IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2022, accepted April 21, 2022, date of publication April 25, 2022, date of current version May 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3170490

Hybrid Deep Learning Framework for Reduction of Mixed Noise via Low Rank Noise Estimation

DAI-GYOUNG KIM[1], YASIR ALI[2], MUHAMMAD ASIF FAROOQ[3], ASIF MUSHTAQ[4], MUHAMMAD AHMAD ABDUL REHMAN[2], AND ZAHID HUSSAIN SHAMSI[2]

[1]Department of Applied Mathematics, Hanyang University (ERICA), Ansan 15588, South Korea
[2]Department of Mathematics, University of the Punjab, Lahore 54590, Pakistan
[3]School of Natural Sciences, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[4]Fakultet for Lærerutd., Kunst Og Kultur, 8049 Bodø, Norway

Corresponding author: Asif Mushtaq (asif.mushtaq@nord.no)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government [Ministry of Science and ICT (MSIT)], Artificial Intelligence Convergence Research Center, Hanyang University (ERICA), under Grant 2020-0-01343.

ABSTRACT In this paper, an innovative hybridized deep learning framework (EN-CNN) is presented for image noise reduction where the noise originates from heterogeneous sources. More specifically, EN-CNN is applied to the benchmark natural images affected by a mixture of additive white gaussian noise (AWGN) and impulsive noise (IN). Reduction of mixed noise (AWGN and IN) is relatively more involved as compared to removing simply one type of noise. In fact, mitigating the impact of a mixture of multiple noise types becomes exceedingly challenging due to simultaneous presence of different noise statistics. Although, various effective deep learning approaches and the classical state-of-the-art approaches like WNNM have been used to suppress AWGN noise only, the same techniques are not suitable in case of mixed noise. In this context, EN-CNN can not only infer changed noise statistics but can also effectively eliminate residual noise. Firstly, EN-CNN employs the classical method of neighborhood filtering followed by non-local low rank estimation to respectively reduce IN noise and estimate the residual noise characteristics after reducing IN noise. As a result of this step, we obtain a pre-processed image with residual noise statistics. Secondly, convolutional neural network (CNN) is applied to the pre-processed image based on the noise statistics inferred in the first step. This two pronged strategy, in conjunction with the deep learning mechanism, effectively handles the mixed noise suppression. As a result, the suggested framework yields promising results as compared to various state-of-the-art approaches.

INDEX TERMS Image denoising, deep learning, convolution neural network, low rank estimation, impulsive noise, Gaussian noise, mixed noise.

I. INTRODUCTION

In the presence of noise, image restoration is a major challenge in the disciplines of computer vision and image processing. Image noise is inevitable during capture, transmission, and compression resulting in image information loss [3]. Despite the fact that image restoration, particularly image denoising, has been extensively investigated, image denoising continues to be a significant research field since it is a test for various mathematically rigorous concepts such as compressed sensing, sparse coding, dictionary learning, harmonic analysis etc. These concepts are well-known for their applications to ill-posed inverse problems (non-unique optimal solutions). Furthermore, the denoising problem becomes more intriguing when the noise model consists of noise distributions from heterogeneous sources. For instance, mixed noise, which is a combination of AWGN and IN, arises due to different noise sources during the image acquisition process.

The AWGN model, which employs a Gaussian distribution, is the most frequently used noise model during natural image acquisition process. IN model refers to discrete type of noise arising during image communication. It's a black or white dot created at random, which could be white pixels in the dark or black pixels there in light (or both). Impulsive noise can be caused by various factors such as severe interference with the image signal, an analog-to-digital

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

converter, or a minor transmission error [4]. There are two types of IN, namely, salt-pepper impulsive noise (SPIN) and random valued impulsive noise (RVIN). Traditional linear filters, such as mean filtering, successfully eliminate noise while distorting essential image characteristics such as edges and textures. Non-linear filtering techniques have been offered as a solution to this problem. As far as IN, alone, is concerned, nonlinear filters such as median filters [5] have been frequently employed. However, by employing this basic filtering approach, the local features of the image will be lost. Certain non-linear filters work only on affected pixels in an image requiring the identification of noisy and noiseless pixels. Representative of such filters are adaptive median filter (AMF) [6], [7] and central weighted median filter (CWMF) [8]. In case of AWGN elimination, the wavelet transform [9], non-local means [7], [10], sparse representation based on an over-complete dictionary [11] and multi-scale geometric analysis transformation [12] are all among popular classical techniques. Another quite intriguing approach suppresses AWGN noise by utilizing hidden Markov Model (HMM) in conjunction with wavelet transform or non-local means filtering [13], [14]. The performance of these approaches may further be extended to blind image denoising using patch based noise estimation strategies entailed in [15], [16]. In addition, to these classical approaches, recent advancements in machine learning methodologies have embarked on image denoising problem [17]–[20].

Addressing either AWGN or IN, alone, is relatively less involved provided that the noise statistics are sampled from a single noise model distribution. However, in pragmatic situations, an image may be affected by a mixture of several noise sources, such as AWGN+SPIN (A+S) or AWGN+RVIN (A+R). In such scenario, the problem of restoring the underlying image data seems more intimidating. There have been several techniques reported for eliminating mixed noise in which both noise statistics are handled concurrently [21]. Despite the pioneering role of these approaches, their effectiveness is restricted since the simultaneous treatment of multiple forms of mixed noise distributions may not adequately capture noise complexity.

Recently, deep learning (DL) techniques, such as convolutional neural network (CNN) with deep architecture [22], have been envisaged for image denoising task. For instance, residual learning based deep convolutional neural network (DnCNN) is one of the earliest attempts in this direction [1]. This mechanism has yielded outstanding results as compared to those obtained by the classical benchmark algorithms in case of AWGN alone. However, the residual learning is not as effective in case of mixed noise due to complex nature of noise statistics. Moreover, DnCNN requires the prior information of AWGN noise for employing trained CNN to remove the noise. Furthermore, few DL-based mechanism tackle the intimidating challenges of mixed noise reduction [23] but the efficiency of such models always hinges upon a single image's low-level information (pixel intensities and smoothness) or middle-level representation (sparse codes and dictionary learning). Thus complex prior design is pre-requisite for these models and consequently incurs a high computational cost.

Focusing on these key limitations, An effective hybridized deep learning framework (EN-CNN) is proposed for mixed noise reduction. The key contribution of EN-CNN is to enable the convolution neural network (CNN) to handle the mixed noise by avoiding direct application of CNN. For this purpose, EN-CNN attempts to identify the locations in the given image which are affected by impulsive noise (IN). This identification process enables the proposed framework to differentiate IN and AWGN noise present in the given image. Once, the IN noise is identified and reduced, non-local self similarity (NSS) prior can easily be invoked to smooth out the noisy details which are not affected by IN noise. These key steps contribute non-trivially for the application of appropriately trained deep learning mechanism. That is, without the simplification and estimation of the mixed noise, the deep learning approaches yield very limited denoising performance. Finally, a convolution neural network (CNN) based on residual learning is trained according to the noise statistics obtained as a result of the above mentioned key steps. Thus, the main contribution of the proposed mechanism is to enhance the limited performance of direct application of CNN in case of mixed noise.

The outline of this paper is as follows: In Section II, we define the mixed noise problem followed by the rudimentary concepts regarding the median filtering and convolutional neural network. Section III entails the description of the key ingredients of the proposed framework. Section IV validates the proposed mechanism by conducting comprehensive experiments on the bench mark test images and then by comparing the results with different state of the art techniques. Finally, in Section V, our findings and limitations regarding the performance and features of the proposed framework are concluded.

II. PRELIMINARIES

A. FORMULATION OF PROBLEM

Let $\mathbf{v}$ be a original desired noise free image of size $p \times q$ and $\mathbf{z}$ be the noisy degraded version of $\mathbf{v}$ defined as follows:

$$\mathbf{z} = g(\mathbf{v}), \tag{1}$$

where $g(.)$ is the degradation function. Let a noisy pixel be represented as $z(m, n)$. We consider two forms of mixed noise: (i) Mixed noise (A+S), (ii) Mixed noise (A+R). If the image is affected by AWGN, then a noisy pixel is defined as:

$$z(m, n) = v(m, n) + \hat{z}(m, n), \tag{2}$$

where $\hat{z}(n_p, n_q)$ is a drawn form of zero-mean Gaussian distribution with standard deviation σ. Let u_{max} and u_{min} be the maximum and minimum values of an image pixel within the given dynamic range. Then, an image is affected by SPIN if $z(m, n)$ is either u_{max} or u_{min} with equal probability

$S/2 (S \leq 1)$. The noisy observation for each pixel of mixed noise (A+S) may be expressed using these definitions as:

$$z(m,n) = \begin{cases} u_{min}, & \text{probability } S/2; \\ u_{max}, & \text{probability } S/2; \\ v(m,n) + \hat{z}(m,n), & \text{probability } (1-S). \end{cases} \quad (3)$$

Similarly, let an image affected with RVIN when $z(m,n)$ achieves a random value $u(m,n)$ with probability $S (S \leq 1)$. Within the dynamic range, $u(m,n)$ is distributed uniformly $[u_{min}, u_{max}]$. The noisy observation for each pixel of mixed noise (A+R) may be expressed using these definition as:

$$z(m,n) = \begin{cases} u(m,n), & \text{probability } S; \\ v(m,n) + \hat{z}(m,n), & \text{probability } (1-S). \end{cases} \quad (4)$$

B. MEDIAN FILTER AND THE ADAPTIVE MEDIAN FILTER

The median filter [24] is a nonlinear statistical signal processing technique [25] to remove the noise of impulsive nature. As apparent from it's name, the noisy pixel value is replaced by the median of the surrounding pixel values. This simple filtering approach is defined as:

$$a(m,n) = med\{b(m-c, n-d), (c,d) \in W\}, \quad (5)$$

where $a(m,n)$ and $b(m,n)$ are the input noisy image and median filtered image, respectively. W is a two dimensional window of size $i \times i$ (i is an odd number), centred at the pixel position (m,n). The median filter works well for, but has deteriorated performance in case of higher probability of impulsive noise (IN). Numerous versions of the median filtering [6], [26]–[30] have been envisaged to address this issue. Among these approaches, the adaptive median filter [6] has received a lot of attention, owing to its adaptive nature with variable mask size.

