNR	SSIM
GBGNRNet	**28.36**	**0.784**
GBGNRNet without MSFE block	28.26	0.779
GBGNRNet without Feature denoising block	27.24	0.734

of ablation study is shown in Fig. 11. From this figure, the effect of inclusion and exclusion of different blocks can be observed clearly.

V. CONCLUSION

This paper presents a gradient and multi-scale feature information based blind Gaussian denoiser using feature denoising. Here two modules are used to extract residual noise from AWGN corrupted images which is then subtracted from the input noisy image to get final denoised image. The first module is used to generate an intermediate image whose gradient information is used together with multi-scale feature information to generate the final residual image. The proposed GBGNRNet method performs blind denoising of image in the sense that it does not require any prior information about the noise. The network is designed such that it learns the noise patterns and produce residual noise as output. Experimental results on three synthetic and two real image databases validate the superiority of the proposed network over the state-of-the-art blind AWGN denoising techniques.

We also compared the performance of the proposed network, with recent CNN based denoising network, in terms of GPU run time, maximum GPU memory consumption, number of parameters and depth of network to show its superiority over these recent techniques. The limitation of the proposed work is in the number of parameters used by the network. Currently the network takes around 1M parameters for denoising due to which it takes little bit more time to denoise than DnCNN. In future studies, we are planning to reduce the parameters to one fourth of the current number of parameters, so that we can use this blind denoising technique for real time video denoising as well. For reducing the number of parameters, we plan to use 32 filters at each convolutional layer instead of the current 64 filters.

REFERENCES

[1] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Netw.*, vol. 131, pp. 251–275, Nov. 2020.

[2] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Vis. Comput. Ind., Biomed., Art*, vol. 2, no. 1, pp. 1–12, Dec. 2019.

[3] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 11, pp. 3365–3385, Nov. 2020.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.

[5] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21, 2008, pp. 769–776.

[6] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 341–349.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[8] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[11] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.

[12] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4539–4547.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[16] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

[17] G. Chen, F. Zhu, and P. A. Heng, "An efficient statistical method for image noise level estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 477–485.

[18] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[19] A. Khmag, A. R. Ramli, S. J. B. Hashim, and S. A. R. Al-Haddad, "Additive noise reduction in natural images using second-generation wavelet transform hidden Markov models," *IEEJ Trans. Electr. Electron. Eng.*, vol. 11, no. 3, pp. 339–347, May 2016.

[20] A. Khmag, S. A. R. Al Haddad, R. A. Ramlee, N. Kamarudin, and F. L. Malallah, "Natural image noise removal using nonlocal means and hidden Markov models in transform domain," *Vis. Comput.*, vol. 34, no. 12, pp. 1661–1675, Dec. 2018.

[21] D. M. Vo, T. P. Le, D. M. Nguyen, and S.-W. Lee, "BoostNet: A boosted convolutional neural network for image blind denoising," *IEEE Access*, vol. 9, pp. 115145–115164, 2021.

[22] S.-F. Wang, W.-K. Yu, and Y.-X. Li, "Multi-wavelet residual dense convolutional neural network for image denoising," *IEEE Access*, vol. 8, pp. 214413–214424, 2020.

[23] J. Gurrola-Ramos, O. Dalmau, and T. E. Alarcon, "A residual dense U-Net neural network for image denoising," *IEEE Access*, vol. 9, pp. 31742–31754, 2021.

[24] J. Zhang, Y. Zhu, W. Li, W. Fu, and L. Cao, "DRNet: A deep neural network with multi-layer residual blocks improves image denoising," *IEEE Access*, vol. 9, pp. 79936–79946, 2021.

[25] S. Li, Y. Chen, R. Jiang, and X. Tian, "Image denoising via multi-scale gated fusion network," *IEEE Access*, vol. 7, pp. 49392–49402, 2019.

[26] Y. Du, G. Han, Y. Tan, C. Xiao, and S. He, "Blind image denoising via dynamic dual learning," *IEEE Trans. Multimedia*, vol. 23, pp. 2139–2152, 2021.

[27] J. Ma, C. Peng, X. Tian, and J. Jiang, "DBDnet: A deep boosting strategy for image denoising," *IEEE Trans. Multimedia*, early access, Jul. 1, 2021, doi: 10.1109/TMM.2021.3094058.

[28] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug- and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 14, 2021, doi: 10.1109/TPAMI.2021.3088914.

[29] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 1833–1844.

[30] X. Ma, X. Lin, M. E. Helou, and S. Susstrunk, "Deep Gaussian denoiser epistemic uncertainty and decoupled dual-attention fusion," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 1629–1633.

[31] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[32] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Class-aware fully convolutional Gaussian and Poisson denoising," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5707–5722, Nov. 2018.

[33] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3262–3271.

[34] K. Uchida, M. Tanaka, and M. Okutomi, "Non-blind image restoration based on convolutional neural network," in *Proc. IEEE 7th Global Conf. Consum. Electron. (GCCE)*, Oct. 2018, pp. 40–44.

[35] Y. Kim, J. W. Soh, and N. I. Cho, "Adaptively tuning a convolutional neural network by gate process for image denoising," *IEEE Access*, vol. 7, pp. 63447–63456, 2019.

