

Image De-noising with Machine Learning: A Review

Rini Smita Thakur¹, Shubhojeet Chatterjee¹, Ram Narayan Yadav¹, and Lalita Gupta¹

¹ Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India

Corresponding author: Rini Smita Thakur (e-mail: rinithakur66@gmail.com).

ABSTRACT Images are susceptible to various kinds of noises, which corrupt the pictorial information stored in the images. Image de-noising has become an integral part of the image processing workflow. It is used to attenuate the noises and accentuate the specific image information stored within. Machine learning is an important tool in the image-de-noising workflow in terms of its robustness, accuracy, and time requirement. This paper explores the numerous state-of-the-art machine-learning-based image de-noisers like dictionary learning models, convolutional neural networks and generative adversarial networks for a range of noises like Gaussian, Impulse, Poisson, Mixed and Real-World noises. The motivation, algorithm and framework of different machine learning de-noisers are analyzed. These de-noisers are compared using PSNR as quality assessment metric on some benchmark datasets. The best de-noising results for different noise type is discussed along with future prospects. Among various Gaussian noise de-noisers, GCBD, BRDNet and PReLU network prove to be promising. CNN+LSTM, and MC2RNet are most suitable CNN-based Poisson de-noisers. For impulse noise removal, Blind CNN, and CNN+PSO perform well. For mixed noise removal, WDL, EM-CNN, CNN, SDL, and Mixed CNN are prominent. De-noisers like GRDN and DDFN show accurate results in the domain of real-world de-noising.

INDEX TERMS Convolutional Neural Networks, Dictionary Learning, Generative Adversarial Networks, Image De-noising, Machine Learning

I. INTRODUCTION

Image de-noising has played a pivotal role in recent years with the advent of many latest computer vision applications. The digital image is prone to noise corruption due to camera sensors, illumination level, transmission error, timing error of A/D converters, storage sensor faulty memory location, capturing medium, transmission channel interference, and compression artifacts. In biological imaging, low-light conditions and shorter exposure time degrade the image quality [1]. Image restoration is required in various fields such as medical imaging, remote sensing, underwater de-noising, dehazing applications [2]. The different medical imaging modalities like computed tomography (CT), magnetic resonance imaging (MRI), X-ray, PET, etc. use appropriate de-noising methods for proper diagnosis. Moreover, image pre-processing also includes de-noising procedure prior to the medical image classification or segmentation problem to attain higher accuracy. Remote sensing de-noising restores relevant data from synthetic aperture radar images, satellite images, hyperspectral images, and underwater images.

A. TYPES OF NOISES

The noise classification is done based on its probability distribution function, correlation, nature, and its source. The

different types of noise based on pdf are Gaussian, Rayleigh, Uniform, Impulse, Poisson, etc. According to the correlation, noise is classified into white and color noise. The white noise has uniform power spectral density and zero autocorrelation, unlike color noise. If an image is corrupted with white noise, it implies that all the pixels are uncorrelated with each other. It is additive or multiplicative (speckle) according to nature, i.e., noisy pixels are added or multiplied with the reference image. It is termed as quantization noise or photon noise as per source classification. The description of commonly used noise types is given as follows:

Gaussian Noise: It is statistical and additive in nature which follows normal distribution with zero mean and σ standard deviation and affects all the pixels in the image. The cause of its occurrence is sensor temperature fluctuation and environmental illumination variations. It is commonly found in magnetic resonance imaging, and confocal laser scanning microscopy imaging [3]. The probability distribution function of Gaussian noise is given by the following equation.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

where x is image pixel value, μ is mean and σ is the standard deviation.