C. CONVOLUTIONAL NEURAL NETWORK (CNN)

One of the key ingredients of the proposed framework is the construction and training of a deep learning model for mixed noise reduction. Among various existing deep learning architectures, the architecture of convolutional neural network (CNN) is considered [1], [31]. The advantages of using CNN rather than using the traditional fully connected neural network are three-fold. Firstly, the hidden layers may not be fully connected. That is, the neurons in one layers are not fully connected with each neuron in the next layer. Thus, the computational cost of training the weight parameters can be reduced. Secondly, CNN can be directly applied to the image data whereas, in case of traditional deep learning approaches, the image data is first transformed into a presentable form for a neural network. Thirdly, one of the most important characteristic of CNN is the automated extraction and selection of features directly from the image data without any manual intervention. This automated extraction and selection of feature map is endowed by the convolution layers. These convolution layers comprise of filters of fixed sizes in different orientations to capture the important details like edges and texture in the image data. Owing to these vital features, CNN architecture is widely utilized in a variety of computer vision applications [32], [33], including image classification, object identification, facial expression recognition, feature extraction and voice recognition [34]–[38] etc.

Despite these illuminating features, there are certain limitations that may affect the efficacy of CNN. For instance, by increasing the number of hidden layers (deeper CNN), the training process of CNN becomes more intimidating or even intractable. Also, the deeper CNN may suffer from performance saturation because of vanishing gradients. However, these limitations can be addressed by certain mechanism such as batch normalization (BN) and dilated convolution process [39].

III. PROPOSED FRAMEWORK

Before delving into the details of the proposed framework, its key steps are summarized as follows. Firstly, the proposed framework transforms the given noisy image into a pre-processed image to identify the pixel values affected by impulsive noise. Secondly, non-local means filtering is applied to the pre-processed image in order to approximately alter the noise statistics of impulsive noise into Gaussian noise statistics. Thirdly, the noise statistics obtained in the second step are estimated via an effective low rank approach [40], [41]. Finally, the deep learning framework is applied to the transformed image based on the noise statistics estimated in the third step. The schematic diagram of the proposed framework is also provided in Fig. 1. In what follows, we provide the details of these key steps.

A. IMAGE PREPROCESSING

For a noisy image z influenced by mixture of noise (AWGN and IN), the first major step is to examine the noise statistics in order to identify the pixel values corrupted by IN. For this purpose, we resort to an effective filtering approach, namely adaptive median filter (AMF) [6], [7]. The basic principle of AMF is to determine the pixel locations affected by IN through comparison of each pixel to its neighbors (within a window of certain size). An intriguing feature of impulsive noise pixel is that it differs from the majority of its neighbors. In the local neighborhood window, such pixels are substituted by the median of their neighboring pixels. As a consequence, the filtered image denoted by x is obtained as:

$$x = AMF(z). \quad (6)$$

The pixel locations in x can now be separated into two distinct categories Γ and Γ^c based on the above mentioned filtering process. The set Γ, of pixels affected by IN, is defined as follows

$$\Gamma = \{m \in \Omega | \, x(m) - z(m) \neq 0\}, \quad (7)$$

where Ω represents the indexing set for all pixel locations in the image z. The rest of the pixels are grouped into the collection, which is defined as:

$$\Gamma^c = \Omega \setminus \Gamma. \quad (8)$$

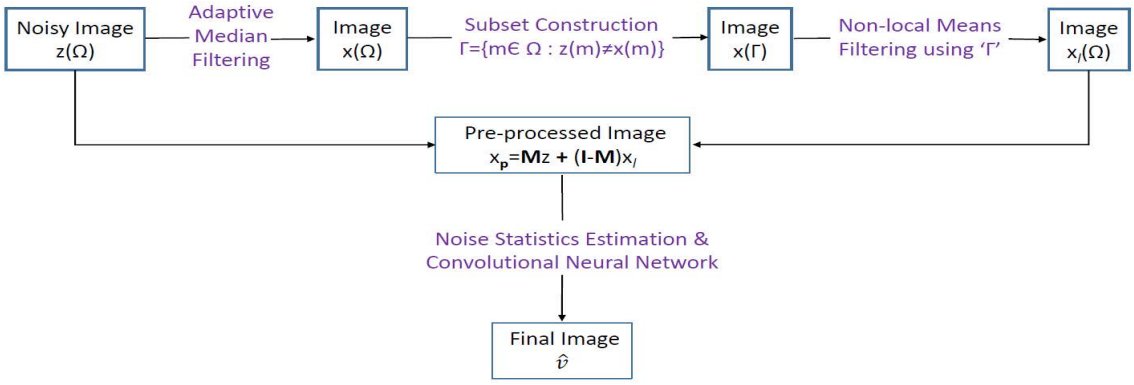
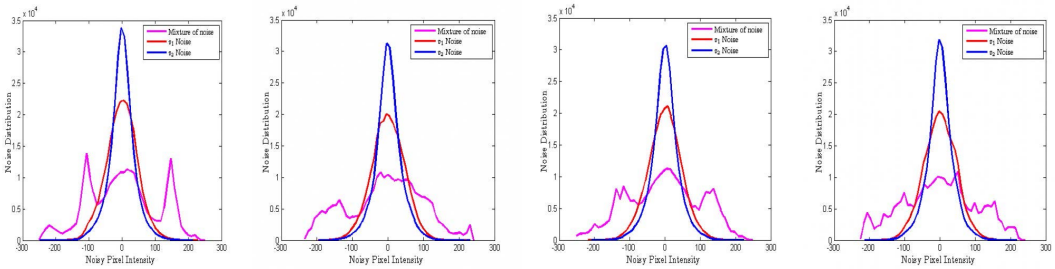


FIGURE 1. Schematic depiction of the proposed framework.



(a) (b) (c) (d)

FIGURE 2. Noise distributions visualization: The magenta colour represents the initial addition of mixed noise. The red colour represents the noise distribution v_1 after the adaptive median filtering step, while the blue colour represents the noise distribution v_2 after the non-local mean filtering step. (a) Boat (b) Hill (c) Couple (d) Lena are the test images used in the plots.

Before proceeding further, it seems reasonable to analyze the noise statistics in the filtered image x. That is, it would be enquired whether the noise statistics of the residual noise in x follows the Gaussian distribution or not. For this purpose, the residual noise is defined as $v_1 = x - v$ and its histogram (red-color curves) is depicted in Fig. 2, where v is the clean image assumed to be available for the sake of analysis. It turned out that the residual noise v_1 does not follow the Gaussian distribution. The source of non-Gaussianity in the v_1 may be attributed to the noisy pixels categorized in Γ by virtue of its construction.

Despite the fact that there are various successful methods for filtering IN, the noise cannot be completely removed. To make matters worse, the residual noise may have a negative impact on the Gaussian noise distribution in the pre-processed image. In order to lessen the adverse impact of residual noise, non-local means filtering (NLM) [10], [42] is performed on the pre-processed image, x. The motivation for applying NLM at this stage may be substantiated by the similar mechanism [43] to avoid the adverse effects of speckle type of noise. Owing to non-local self similarity (NSS) prior of NLM, the pixel values influenced by IN (categorized in Γ) can be averaged so that the residual noise, v_1, approximately follows Gaussian noise statistics. Non-local means filtering for sub-image indexed by Γ, is defined as [44]:

$$x_l^m = \frac{1}{K} \sum_{n \in w_m;\, n \in \Gamma} D(m,n) H(x_n), \tag{9}$$

where x_l^m is the averaged pixel value in $x(m)$ with patch $H(x_m)$ (centered at location $m \in \Gamma$) and w_m is the window centred at pixel position $m \in \Gamma$. $D(m,n) = e^{-\frac{\|H(x_m) - H(x_n)\|^2}{t^2}}$ denotes similarity weight of the patch centred at $x(n)$ with the reference patch centered at $z(m)$. The value of normalization factor K, is given by

$$K = \sum_{m \in \Gamma} \sum_{n \in w_m;\, n \in \Gamma} D(m,n). \tag{10}$$

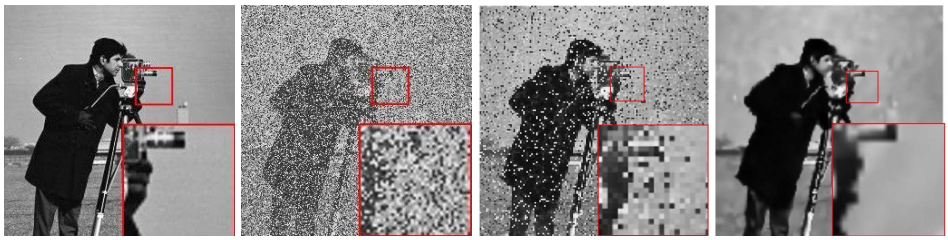
Let x_l be the resulting image obtained by NLM filtering to x. Fig. 2 shows the noise statistics, v_2, in the image x_l. It can be noticed that v_2 approximately follows a Gaussian distribution having zero mean value. However, the variance of the noise is to be determined as described in the subsequent section. Based on the defining characteristic of sets Γ and Γ^c, an indicator matrix χ is constructed as follows

$$\mathbf{M} = \begin{cases} 0, & \text{if } m \in \Gamma, \\ 1, & \text{otherwise.} \end{cases} \tag{11}$$

Finally, the first phase (pre-processed image), x_p, can be defined as a convex combination of z and x_l as follows

$$x_p = \mathbf{M}z + (\mathbf{I} - \mathbf{M})x_l, \tag{12}$$

where $\mathbf{I}$ is a matrix containing all of the entries 1. Furthermore, the Gaussian noise distributions in x_p are independent, since the image components $\mathbf{M}z$ and $(\mathbf{I} - \mathbf{M})x_l$ are mutually disjoint.