[36] R. G. Pires, D. F. Santos, C. F. Santos, M. C. Santana, and J. P. Papa, "Image denoising using attention-residual convolutional neural networks," in *Proc. 33rd SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Nov. 2020, pp. 101–107.

[37] C. Godard, K. Matzen, and M. Uyttendaele, "Deep burst denoising," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2018, pp. 538–554.

[38] W. Bae, J. Yoo, and J. Chul Ye, "Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 145–153.

[39] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, "Deep shrinkage convolutional neural network for adaptive noise reduction," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 224–228, Feb. 2018.

[40] F. Zhang, N. Cai, J. Wu, G. Cen, H. Wang, and X. Chen, "Image denoising method based on a deep convolution neural network," *IET Image Process.*, vol. 12, no. 4, pp. 485–493, Apr. 2018.

[41] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[42] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, and N. Luo, "Enhanced CNN for image denoising," in *Proc. CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 17–23, Mar. 2018.
[43] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.
[44] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Netw.*, vol. 121, pp. 461–473, Jan. 2020.
[45] S. Yu, B. Park, and J. Jeong, "Deep iterative down-up CNN for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 1–9.
[46] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.
[47] B. Park, S. Yu, and J. Jeong, "Densely connected hierarchical network for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 1–10.
[48] P. Liu, H. Zhang, W. Lian, and W. Zuo, "Multi-level wavelet convolutional neural networks," *IEEE Access*, vol. 7, pp. 74973–74985, 2019.
[49] Y. Zhong, L. Liu, D. Zhao, and H. Li, "A generative adversarial network for image denoising," *Multimedia Tools Appl.*, vol. 79, nos. 23–24, pp. 16517–16529, Jun. 2020.
[50] G. Li, X. Xu, M. Zhang, and Q. Liu, "Densely connected network for impulse noise removal," *Pattern Anal. Appl.*, vol. 23, no. 3, pp. 1–13, 2020.
[51] C. Tian, Y. Xu, Z. Li, W. Zuo, and H. Liu, "Attention-guided CNN for image denoising," *Neural Netw.*, vol. 124, pp. 117–129, Apr. 2020.
[52] M. El Helou and S. Susstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," *IEEE Trans. Image Process.*, vol. 29, pp. 4885–4897, 2020.
[53] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2Self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1890–1898.
[54] Y. Quan, Y. Chen, Y. Shao, H. Teng, Y. Xu, and H. Ji, "Image denoising using complex-valued deep CNN," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107639.
[55] H. Shen, C. Zhou, J. Li, and Q. Yuan, "SAR image despeckling employing a recursive deep CNN prior," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 273–286, Jan. 2021.
[56] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 501–509.
[57] Y. Liu, S. Anwar, L. Zheng, and Q. Tian, "GradNet image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 508–509.
[58] F. G. Irwin, "An isotropic 3×3 image gradient operator," Presentation Stanford AI Project, Stanford, CA, USA, Tech. Rep. 02, 1968.
[59] J. Zhang, L. Sang, Z. Wan, Y. Wang, and Y. Li, "Deep convolutional neural network based on multi-scale feature extraction for image denoising," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2020, pp. 213–216.
[60] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1664–1673.
[61] 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 IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.
[62] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," 2019, *arXiv:1903.10082*.
[63] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug- and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 14, 2021, doi: 10.1109/TPAMI.2021.3088914.
[64] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2802–2810.
[65] S. Yu, B. Park, and J. Jeong, "Deep iterative down-up cnn for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 1–9.
[66] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2480–2495, Jul. 2020.
[67] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

RAMESH KUMAR THAKUR (Member, IEEE) received the B.Tech. degree in information technology from the University of Kalyani, West Bengal, India, in 2014, and the M.Tech. degree in high performance computing from the National Institute of Technology Durgapur, West Bengal, in 2016. He is currently pursuing the Ph.D. degree in computer science and engineering with the Indian Institute of Technology Patna, Bihar, India.

From 2016 to 2017, he was an Assistant Professor with the Computer Science and Engineering Department, Smt. S. R. Patel Engineering College Unjha, Gujarat, India. Since 2017, he has been a Research Assistant with the Indian Institute of Technology Patna. His research interests include image processing, deep learning, visual cryptography, and visual steganography.

Mr. Thakur's awards and honors include the Child Scientist Award by the National Science Congress, in 2006, the District Topper Award in Intermediate Examination, in 2010, the Class First and University Topper Award for his B.Tech. degree by the University of Kalyani, in 2014, and the Gold Medalist Award for his M.Tech. degree from NIT Durgapur, in 2016.

SUMAN KUMAR MAJI (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, India, in 2006, the Postgraduate degree in telecommunication networks from the Indian Institute of Technology Kharagpur, India, in 2008, and the Ph.D. degree in computer science from INRIA Bordeaux, France, in 2013.

From 2014 to 2015, he worked as a Research Engineer with the Institute of Hematology, University Paris 7, and INSERM. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, India. His research interests include medical imaging, bioinformatics, machine learning, and image processing.

Dr. Maji has authored several conferences and journal papers and was a recipient of various research fellowships and awards such as the European CORDIS Doctoral Fellowship, in 2010, the Region Aquitaine OPTAD Research Fellowship, in 2010, the FRM Research Fellowship, in 2014, and the SERB Early Career Research Award from DST, Govt of India, in 2017.