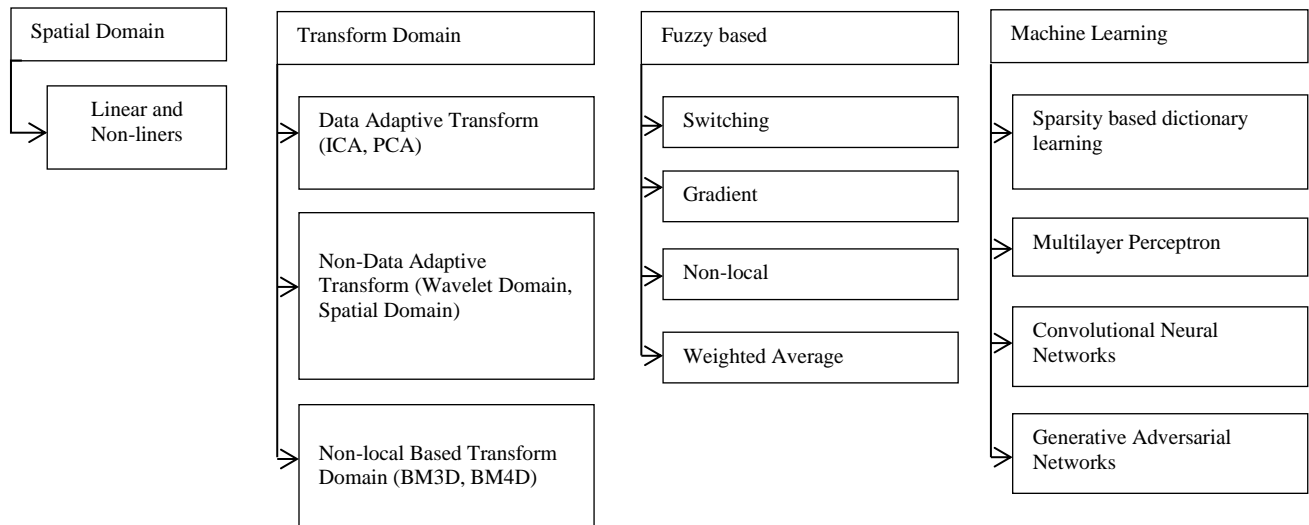


FIGURE 1. Block Diagram illustrating classification of image de-noising techniques

Impulse Noise: It is an additive noise which occurs due to faulty sensors and transmission error. It affects only certain pixels in the entire image, unlike Gaussian noise. It is divided into two parts, i.e., salt and pepper impulse noise (SPIN) and random valued impulse noise (RVIN). In salt and pepper noise corruption, some image pixels take either maximum or minimum value of image dynamic range. Whereas RVIN corruption changes some image pixels with a random value, its detection is more difficult than salt and pepper noise detection. The salt and pepper impulse noise is given by [4]

$$p(x) = \begin{cases} P_a & \text{for } x = a \\ P_b & \text{for } x = b \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where a and b are minimum and maximum pixel values of an image dynamic range. P_a and P_b are probabilities are equal for salt and pepper noise.

Poisson or Photon Noise: The Poisson distribution is used to model photon noise caused by the photon's random arrival on the image sensor [5]. The applications of Poisson noise removal include astronomy, medical imaging, and low-light photography. The conditional probability of Poisson distributed image y for clean image x is given by [6]

$$p(y|x) = \prod_{i,j=1}^N \frac{e^{-x_{i,j}} x_{i,j}^{y_{i,j}}}{y_{i,j}!} \quad (3)$$

where i, j denotes pixel indices.

Gamma Noise: The speckle noise in ultrasound images occurs due to coherent imaging mechanisms from the scatters [7]. It reduces the image sharpness and creates difficulty for lesion diagnosis. It is modeled by Gamma distribution, whose probability distribution function is given by the following equation.

$$p(x) = \begin{cases} \frac{a^b x^{b-1}}{(b-1)!} e^{-ax}, & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where parameters a and b are positive integers.