(a) (b) (c) (d)

FIGURE 3. The figure shows the importance of the proposed noise estimation strategy for the deep learning framework. The figures from left to right represent the noise free image, noisy image with mixed noise ($\sigma = 30\%$ and $S = 40\%$), residual learning based CNN without noise estimation DLM and the proposed deep learning framework with noise estimation (EN-CNN), respectively.

TABLE 1. The table depicts the limited performance of deep learning mechanism (DLM) for (A+S) without noise estimation for $S = 50\%$.

Images	$\sigma = 20$			$\sigma = 30$			$\sigma = 50$			$\sigma = 70$		
	Cai's	WESNR	DLM	Cai's	WESNR	DLM	Cai's	WESNR	DLM	Cai's	WESNR	DLM
Man	26.90	**27.48**	9.39	25.70	**25.94**	10.22	**23.30**	15.27	11.71	**20.86**	7.23	12.55
	0.8909	**0.9035**	0.5372	0.8619	**0.8830**	0.5486	**0.8070**	0.6601	0.5911	**0.7520**	0.5269	0.5961
Lena	28.43	**29.03**	9.42	26.99	**27.57**	10.28	**24.23**	15.73	11.76	**21.63**	6.67	12.74
	0.9177	**0.9300**	0.4801	0.8905	**0.9060**	0.4941	**0.8346**	0.6222	0.5458	**0.7867**	0.5200	0.5558
Couple	24.44	**26.71**	9.48	24.31	**25.10**	10.36	**22.31**	14.52	11.86	**20.57**	6.63	12.81
	0.8664	**0.9147**	0.5800	0.8317	**0.8801**	0.5920	**0.7669**	0.6519	0.6306	**0.7100**	0.5007	0.6360
Hill	27.11	**27.77**	9.38	25.94	**26.38**	10.19	**24.57**	15.99	11.63	**21.18**	7.94	12.52
	0.8886	**0.9030**	0.5290	0.8605	**0.8809**	0.5408	**0.8072**	0.6918	0.5912	**0.7589**	0.5078	0.5982
Boat	25.83	**26.95**	9.40	24.67	**24.93**	10.29	**22.46**	14.12	11.78	**20.46**	6.07	12.75
	0.8792	**0.9100**	0.5489	**0.8470**	0.7851	0.5619	**0.7851**	0.6277	0.6025	**0.7258**	0.5199	0.6068
Average	26.54	**27.58**	9.41	25.52	**25.98**	10.26	**23.37**	15.12	11.74	**20.94**	6.09	12.67
	0.8881	**0.9134**	0.5350	0.8583	**0.8670**	0.5474	**0.8001**	0.6507	0.5922	**0.7466**	0.5150	0.5985

B. NOISE ESTIMATION

It is worth noticing that the image, x_p defined in Eq. (12), is a convex combination of two noisy images. That is, the initially given noisy image, z, with Gaussian noise of known standard deviation σ_0 and the pre-processed noisy image, x_p, with Gaussian noise of unknown standard deviation denoted by $\hat{\sigma}_n$.

In fact, the employment of the classical filtering and estimation of $\hat{\sigma}_n$ are highly non-trivial in the sense that the straightforward implementation of neural network is not possible until and unless the mixed noise has been properly transformed and estimated. In order to substantiate the above arguments, Fig. 3(c) is used to show the drastic result of direct application of neural network (DLM) without prior classical filtering step. To precisely depict the limitation of DLM, the quantitative results are also shown in Table 1. It can be observed that the DLM performance is quite low as compared to the rest of the methods. Furthermore, the non-trivial impact of classical filtering and effective noise estimation strategy can further be strengthened by comparing the results of Fig. 3(c) and Fig. 3(d), wherein Fig. 3(d) is indirectly obtained by the classical mechanism followed by the deep learning framework (EN-CNN). In addition, the intermediate image and the final image are shown in Fig. 4 to emphasize the non-trivial impact of the classical filtering and noise estimation steps before applying deep learning mechanism. In what follows, the estimation of $\hat{\sigma}_n$ will be explained in detail, enabling an appropriately trained deep learning framework to effectively suppress the mixed noise and recover the underlying clean image.

In order to estimate variance of AWGN, various block-based and wavelet-based methods have been proposed [45]–[48] where the variance is estimated by comparing the noisy image to the filtered image obtained by low pass filter to the noisy image. In recent years, low rank patch-based strategies for noise estimation have been proposed [49]–[52]. It is worth noticing that the low rank approximation can also be effectively used to recover the underlying signal from the observed multi-channel data in the impulsive noise environments [52]. Inspired by these approaches, we exploited the principal component analysis (PCA) based mechanism described in [16] for noise estimation. That is, the minimal eigenvalue of the covariance matrix $\mathbf{C}$ is employed to estimate the standard deviation $\hat{\sigma}_n$ of noise v_2.

$$\mathbf{C} = \frac{1}{s}\sum_{m=1}^{s}(\nu_m - \mu_c)(\nu_m - \mu_c), \tag{13}$$

where $\mu_c = \frac{1}{s}\sum(\nu_m)$ represents the mean, s denotes the total number of patches that have been compared and ν_m denotes m^{th} eigenvalue of the low-rank covariance matrix $\mathbf{C}$. The standard deviation $\hat{\sigma}_n$, can be computed as follows:

$$\hat{\sigma}_n = \min_{1\le m\le s} \nu_m, \tag{14}$$

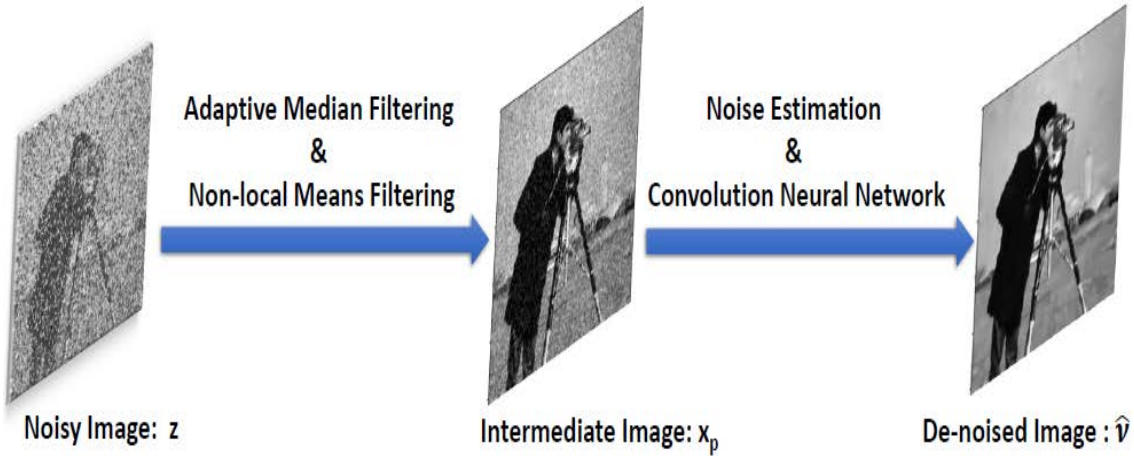


FIGURE 4. The figure shows the intermediate image x_p of the pre-processing step and the final denoised image $\hat{v}$ obtained after noise estimation and application of CNN on the intermediate image x_p.

Moreover, in the pre-processed image x_p given by Eq.12, the noise is a mixture of two independent Gaussian distributions with zero means and respective standard deviations $\hat{\sigma}$ and $\hat{\sigma}_n$. Owing to the the independence of these Gaussian distributions as discussed in previous section, the noise in x_p, is just another Gaussian distribution with zero mean and variance σ_p^2 given as:

$$\sigma_p^2 = \sigma_0^2 + \hat{\sigma}_n^2. \tag{15}$$

Now, we are in a position to employ the deep learning framework corresponding to the estimated variance of AWGN defined in Eq. 15.

C. PROPOSED DEEP LEARNING FRAMEWORK

After having obtained a pre-processed image image, x_p, and noise estimation σ_p^2 of AWGN in x_p, the problem of mitigating mixed noise has been reduced to a relatively simple problem of removing AWGN which can be mathematically defined as:

$$x_p = v + \mathcal{N}(0, \sigma_p^2), \tag{16}$$

where v is the desired noise free image to be recovered and $\mathcal{N}(0, \sigma_p^2)$ is Gaussian noise distribution provided by Eq. (15).

Since, the noise statistics now closely follows the Gaussian distribution, the convolution neural network (CNN) based on residual learning mechanism [1] is considered for training purpose. Prior to discussing the proposed CNN approach, it is important to have a brief description of CNN architecture which motivates to use CNN framework. A typical CNN inherits conventional structure of feed forward neural network such as input layers, hidden layers, activation functions and output layers. In addition, what marks the distinctive feature of CNN is the notion of convolution layer. The convolution layers enable the neural network to learn the optimal feature map for the given data without any human intervention where as in the traditional approach, the features were manually chosen and provided at the input layer of the neural network. Moreover, convolution layer, as apparent from its name, involves the process of convolving the image data (patches) with various filters in different orientations in order to obtain an appropriate feature map.