Rayleigh Noise: The noise in synthetic aperture radar (SAR) images is granular in nature, and it is modeled by Rayleigh distribution [8]. Sometimes, ultrasound images are also prone to Rayleigh noise corruption. The Rayleigh distribution is given by the following probability density

$$p(x) = \begin{cases} \frac{2}{b} (x-a) e^{-\frac{(x-a)^2}{b}} & \text{for } x \geq a \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Cauchy Noise: The atmospheric and underwater acoustic signals of radar and sonar imaging are corrupted with additive heavy-tailed impulse like noise, known as Cauchy noise [9]. The probability distribution function of Cauchy distribution is given by:

$$p(x; \delta, \gamma) = \frac{1}{\pi\gamma} \frac{\gamma^2}{\gamma^2 + (x-\delta)^2} \quad (6)$$

where $\gamma > 0$ denotes the scale parameter, and $\delta \in \mathbb{R}$ denotes the localization parameter.

Mixed Noise: In many real-life applications, images are corrupted by more than one noise type. The mixture of Gaussian and impulse noise is found in computed tomography (CT) images and cDNA microarray imaging [10], [11]. The mixed noise in cDNA microarray imaging occurs due to photon and electronic noise interaction, dust particles on surface of glass slides, and laser reflection. In hyperspectral images, the combination of signal independent additive Gaussian noise and signal dependent multiplicative Poisson noise is found [12].

B. CLASSIFICATION OF IMAGE DE-NOISING TECHNIQUES

The image de-noising methods can be grouped into spatial domain techniques, transform domain techniques, fuzzy filtering-based techniques, and machine learning techniques [13], [14]. The block diagram illustrating the classification of image de-noising techniques is given in Fig.1.

The spatial domain filtering is widely used for image restoration in which filtering operation is directly applied to the image pixels. They are further divided into linear and

non-linear filters. The most common linear filters are the mean filter, Gaussian filter, and Wiener filter. The basic mean filter replaces the particular pixel of operation with the mean value from the pre-defined neighborhood. Similarly, Gaussian filters use a Gaussian kernel with a particular mean and deviation. They suffer from the problem of over-smoothing and blurring of edges. To overcome this problem, Wiener filter was introduced but it is also unsuccessful while operating on sharp edges. Later, non-linear filters were introduced in which output is a non-linear function of input for edge, detail, and texture preservation. The primary examples of non-linear filters are total variation filters, anisotropic diffusion filters, bilateral filter, and fourth-order partial differentiation filter. The bilateral filter replaces pixel value with neighborhood weights which are function of both Euclidian distance and range difference [15],[16]. The detailed comprehensive review of impulse and Gaussian de-noising filters is given in [14].

The transform domain techniques convert the image into the transform domain, and then mathematical operations are carried out on transform domain coefficients. It is followed by inverse transform to restore de-noised image. These techniques are divided into data-adaptive and non-data adaptive techniques based on the transform basis function. The independent component analysis (ICA) and principal component analysis (PCA) are data-adaptive transform methods. The ICA is successfully utilized for non-Gaussian de-noising. PCA is a de-correlation method that transforms the original image dataset into the PCA domain and selects the most significant principal components (maximum Eigen-vectors) for image restoration [17]. Wavelet-based image de-noising is a multi-resolution image analysis technique that uses different mother wavelets such as Daubechies, Haar etc., to obtain wavelet coefficients. It has been used to de-noise Gaussian, salt and pepper, and Poisson noise using the appropriate thresholding operator [18], [19]. In recent years, the most promising non-local means, collaborative filtering method in the transform domain is block-matching and 3D filtering (BM3D) [20]. In this approach, similar 2D image patches are compiled into 3D groups by the block matching process. The collaborative Wiener filtering is done in the transform domain on this 3D group. The improved versions of BM3D are given in [21], [22]. Curvelet filter is based on theory of multiscale geometry (i.e., position, scale, and orientation usage). It gives a better de-noising performance on edges and borders than state-of-the-art wavelet de-noising methods [23]. It uses ridgelet transform as a primary step, and curvelet sub-bands are formed with a filter-bank structure formed by trous wavelet filters. The 2-D contourlet transform provides spatial and directional resolution to keep contours and details intact [24].