Adhering to these prominent characteristics of CNN, the use CNN is envisaged for the preprocessed image data x_p. The underlying concept of residual deep learning mechanism is to train the neural network to learn the residual mapping, $\pi(x) \approx \mathcal{N}(0, \sigma_p^2)$ which leads to the following approximation

$$\hat{v} = x_p - \pi(x), \tag{17}$$

where $\hat{v}$ denotes the learned approximation of clean image v. The corresponding penalty/loss function required to train the proposed CNN is defined by

$$\mathscr{L}(\vartheta) = \frac{1}{2N} \sum_{m=1}^{N} \| \pi\left(x_p^m; \vartheta\right) - \left(x_p^m - v^m\right) \|, \tag{18}$$

where ϑ indicates the weights (parameters) to be learnt from the training data samples x_p^m with the desired output v^m. Since, the training process is carried out in batches of randomly selected training samples, a stochastic optimization approach would be appropriate to train the neural network. Therefore, the Adaptive Moment Estimation (ADAM) strategy [53] is implemented as follows. The method employs individual adaptive learning rates for the training parameters ϑ from the respective moving averages (estimates) of first and second moments of the gradient. The process of optimization at the k^{th} iteration can be mathematically described as:

$$\eta_k = \nabla_\vartheta \mathscr{L}(\vartheta_{k-1}), \tag{19}$$

where $\nabla_\vartheta \mathscr{L}(\vartheta_{k-1})$ denotes the gradient of the loss function $\mathscr{L}(\vartheta_{k-1})$ with respect to weight parameters ϑ. Subsequently, the first and second moments μ_1 and μ_2 at the k^{th} step are

expressed as the moving average of their respective previous values and the current gradient η_k as follows

$$\mu_1^{(k)} = \tau_1 \mu_1^{(k-1)} + (1-\tau_1)\eta_k, \tag{20}$$

$$\mu_2^{(k)} = \tau_2 \mu_2^{(k-1)} + (1-\tau_2)\eta_k^2, \tag{21}$$

where τ_1 and τ_2 are the exponential decay rates to define the exponential moving averages for μ_1 and μ_2, respectively. Since the moments μ_1 and μ_2 are initially set to zero vectors, therefore the estimates obtained with Eqs. 20-21 are the biased estimators. That is, the expected values of these moments satisfy the relations $\mathbb{E}[\mu_1^k] \neq \mathbb{E}[\eta_k]$ and $\mathbb{E}[\mu_2^k] \neq \mathbb{E}[\eta_k^2]$, respectively. However, these estimates can be updated so as to achieve unbiasedness asymptotically as follows [53]. The updating process for the first moment μ_1 is being explained which can be applied to the second moment μ_2 in a similar fashion. By setting $\mu_1^{(0)} = 0$ and exploiting the iterative process of moving averages, its k^{th} estimate is expressed as:

$$\mu_1^{(k)} = (1-\tau_1)\sum_{i=0}^{k} \tau_1^{k-i}\eta_i. \tag{22}$$

Taking the expected value on both sides the above expression, we arrive at

$$\mathbb{E}(\mu_1^{(k)}) = (1-\tau_1)\sum_{i=0}^{k} \tau_1^{k-i}\mathbb{E}(\eta_i). \tag{23}$$

Replacing $\mathbb{E}(\eta_i)$ with $\mathbb{E}(\eta_k)$ in the above series yields an approximation, inducing an error term ϵ as follows

$$\mathbb{E}(\mu_1^{(k)}) = \mathbb{E}(\eta_k)\,(1-\tau_1)\sum_{i=0}^{k} \tau_1^{k-i} + \epsilon, \tag{24}$$

$$= \mathbb{E}(\eta_k)(1-\tau_1^k) + \epsilon. \tag{25}$$

This error time ϵ looks intimidating. However, this error can be reduced due to the fact that τ_1 is the exponential decay rate which assigns least weight to strayed gradient values in the past. Keeping in view this important characteristic of τ_1 and τ_2, the unbiased estimator of μ_1 and μ_2 can then be rewritten as follows

$$\hat{\mu_1}^{(k)} = \frac{\mu_1^{(k)}}{(1-\tau_1^k)}, \tag{26}$$

$$\hat{\mu_2}^{(k)} = \frac{\mu_2^{(k)}}{(1-\tau_2^k)}. \tag{27}$$

Finally, the parameters ϑ can be updated at k^{th} step as:

$$\vartheta_k = \vartheta_{k-1} - \alpha \frac{\hat{\mu_1}^k}{\sqrt{\hat{\mu_2}^k} + \epsilon}, \tag{28}$$

where α is the step size and $\epsilon = 10^{-8}$ to avoid potential division by 0.

The proposed mechanism is comprised of three main blocks as follows. The first block consists of convolution operation followed by activation function such as rectifying linear unit (ReLU) [56]. The convolution operation in this block involves 64 filters, each of size 3×3. In other words, the size of convolution layer in the first block is $1 \times 3 \times 3 \times 64$ for a gray-scale image. The second block consists of 15 convolution layers, each of which has size $64 \times 3 \times 3 \times 64$ followed by batch normalization (BN) [39] and thresholding process (ReLU).

Insertion of BN prior to the application of ReLU is critical in order to standardize the convoluted data and to avoid vanishing gradient problem. It is also important to note that the number of convolution layers in this block is set heuristically. Increasing the number of layers incurs more computational cost with a minuscule improvement. The second block is responsible for getting a reliable feature map from the training data which would be subsequently used for residual image reconstruction. Finally, the third block comprises of just one convolution layer of size $64 \times 3 \times 3 \times 1$ to reconstruct the residual image. In order to succinctly describe the key steps of the proposed framework is outlined in the Algorithm 1.

Algorithm 1 Mixed Noise Removal Using the Proposed Deep Learning Framework (EN-CNN)

Input: 400-training images, Epochs=50,
Trained CNN for AWGN; AWGN with parameter: $\boldsymbol{\sigma}_0$, IN with parameter: **S**%, RVIN with parameter: **r**% Noisy image: **z**.

1: Set the mixed noise parameters ($\boldsymbol{\sigma}_0$, **S**%, **r**%) as described in Section.IV.
STEP-I.
2: Get the median filtered image, x, using Eq. (6).
3: Detect IN pixels by using Γ and Γ^c defined in Eqs. (7-8).
4: Get the NLM filtered image, x_l, using Eq. (9).
5: Get a preprocessed image, x_p, defined by Eq. (12).
STEP-II.
6: Estimate the level of noise $\hat{\sigma_n}$ in x_l by Eq. (14).
7: By using $\hat{\sigma_n}$ to estimate the noise level σ_p defined in Eq. (15).
STEP-III.
8: Load the models of trained CNN for estimated noise σ_p.
9: Feed the processed image x_p using Eq. (16) at the CNN input.
10: Get the final denoised image v at the CNN output.

Output: Desired Image: v.

IV. EXPERIMENTS

The suggested hybrid deep learning framework (EN-CNN) is evaluated for 10 frequently used gray-scale test images with rich texture and geometric details as depicted in Fig. 5. The qualitative and quantitative evaluations were then compared with the state-of-the-art mixed noise reduction approaches, namely, Cai's [54], $l_1 - l_0$ [57] and WESNR [55].

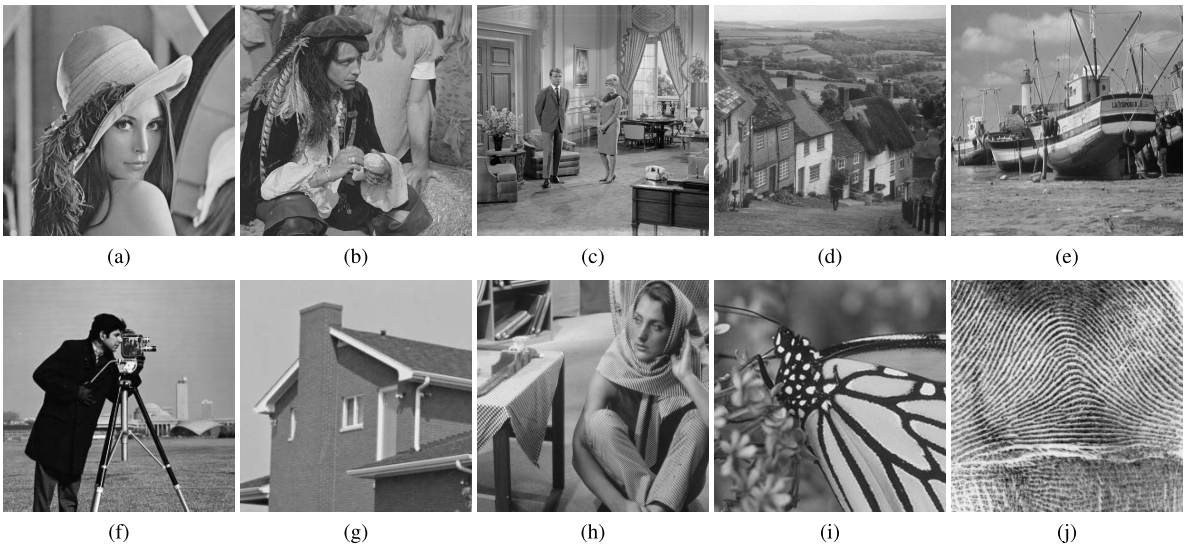


FIGURE 5. Test images for comparison.

TABLE 2. At $S = 50\%$, the proposed framework (EN-CNN) is compared to alternative approaches for mixed noise (A+S) using PSNR and FSIM.

Images	$\sigma_0 = 20$			$\sigma_0 = 30$			$\sigma_0 = 50$			$\sigma_0 = 70$		
	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN
Man	26.90	27.48	**27.63**	25.70	25.94	**25.97**	23.30	15.27	**23.95**	20.86	7.23	**22.28**
	0.8909	0.9035	**0.9174**	0.8619	0.8830	**0.8836**	0.8070	0.6601	**0.8400**	0.7520	0.5269	**0.7996**
Lena	28.43	**29.73**	29.53	26.99	27.57	**27.63**	24.23	15.73	**25.12**	21.63	6.67	**23.17**
	0.9177	0.9300	**0.9397**	0.8905	0.9060	**0.9034**	0.8346	0.6222	**0.8512**	0.7867	0.5200	**0.8018**
Couple	24.44	26.71	**26.94**	24.31	25.10	**25.32**	22.31	14.52	**23.17**	20.57	6.63	**21.48**
	0.8664	0.9147	**0.9153**	0.8317	0.8801	**0.8821**	0.7669	0.6519	**0.8232**	0.7100	0.5007	**0.7686**
Hill	27.11	27.77	**28.11**	25.94	26.38	**26.47**	24.57	15.99	**24.26**	21.18	7.94	**22.53**
	0.8886	0.9030	**0.9193**	0.8605	0.8809	**0.8912**	0.8072	0.6918	**0.8496**	0.7589	0.5078	**0.8053**
Boat	25.83	26.95	**27.24**	24.67	24.93	**25.52**	22.46	14.12	**23.50**	20.46	6.07	**21.72**
	0.8792	0.9100	**0.9186**	0.8470	0.7851	**0.8876**	0.7851	0.6277	**0.8383**	0.7258	0.5199	**0.7802**
Average	26.54	27.72	**27.89**	25.52	25.98	**26.18**	23.37	15.12	**24.00**	20.94	6.09	**22.23**
	0.8881	0.9134	**0.9220**	0.8583	0.8670	**0.8895**	0.8001	0.6507	**0.8404**	0.7466	0.5150	**0.7911**

TABLE 3. At $\sigma_0 = 20$, the proposed framework (EN-CNN) is compared to alternative approaches for mixed noise (A+S) using PSNR and FSIM.

Images	$S = 30\%$			$S = 40\%$			$S = 50\%$		
	Cai's	$l_1 - l_0$	EN-CNN	Cai's	$l_1 - l_0$	EN-CNN	Cai's	$l_1 - l_0$	EN-CNN
Man	27.83	28.45	**28.50**	27.38	27.93	**28.04**	26.82	27.32	**27.43**
	0.9161	0.9225	**0.9337**	0.9041	0.9126	**0.9267**	0.8876	0.9027	**0.9174**
Lena	29.34	30.76	**31.04**	28.96	30.01	**30.33**	28.42	29.27	**29.54**
	0.9315	0.9430	**0.9520**	0.9263	0.9395	**0.9475**	0.9181	0.9288	**0.9397**
Couple	26.81	28.08	**28.17**	26.25	**27.51**	27.49	25.57	26.63	**26.64**
	0.9128	0.9314	**0.9318**	0.8973	0.9142	**0.9247**	0.8753	0.9012	**0.9133**
Hill	27.89	28.70	**28.72**	27.50	**28.27**	28.23	26.98	27.69	**27.70**
	0.9101	0.9246	**0.9321**	0.8972	0.9133	**0.9267**	0.8792	0.9013	**0.9193**
Boat	27.09	**28.71**	28.34	26.50	27.61	**27.62**	25.77	26.76	**26.78**
	0.9128	0.9321	**0.9362**	0.8988	0.9171	**0.9286**	0.8785	0.9028	**0.9186**
Average	27.79	28.94	**28.95**	27.31	28.26	**28.34**	26.71	27.53	**27.61**
	0.9166	0.9307	**0.9321**	0.9047	0.9193	**0.9308**	0.8877	0.9073	**0.9216**

A. TRAINING AND PARAMETER SETTING

In order to perform training and validation of EN-CNN, Berkeley Segmentation Dataset and Benchmark (BSD500) dataset [58] is used, which contains natural images with variety of geometric and textural details. The noise levels for AWGN are set into the range of $[1, 75]$ and $4,000$ patches of size 50×50 are randomly cropped to train the model with batch size of 128. For testing phase, ten well-known bench mark images from SIPI image data (University of California) are considered for fair comparison with the previous approaches. The number of epochs is set as 50 since the training phase does not reflect any significant improvements

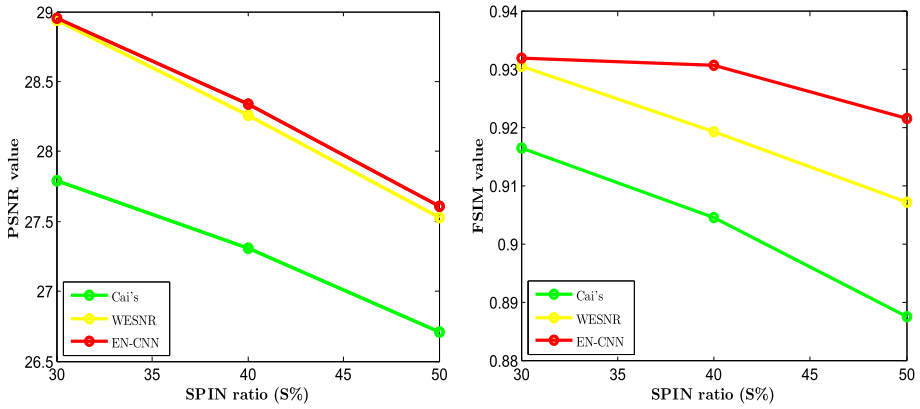


FIGURE 6. Averaging PSNR and FSIM in Table 3 are depicted graphically for $\sigma_0 = 20$.

TABLE 4. At $\sigma_0 = 10$, the proposed framework (EN-CNN) is compared to alternative approaches for mixed noise (A+R) using PSNR and FSIM.

Images	$r = 5\%$				$r = 10\%$				$r = 15\%$			
	Cai's	$l_1 - l_0$	WESNR	EN-CNN	Cai's	$l_1 - l_0$	WESNR	EN-CNN	Cai's	$l_1 - l_0$	WESNR	EN-CNN
Hill	29.94	30.17	30.96	**31.22**	29.72	29.95	30.69	**30.95**	29.48	29.70	30.46	**30.62**
	0.9540	0.9480	0.9634	**0.9684**	0.9513	0.9554	0.9612	**0.9665**	0.9481	0.9427	0.9591	**0.9633**
Couple	28.63	29.13	30.16	**30.55**	28.29	28.81	29.84	**30.13**	27.97	28.47	29.48	**29.59**
	0.9502	0.9477	0.9642	**0.9684**	0.9467	0.9440	0.9619	**0.9658**	0.9422	0.9399	0.9580	**0.9615**
Boat	28.84	29.29	30.21	**30.67**	28.51	28.91	29.80	**30.29**	28.20	28.61	29.52	**29.70**
	0.9530	0.9508	0.9648	**0.9704**	0.9494	0.9471	0.9619	**0.9678**	0.9456	0.9439	0.9598	**0.9637**
Man	29.96	30.13	30.96	**31.54**	29.66	29.86	30.68	**31.12**	29.38	29.53	30.34	**30.64**
	0.9566	0.9500	0.9652	**0.9711**	0.9537	0.9472	0.9628	**0.9683**	0.9502	0.9437	0.9604	**0.9645**
Lena	32.05	32.99	33.78	**34.26**	31.76	32.63	33.46	**33.86**	31.41	32.28	33.15	**33.33**
	0.9661	0.9650	0.9742	**0.9778**	0.9643	0.9631	0.9729	**0.9765**	0.9622	0.9611	0.9717	**0.9741**
Fingerprint	28.47	29.73	30.15	**30.79**	27.97	29.12	29.63	**30.14**	27.44	28.47	29.03	**29.30**
	0.9852	0.9870	0.9885	**0.9902**	0.9828	0.9844	0.9868	**0.9882**	0.9797	0.9813	0.9847	**0.9850**
Average	29.64	30.24	31.03	**31.50**	29.31	29.88	30.68	**31.08**	28.98	29.51	30.33	**30.53**
	0.9611	0.9580	0.9700	**0.9743**	0.9580	0.9568	0.9679	**0.9721**	0.9546	0.9521	0.9656	**0.9686**

TABLE 5. At $\sigma_0 = 50$, the proposed framework (EN-CNN) is compared to alternative approaches for mixed noise (A+R) using PSNR and FSIM.

Images	$r = 25\%$			$r = 30\%$			$r = 40\%$			$r = 50\%$		
	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN	Cai's	WESNR	EN-CNN
Hill	22.12	23.47	**23.83**	21.60	23.00	**23.43**	20.41	21.78	**22.19**	19.08	20.45	**20.85**
	0.7756	0.8060	**0.8107**	0.7638	0.7956	**0.7994**	0.7373	0.7727	**0.7832**	0.7089	0.7426	**0.7622**
Man	22.00	23.24	**23.59**	21.55	22.79	**23.19**	20.43	21.64	**22.03**	19.18	20.41	**20.81**
	0.7819	0.8136	**0.8163**	0.7742	0.8029	**0.8033**	0.7474	0.7784	**0.7860**	0.7175	0.7490	**0.7684**
Couple	21.73	22.62	**22.76**	21.34	22.27	**22.45**	20.40	21.41	**21.64**	19.30	20.31	**20.60**
	0.7968	**0.8015**	0.7760	0.7903	**0.8014**	0.7741	0.7648	0.7744	**0.7520**	0.7339	**0.7452**	0.7323
Lena	22.25	24.32	**25.05**	22.04	23.79	**24.47**	20.87	22.44	**23.07**	19.61	21.10	**21.67**
	0.7720	0.8310	**0.8692**	0.7579	0.8190	**0.8609**	0.7293	0.7902	**0.8440**	0.6969	0.7533	**0.8178**
Boat	21.80	22.77	**23.09**	21.36	22.35	**22.70**	20.38	21.41	**21.66**	19.22	20.33	**20.63**
	0.7839	0.8050	**0.8065**	0.7733	0.7936	**0.7995**	0.7477	0.7697	**0.7728**	0.7142	0.7385	**0.7508**
Average	21.98	23.28	**23.66**	17.57	22.84	**23.24**	20.49	21.73	**22.11**	19.42	20.52	**20.91**
	0.7820	0.8114	**0.8157**	0.7719	0.8025	**0.8074**	0.7453	0.7770	**0.7876**	0.7130	0.7461	**0.7663**

beyond 50 epochs. The exponential decay rate parameters τ_1 and τ_2 for ADAM optimizer are set as 0.9 and 0.999, respectively. The learning rate is set as 0.001 whereas the first and second moments are initialized as $\mu_1^{(0)} = 0$ and $\mu_2^{(0)} = 0$, respectively. The training phase with these specifications took about four days.

Moreover, in order to train the convolutional neural network, GTX 1080Ti GPU is used. As far as coding for CNN is concerned, Python programming language is used with TensorFlow [59]. In case of classical filtering approaches such as adaptive median filtering and non-local means filtering, MATLAB 2019 is employed. Also, an interface was developed to export intermediate image data from MATLAB to TensorFlow for final processing.