The image restoration using fuzzy-based methods considers the image as a fuzzy set and its pixel values as its member. Fuzzy-based filters use fuzzy rules to design membership functions by calculating the gradient's degree in various

directions. The fuzzy impulse noise detection and reduction method calculate the gradient in eight directions for noisy pixel detection prior to the filtering [25]. In histogram fuzzy de-noising filters, the membership function is derived from the input histogram [26]. It consists of the fuzzy detection phase and cancellation phase. A detailed explanation of fuzzy-based techniques is given in [27], [14].

The image de-noising models can be grouped into analytical models (stochastic and deterministic) and machine learning-based models. In analytical models, forward de-noising model is explicitly known to the user, and the solution approach is used based on certain chosen criteria. The deterministic modeling of spatial filters is challenging for each image type. The edge deterioration and blurring are common artifacts in spatial and transform domain techniques. On the other hand, in the machine learning models, the inverse model is learned with the help of image datasets containing clean and noisy image pairs. The most important question arises: what is the relative advantage of machine (deep) learning approach over analytical methods? In deep learning models, computational burden exists in the learning phase, whereas the testing phase consists of a feed-forward model. Whereas analytical methods rely on a computationally demanding optimization process and heuristic selection of hyper-parameters is not the solution for getting good de-noising results. It has been observed that machine learning models give superior performance compared to analytical methods, as feature learning makes a single model apt for considerable variation in the noise level.

Some de-noisers are based on an analytical optimization, which involves iterative process based on some stopping criteria. Although, analytical optimization is involved but it cannot be directly categorized in the machine learning domain which is basically a numerical optimization problem. Some of the important analytical optimization methods are total variation regularization [28] and weighted nuclear norm minimization (WNNM) [29]. Variational-based methods find the appropriate priors such as low-rank priors, non-local self-similarity priors, sparse priors, low-rank priors, and gradient priors. WNNM assigns the weight to the singular value of an image and analytical optimization is done based on some energy function.

In recent years, there is a paradigm shift from analytical models to machine learning models owing to improved image quality assessment metrics. In the following section, machine learning-based image de-noisers are explained in detail. In this paper, the following convention is followed in explaining methods: \mathbf{y} is the noisy input image, \mathbf{x} is the clean image or ground-truth image, \mathbf{v} is the noise component added \mathbf{x} to generate \mathbf{y} , and the final predicted de-noised image from de-noiser is $\hat{\mathbf{x}}$.

C. MACHINE LEARNING BASED IMAGE DE-NOISING

The machine learning image de-noising techniques have made considerable progress with introducing benchmark datasets for a particular application, deep learning

advancements, and increased computational power with Graphical Processing Unit's (GPU's). They are further broadly classified into sparsity-based dictionary learning models, multi-layer perceptron models, convolutional neural network-based models, and generative adversarial network-based models.

1) SPARSITY-BASED DICTIONARY LEARNING MODELS

In the sparsity-based techniques, every image patch is constituted as a linear combination of several patches from an overcomplete dictionary \mathbf{D} . The image encoding is done with the coding vector α over a complete dictionary and l_1 -norm sparse regularizer on coding vector α , i.e., $\min_{\alpha} \|\alpha\|_1$ s.t. $\mathbf{x} = \mathbf{D}\alpha$, following a generalized model given by [30], [31]:

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1 \quad (7)$$

Here, λ is a sparseness-balancing regularization parameter, and $\|\alpha\|_1$ is α 's 1-norm. Another design of the model uses $\|\alpha\|_0$ (α 's 0-norm) in place of $\|\alpha\|_1$. K-Singular Value Decomposition (K-SVD) technique is the pioneering work that uses dictionary learning to frame the sparse representation model. The learning of this model can take place from the benchmark datasets as well as from the input image by K-SVD [32]. The K-SVD is the iterative process in which two consecutive steps take place sparse coding of the examples using the current dictionary and updating the dictionary atoms for optimum data fitting. Some other works as in [33], [34] follow the same workflow like that of K-SVD with variation in dictionary and optimization problem. The clustering-based sparse representation involves a cost function (double header l_1 optimization problem) in which both structural structuring and dictionary learning is used as the regularizer. A typical sparsity-based image de-noising algorithm is given in Algorithm 1.