B. RESULTS

For the sake of comparison, the results were obtained for both of types of mixed noise, namely, AWGN+SPIN (A+S) and AWGN+RVIN (A+R). In case of (A+S), the experiments were performed by setting standard deviation of AWGN

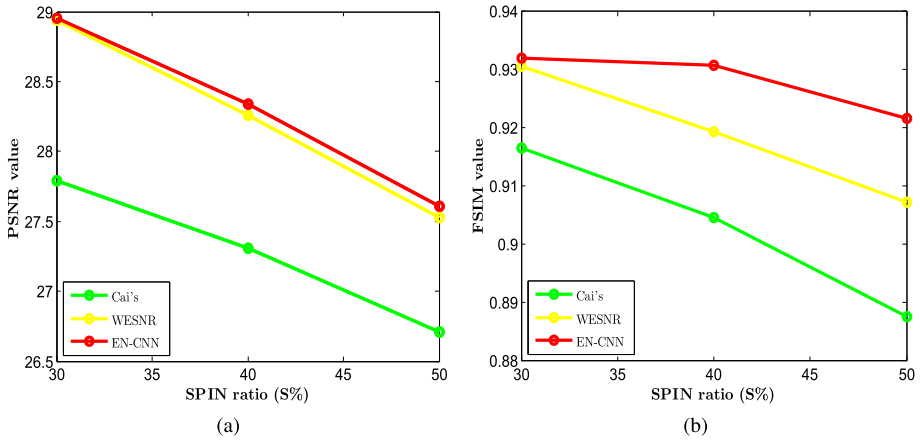


FIGURE 7. Averaging PSNR and FSIM in Table 4 are depicted graphically for $\sigma_0 = 10$.

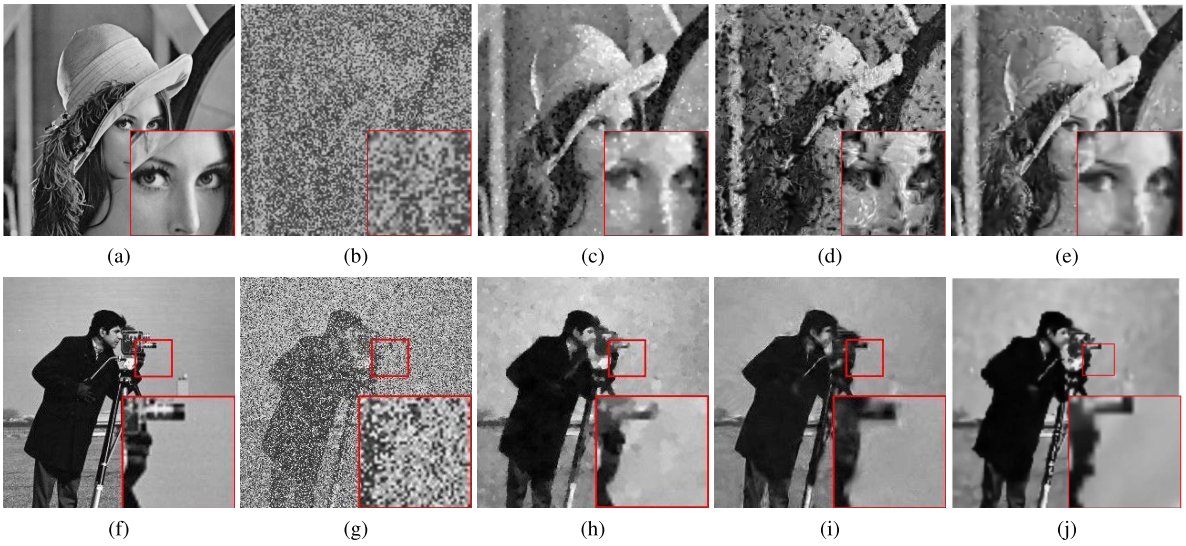


FIGURE 8. (A+S) image noise reduction. Top and bottom rows show (a) the clean image, (b) noisy image, (c) Cai's [54], (d) WESNR [55] and (e) EN-CNN for ($\sigma_0 = 50$ and $S = 50\%$) and ($\sigma_0 = 30\%$ and $S = 40\%$), respectively.

$\sigma_0 = 20, 30, 50$ and 70 in order to see the effect of varying AWGN on the performance of EN-CNN while keeping the SPIN ratio fixed at $S = 50\%$. Similarly, to analyze resilience of EN-CNN against SPIN noise, SPIN ratio was substituted as $S = 30, 40, 50\%$ for the fixed standard deviation of AWGN given by $\sigma_0 = 20$. The similar procedure was adopted for (A+R) type of noise. That is, the standard deviation of AWGN was fixed at $\sigma_0 = 10$ and the RVIN ratio was varied as $r = 5, 10, 15\%$ and then for fixed $\sigma_0 = 50$, RVIN was switched to $r = 25, 30, 40, 50\%$ respectively. The obtained results are reported in Tables 2-5.

C. QUANTITATIVE COMPARISON

For quantitative evaluation, the frequently used measures such as peak signal to noise ratio (PSNR) and feature similarity index measure (FSIM) are selected. The experiments include the results for small AWGN and large IN values and vice versa in order to see the robustness of the proposed mechanism against the variations in each individual component of the mixed noise.

In case of (A+S), it can be observed from the results reported in Tables 2 and 3, that the proposed framework yields significantly better results in case of both small and large value of AWGN and SPIN. For clear visualization of the enhancements, the results are also presented in Fig. 6. In case of (A+R), the similar quantitative analysis is reported in Tables 4 and 5 for various values of RVIN ratio and standard deviation of AWGN. It turned out that the proposed framework performs significantly better than rest of the approaches for small as well as large values of impulsive noise and Gaussian noise. The average values of PSNR for small and large RVIN ratios are graphically depicted in Fig. 7 to clearly visualize the enhancements provided by the EN-CNN. Thus, the proposed framework performs well for both kinds of

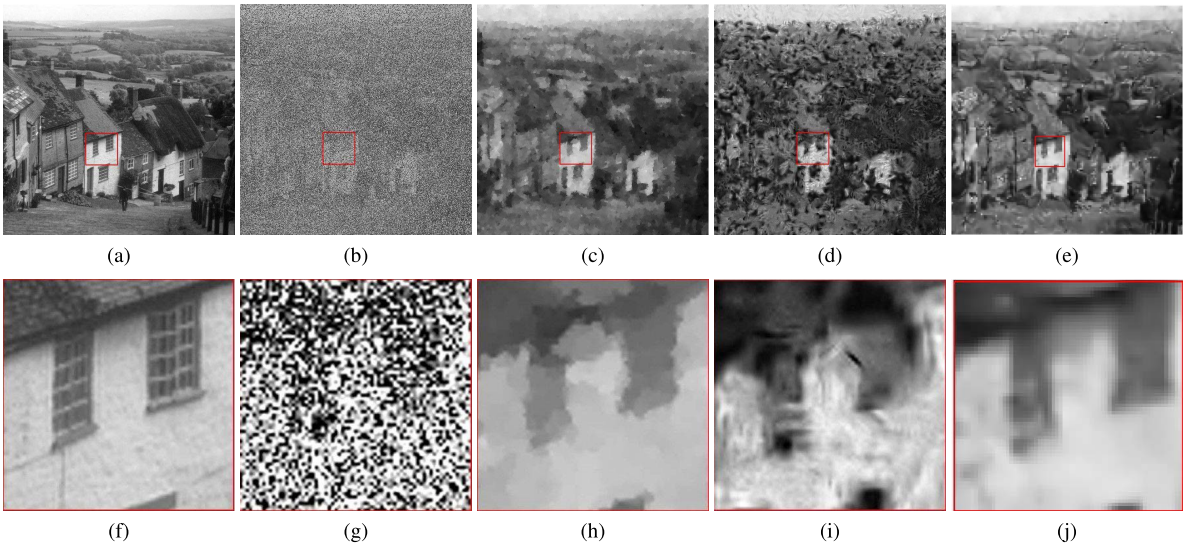


FIGURE 9. (A+S) image noise reduction. Top row shows (a) the clean image, (b) noisy image, (c) Cai's [54], (d) WESNR [55] and (e) EN-CNN for ($\sigma_0 = 50$ and $S = 50\%$). The bottom row shows the magnified sub-image selected from the respective image in the top row.

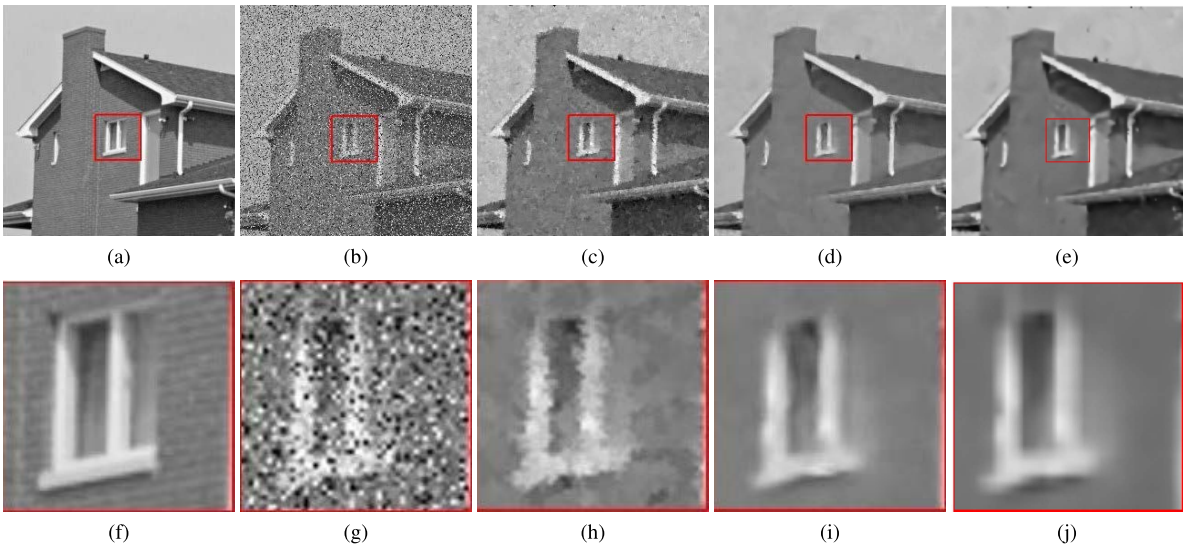


FIGURE 10. (A+R) image noise reduction. Top row shows (a) the clean image, (b) noisy image, (c) Cai's [54], (d) WESNR [55] and (e) EN-CNN for ($\sigma_0 = 20$ and $r = 30\%$). The bottom row shows the magnified sub-image selected from the respective image in the top row.

mixed noise, namely (A+S) and (A+R). In particular, for large values of SPIN or RVIN, the gain in the PSNR and FSIM is more visible. Thus, the enhancements in the results may be attributed to the effective noise estimation strategy prior to applying the deep learning framework to the pre-processed image for mixed noise reduction.

D. QUALITATIVE COMPARISON

In addition to the quantitative analysis, the qualitative analysis of EN-CNN with Cai's [54] and WESNR [55] is also provided. In order to have fair and comprehensive visual comparison, the denoising results are shown for various levels of impulsive noise and Gaussian noise as shown in Figs. 8-11. Among these results, Fig. 8 depicts the restoration results of the cameraman image for (A+S) noise with $\sigma_0 = 30$ and $S = 40\%$. The visual output clearly demonstrates the greater capability of edge and texture retention in comparison to the state-of-the-art approaches. Cai's approach has added some additional structures and diluted the edges around the camera with the sky backdrop as shown in the enlarged sub-images. Furthermore, WESNR has turned the light grey area around the camera's lens into a pure dark region. The suggested method, on the other hand, maintains the edges considerably better. Proceeding further with (A+S) noise for more severe

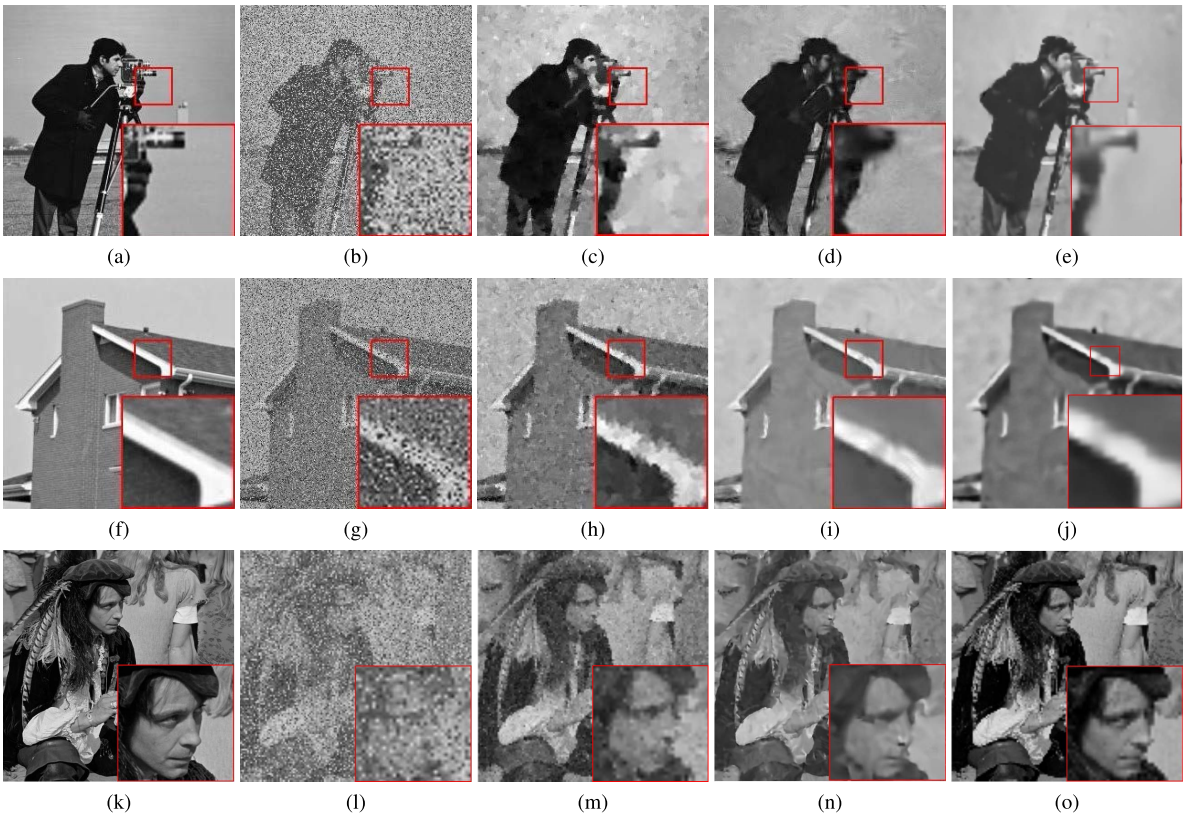


FIGURE 11. (A+R) image noise reduction. Images from left to right show (a) the clean image, (b) noisy image, (c) Cai's [54], (d) WESNR [55] and (e) EN-CNN, respectively. The rows from the top to bottom correspond to mixed noise values ($\sigma_0 = 20$ and $r = 30\%$), ($\sigma_0 = 30$ and $r = 30\%$) and ($\sigma_0 = 30$ and $r = 40\%$), respectively.

TABLE 6. Running time comparison (in seconds) for mixed noise (AWGN+SPIN), at $S = 30\%$.

Image	$\sigma_0 = 10$			$\sigma_0 = 30$			$\sigma_0 = 50$		
	Cai et al.	WESNR	EN-CNN	Cai et al.	WESNR	EN-CNN	Cai et al.	WESNR	EN-CNN
Lena	122	103	**35**	179	167	**38**	245	166	**38**
Barbara	162	75	**30**	186	123	**34**	256	124	**35**
House	28	30	**16**	42	45	**16**	49	48	**17**
Man	133	104	**35**	191	190	**36**	252	130	**34**
Average	111	78	**29**	149	131	**31**	200	117	**31**

noise level given by $\sigma_0 = 50$ and $S = 50\%$, the visual results are shown in Fig. 9. Here, WESNR has deteriorated performance because of the large number of artifacts created. Cai's algorithm performs slightly better than WESNR, however, the geometric structure is substantially smeared, as demonstrated in the enlarged sub-image. On the other hand, the suggested technique performs significantly better in terms of recovering image structure while generating no major artifacts.

Similarly the visual comparisons for (A+R) noise are shown in Fig. 10 and Fig. 11 for AWGN given by ($\sigma_0 = 20, 30$ and 40 whereas RVIN assumes the values $r = 30\%$ and 40%. It can be observed that analogous to case of (A+S) noise, the proposed framework performs equally well for (A+R) noise for both small and large variation in the individual constituents of the mixed noise. Thus, the visual results are in conformity with the quantitative analysis validating the significant advancements over the rest of the approaches compared.

E. TIME COMPLEXITY

Computational complexity of EN-CNN has three main sources, namely, adaptive median filtering (AMF), non-local means filtering (NLM) and run time complexity of EN-CNN. The basic operation during AMF is to arrange the pixels within the neighborhood of referenced pixel location. In case of image with size $k \times k$, the complexity of sorting is $O(k \log_2 k)$ [60]. In case of NLM step, time complexity is $O(k^2 lm)$ where l and m denote the size of search window and patch size, respectively [42]. Lastly, run the time complexity of EN-CNN which is quite negligible owing to GPU implementation of testing phase. For instance,

the denoising process of a 512 × 512 image takes only a fraction of a second. The computational time comparison of the proposed algorithm is depicted in Table 6. It can be observed that on average EN-CNN has the lowest time complexity. It can also be noticed that in case of higher noise densities, EN-CNN takes slightly greater amount of time. This time over head seems reasonable since NLM requires an increase number and size of patches in order to compute the similarity measure. Nevertheless, the computational cost of EN-CNN is significantly less than rest of the algorithms compared.

V. CONCLUSION

In this paper, it turned out that the straight forward training and application of CNN for mixed noise is not capable of learning the noise complexity due to entirely different noise statistics of individual constituents (impulsive noise and Gaussian noise) in the mixed noise. It was also observed that the CNN, effectively trained for certain combinations of impulsive and Gaussian noise yields poor generalization results for any arbitrary combinations of these noise types. In fact, it may become intractable to train a CNN for earch and every combination of both the noise components. To address this problem, an innovative deep learning framework based on convolutional neural network (EN-CNN) is presented. The proposed framework leverages the performance of CNN by introducing the notion of pre-processed image followed by an efficient noise estimation strategy. The pre-processed image should not be confused with data augmentation mechanism which is the inherent part of training process of CNN. In fact, the pre-processed image is the mandatory but external component connected to CNN in order to transform the mixed noise statistics into Gaussian noise statistics alone. However, a mere introduction of a pre-processed image is not sufficient to effectively utilize CNN until and unless the noise in the pre-processed image is precisely determined. For this purpose, a low rank approach based on singular value decomposition is employed on the pre-processed image. In order to confirm the proposed modifications, we utilized CNN with and without these modifications. It turned out that without these modifications, the performance of CNN has been drastically degraded as compared to the state-of-the-art mixed noise approaches. However, in conjunction with the proposed modifications, CNN has successfully outperformed the rest of the mixed noise approaches.

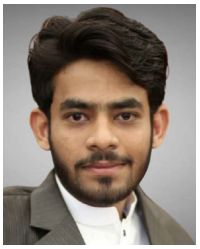
One of the concerns regarding the proposed mechanism is its sole reliance on the efficiency of its first phase. That is, without using low rank approximation and pre-processing steps, the deep learning framework can not infer the types and magnitude of the mixed noise independently. Towards this end, we focus in our future work to include variational auto-encoder mechanism to extract the noise statistics in terms of latent variables. Thus, the estimation as well as removal of the noise may be carried out within the deep learning framework, independently, without considering any classical filtering phase.