Algorithm 1: De-noising algorithm of sparsity-based de-noiser [30]

1. Input: \mathbf{y} , where \mathbf{y} is the image observed in the noisy environment
2. Find $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$, where \mathbf{D} is a sparse dictionary constructed to suit \mathbf{x} , $\hat{\alpha}$ is the sparsity constraint, and α is an unknown parameter, λ is a sparseness-balancing regularization parameter set according to:

$$L = \frac{1}{2} \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_0$$
 ($\|\alpha\|_0$ may be replaced by $\|\alpha\|_1$)
 Such that L is as low as possible
3. Find the estimate of \mathbf{x} according to the following:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{D}\alpha\|_2 < \Lambda, \quad \mathbf{x} = \mathbf{D}\hat{\alpha},$$
 where Λ is a small-value limiting parameter
4. Solve the above non-deterministic polynomial problem by using greedy pursuit or convex relaxation
5. Output: $\hat{\mathbf{x}}$

2) MULTI-LAYER PERCEPTRON MODELS

The multi-layer perceptron (MLP) network, as shown in Fig. 2, is the feed-forward model that maps the input image vector (\mathbf{y}) with the output image vector ($\hat{\mathbf{x}}$) with several

intermediate hidden layers. The general equation of MLP network with two hidden layers is given by

Algorithm 2: De-noising algorithm of multi-layer perceptron-based de-noiser [35]

1. Multi-layer perceptron (MLP) (supervised) learns a function $f(\cdot): \mathbb{R}^m \rightarrow \mathbb{R}^o$ by training on a dataset, where m and o are the input dimensions and the output dimensions, respectively
2. Given $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_m\}$ and a target x , it learns a non-linear function approximator for either regression or classification
3. Features \mathbf{Y} are input to the MLP architecture, which has an input layer, one or more non-linear hidden layers, and an output layer that outputs $f(\mathbf{Y})$
4. Each neuron in the hidden layer transforms the values from the previous layer as $g(w_1y_1 + \dots + w_my_m)$, where $g(\cdot): \mathbb{R} \rightarrow \mathbb{R}$ is a non-linear activation function like hyperbolic tan function
5. The output layer transforms the values received from the hidden layer into output values

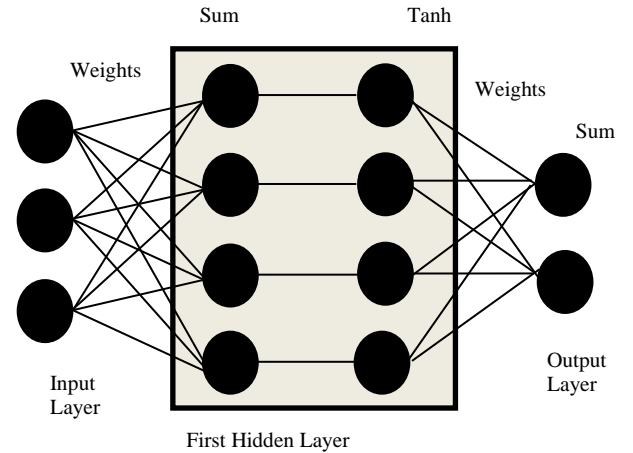


FIGURE 2. Multi-layer Perceptron Network [35]

$$\hat{\mathbf{x}} = \mathbf{b}_3 + \mathbf{w}_3 \tanh(\mathbf{b}_2 + \mathbf{w}_2 \tanh(\mathbf{b}_1 + \mathbf{w}_1 \mathbf{y})) \quad (8)$$
 where \mathbf{w} is the weight matrix, \mathbf{b} is vector-valued bias, and an activation function is \tanh , which operates component-wise. The stochastic gradient is used for training with noisy and clean image pairs. The parameters of MLP are updated by back-propagation, minimizing the mean-square error. To increase the training efficiency, data normalization, proper weight initialization, and learning rate division is done. The noisy image is broken into overlapping patches, and each patch is de-noised separately. MLP estimates the de-noised version of the overlapping noisy patches, and then the average is calculated for overlapped de-noised patches [35]. There is an improvement in de-noising performance when de-noised patches are weighted by the Gaussian window. The MLP with four hidden layers uses time-series images and has shown significant improvement in keeping details and edges intact for SAR images [36]. The trainable non-linear reaction-diffusion model [37] is a feed-forward

architecture that embeds a standard non-linear diffusion model in the neural network. The number of layers of MLP is less because of vanishing gradient compared to convolutional neural networks that limit their performance. A multi-layer perceptron de-noising algorithm is given in Algorithm 2. This algorithm is in accordance with Fig. 3, which depicts a single hidden layer MLP.

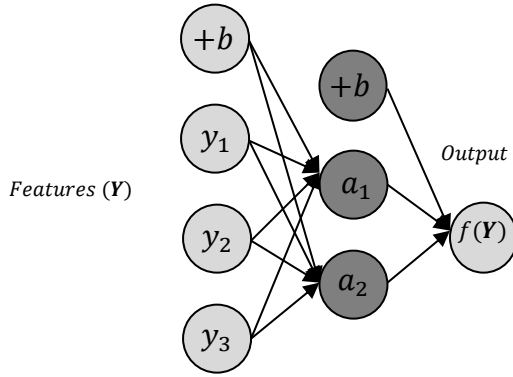


FIGURE 3: One hidden-layer MLP

3) CNN-BASED DE-NOISING MODELS

In recent years, the convolutional neural network (CNN) based models have shown significant improvement in various image quality metrics compared to other state-of-the-art methods [39]. The success of CNN models can be attributed to large modeling capacity and significant advancement in network training and design. The CNN is designed for grid or matrix kind of data as input taking inspiration from the visual cortexes of animals. In CNN models, the convolutional kernel with learnable parameters is shared across all image positions. The convolutional kernel can be visualized as a feature extractor for a particular image restoration application. The convolutional layers have a cascade connection, so extracted features become more complex, hierarchically, and progressively. CNN model consists of an input layer, series of intermediate hidden layers, and the output layer. The convolutional kernel with learnable weights is applied on each layer, followed by some activation function. The output of each layer is fed as the input of the next one. The output of intermediate layers is termed as feature maps. The general equation of the intermediate feature maps (FM) of l^{th} layer of CNN is given by [40]

$$FM_j = A(\sum_{i \in S_j} FM_i^{l-1} * w_{ji}^l + b_j^l) \quad (9)$$

where S_j represents selection of the input feature map, FM_i^{l-1} is the previous feature map, w_{ji}^l is the weight of the convolution kernel of the l^{th} layer, A is the activation function which can be a rectified linear unit, sigmoid function etc. and b_j^l is the bias in the l^{th} layer. The training procedure involves optimizing parameters such as kernels by using clean and noisy image labels with stochastic gradient descent, Adam's algorithm, etc. The Cost function optimization takes place during the training process. The mean square error between the clean image and its de-

noised version is the fundamental cost function. Fig. 4 illustrates the basic architecture of CNN. Algorithm 3 gives CNN de-noising process.