REFERENCES

[1] 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.
[2] 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.
[3] A. Vyas, S. Yu, and J. Paik, "Image restoration," in *Multiscale Transforms With Application to Image Processing*. Singapore: Springer, 2018, pp. 133–198.
[4] G. Pok, J. C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 85–92, Jan. 2003.
[5] G. Sicuranza and G. Ramponi, "Adaptive nonlinear digital filters using distributed arithmetic," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 3, pp. 518–526, Jun. 1986.
[6] H. Hwang and R. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.
[7] D.-G. Kim, M. Hussain, M. Adnan, M. A. Farooq, Z. H. Shamsi, and A. Mushtaq, "Mixed noise removal using adaptive median based non-local rank minimization," *IEEE Access*, vol. 9, pp. 6438–6452, 2021.
[8] V. Oliinyk, O. Ieremeiev, and I. Djurović, "Center weighted median filter application to time delay estimation in non-Gaussian noise environment," in *Proc. IEEE 2nd Ukraine Conf. Electr. Comput. Eng. (UKRCON)*, Jul. 2019, pp. 985–989.
[9] Y. Zhang, "Wavelet-based Bayesian fusion of multispectral and hyperspectral images using Gaussian scale mixture model," *Int. J. Image Data Fusion*, vol. 3, no. 1, pp. 23–37, Mar. 2012.
[10] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2005, pp. 60–65.
[11] L. Li, C. Hao, Y. Zhang, W. Lin, A. C. Kot, and X. Sun, "Sparse representation-based image quality index with adaptive sub-dictionaries," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3775–3786, Aug. 2016.
[12] O. Levi, S. Cohen, and Z. Mhabary, "Effective hyper-spectral image segmentation using multi-scale geometric analysis," in *Proc. IADIS Int. Conf. Comput. Graph., Vis., Comput. Vis. Image Process. (MCCSIS)*, 2010, pp. 396–400.
[13] 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.
[14] 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.
[15] 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)*, Los Alamitos, CA, USA, Dec. 2015, pp. 477–485.
[16] X. Liu, M. Tanaka, and M. Okutomi, "Noise level estimation using weak textured patches of a single noisy image," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 665–668.
[17] 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.
[18] M. El Helou and S. Süsstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," *IEEE Trans. Image Process.*, vol. 29, pp. 4885–4897, 2020.
[19] 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.
[20] W. Shan, P. Liu, L. Mu, C. Cao, and G. He, "Hyperspectral image denoising with dual deep CNN," *IEEE Access*, vol. 7, pp. 171297–171312, 2019.
[21] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1791–1798.
[22] Y. Bengio, *Learning Deep Architectures for AI*. Boston, MA, USA: Now, 2009.
[23] F. Wang, H. Huang, and J. Liu, "Variational-based mixed noise removal with CNN deep learning regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 1246–1258, Sep. 2019.

[24] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, vol. 84. Berlin, Germany: Springer, 2013.
[25] J. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in *Proc. Congr. Rec. EASCOM*, 1974, pp. 673–681.
[26] G. Gupta, "Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter," *Int. J. Soft Comput. Eng.*, vol. 1, no. 5, pp. 304–311, 2011.
[27] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, Sep. 1991.
[28] J.-H. Wang and L.-D. Lin, "Improved median filter using minmax algorithm for image processing," *Electron. Lett.*, vol. 33, no. 16, pp. 1362–1363, Jul. 1997.
[29] P. K. Sinha and Q. H. Hong, "An improved median filter," *IEEE Trans. Med. Imag.*, vol. 9, no. 3, pp. 345–346, Sep. 1990.
[30] Y. Zhu and C. Huang, "An improved median filtering algorithm for image noise reduction," *Phys. Proc.*, vol. 25, pp. 609–616, Apr. 2012.
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[32] J. Wang, J. Lin, and Z. Wang, "Efficient convolution architectures for convolutional neural network," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–5.
[33] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Netw.*, vol. 121, pp. 461–473, Jan. 2020.
[34] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.
[35] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, Washington, DC, USA: Curra Associates, Inc, vol. 26, 2013.
[36] T. Clark, A. Wong, M. A. Haider, and F. Khalvati, "Fully deep convolutional neural networks for segmentation of the prostate gland in diffusion-weighted mr images," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, 2017, pp. 97–104.
[37] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning," in *Proc. 3rd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2018, pp. 2319–2323.
[38] T. N. Sainath, B. Kingsbury, A.-R. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc. IEEE Workshop Automat. Speech Recognit. Understand.*, Dec. 2013, pp. 315–320.
[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 37, Lille, France, F. Bach and D. Blei, Eds., Jul. 2015, pp. 448–456.
[40] D.-G. Kim and Z. H. Shamsi, "Enhanced residual noise estimation of low rank approximation for image denoising," *Neurocomputing*, vol. 293, pp. 1–11, Jun. 2018.
[41] Z. H. Shamsi, D.-G. Kim, M. Hussain, and R. M. B. K. Sajawal, "Low-rank estimation for image denoising using fractional-order gradient-based similarity measure," *Circuits, Syst., Signal Process.*, vol. 40, pp. 1–23, Apr. 2021.
[42] Z. H. Shamsi and D.-G. Kim, "Multiscale hybrid nonlocal means filtering using modified similarity measure," *Math. Problems Eng.*, vol. 2015, Aug. 2015, Art. no. 318341.
[43] Z. Zhou, E. Y. Lam, and C. Lee, "Nonlocal means filtering based speckle removal utilizing the maximum a posteriori estimation and the total variation image prior," *IEEE Access*, vol. 7, pp. 99231–99243, 2019.
[44] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, Jul. 2005.
[45] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.
[46] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995.
[47] M. Hashemi and S. Beheshti, "Adaptive noise variance estimation in BayesShrink," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 12–15, Jan. 2010.
[48] S. I. Olsen, "Estimation of noise in images: An evaluation," *CVGIP, Graph. Models Image Process.*, vol. 55, no. 4, pp. 319–323, Jul. 1993.
[49] M. M. Ghazi and H. Erdogan, "Image noise level estimation based on higher-order statistics," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 2379–2397, Jan. 2017.
[50] P. Jiang and J.-Z. Zhang, "Fast and reliable noise level estimation based on local statistic," *Pattern Recognit. Lett.*, vol. 78, pp. 8–13, Jul. 2016.
[51] A. Khmag, S. A. R. Al-Haddad, and N. Kamarudin, "Natural image noise level estimation based on local statistics for blind noise reduction," *Vis. Comput.*, vol. 34, no. 4, pp. 575–587, 2018.
[52] Z. Zhou, L. Huang, M. G. Christensen, and S. Zhang, "Robust spectral analysis of multi-channel sinusoidal signals in impulsive noise environments," *IEEE Trans. Signal Process.*, vol. 70, pp. 919–935, 2022.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[54] J.-F. Cai, R. H. Chan, and M. Nikolova, "Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise," *Inverse Problems Imag.*, vol. 2, no. 2, p. 187, 2008.
[55] J. Jiang, L. Zhang, and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2651–2662, Jun. 2014.
[56] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "RMAF: Relu-memristor-like activation function for deep learning," *IEEE Access*, vol. 8, pp. 72727–72741, 2020.
[57] Y. Xiao, T. Zeng, J. Yu, and M. K. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via l_1–l_0 minimization," *Pattern Recognit.*, vol. 44, no. 8, pp. 1708–1720, Aug. 2011.
[58] 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.
[59] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[60] B.-G. Huang, Z.-T. Lu, C.-M. Ma, and J.-X. Zhao, "Improved adaptive median filtering algorithm," *J. Comput. Appl.*, vol. 31, no. 7, p. 1835, 2011.

DAI-GYOUNG KIM received the Ph.D. degree in applied mathematics from Purdue University, in 1994. He did his Postdoctoral Research with IMA, University of Minnesota, in 1995. He was a Visiting Scholar with the Image Processing Group, UCLA, from 2003 to 2004. He is currently a Professor with the Department of Applied Mathematics and a joint Professor with the Department of Artificial Intelligence, Hanyang University, South Korea. His research interests include numerical PDE, optimizations, image processing, deep learning, and quantum computations, as well as scientific computations in industry.



YASIR ALI received the M.Sc. degree in mathematics from Bahauddin Zakariya University, Multan, Pakistan, in 2018. He is currently pursuing the M.Phil. degree in computational mathematics with the University of the Punjab, Pakistan, under the supervision of Dr. Zahid Hussain Shamsi. His research interests include computer vision, image processing, machine learning, and numerical analysis.

MUHAMMAD ASIF FAROOQ received the Ph.D. degree from the NTNU, Norway, in 2012. He has been working as an Associate Professor at the School of Natural Sciences, National University of Sciences and Technology, Islamabad, Pakistan, since 2012, and still hold this position. His research interests include computational fluid dynamics and numerical analysis.

ASIF MUSHTAQ received the Ph.D. degree in mathematical sciences from the NTNU, Trondheim, Norway, in 2014. He was a Postdoctoral Fellow with Sintef ICT and with the Department of Mathematical Sciences, NTNU. He is currently working as an Associate Professor of mathematics at Nord University, Norway. His research interests include applied mathematics, mathematics education, and computational mathematics.

MUHAMMAD AHMAD ABDUL REHMAN received the M.Sc. degree in mathematics from the University of Education Township Lahore, Pakistan, in 2019. He is currently pursuing the M.Phil. degree in computational mathematics with the University of the Punjab, Pakistan, under the supervision of Dr. Zahid Hussain Shamsi. His research interests include image processing, machine learning, and numerical analysis.

ZAHID HUSSAIN SHAMSI received the M.Sc. degree in mathematics and the M.Sc. degree in computer science from the University of the Punjab, Pakistan, in 2000 and 2003, respectively, and the Ph.D. degree in applied mathematics from Hanyang University, South Korea, in 2016. He is currently an Assistant Professor at the Department of Mathematics, University of the Punjab. His research interests include image processing, machine learning, quantum information theory, relativistic quantum information, and quantum machine learning. He was awarded prestigious HEC Scholarship by the Government of Pakistan for his Ph.D. degree.