Algorithm 3: De-noising algorithm of CNN based de-noiser [38]

1. Input noisy image y , noise standard deviation σ and clean image x
i.e. $y = x + v$
2. CNN Module: Input: input image or image plus noise level maps
Intermediate units: Convolution + Batch Normalization + Activation function
Output unit: Convolution+ Residual learning
3. Intermediate output is feature maps given by:
 $FM_{ji} = A(\sum_{i \in M_j} FM_i^{l-1} * w_{ji}^l + b_j^l)$, where FM_i^{l-1} represents the feature map of $l-1$ layer, w_{ji}^l and b_j^l are weight and bias of l layer, A is activation function, M_j is selection operator of feature maps.
4. Residual learning implies $\hat{x} = y - R(y)$, where R represents residual learning CNN operator.
5. Loss function: $l(\theta) = \frac{1}{2N} \sum_{i=1}^N \|R(y_i, \theta) - (y_i - x_i)\|^2$, where θ denotes CNN parameters, N are the number of images in the training dataset, y and x represent a noisy and clean image, R is residual learning.
6. If $l(\theta) \cong 0$, model is trained, else retrain for the next epoch.

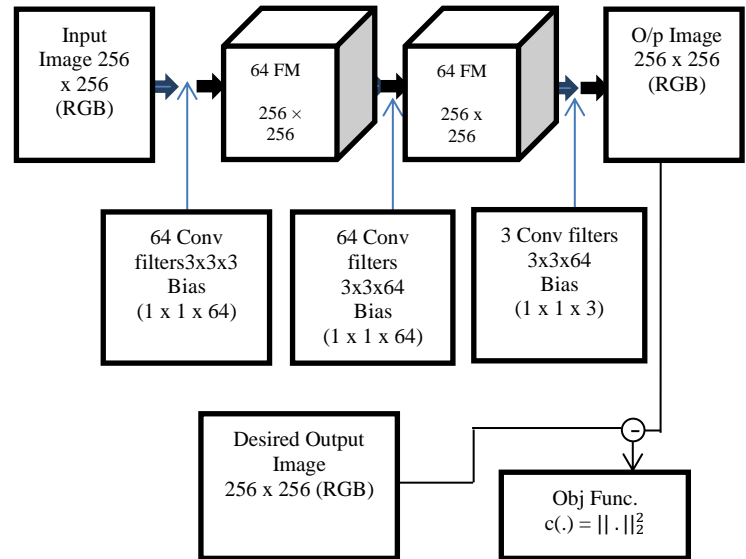


FIGURE 4: CNN Architecture for image restoration [39]

3) GAN-BASED DE-NOISING MODELS

The generative adversarial network (GAN) uses generative modeling with two sub-models, termed generator and discriminator [41]. This network is designed to overcome deep generative model difficulty of learning complex probabilistic distributions. The generator model is used for extracting new plausible images from the problem domain,

whereas the discriminator determines whether the generated image samples are real or fake. The discriminator model acts as an adversarial network. The main motive of the generator network is to obtain image samples that can disguise the discriminator network. Usually, the generator network maps the noisy image with the ground-truth, and the discriminator network uses the loss function to find the difference between the output image of the generator and the ground truth. The discriminator finds whether the predicted image by generator output $\hat{x} = G(y)$ is real or fake. The other de-noising methodology involves the extraction of noise blocks from the input noisy images with the GAN. Thereafter, generated noise blocks along with clean images form a training dataset for CNN to produce de-noised output [42]. The following equation represents GAN objective function :

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{y \sim p_y(y)} [\log (1 - D(G(y)))] \quad (10)$$

where $D(x)$ is the discriminator model, $G(y)$ is the generator model, $p_{data}(x)$ is the real data distribution, $p_y(y)$ is the generated data distribution (i.e., input noisy image y), and E is the expected output. Fig. 5 shows the architecture of GAN for image restoration. Algorithm 4 gives GAN de-noising pseudocode. TABLE I gives advantages and disadvantages of different machine learning image de-noisers.

Algorithm 4: Image de-noising for GAN-based image de-noiser [41], [42]

1. Input noisy image (y) to Generator G ; $p_y(y)$ being the noise distribution.
2. Generator generates reconstructed image data $G(y)$ with distribution $p_g(y)$
3. Pass original image data (x) with distribution $p_x(x)$ and reconstructed image data $G(y)$ to the discriminator
4. Discriminator D outputs the probability of the input belonging to the original data
5. G and D play a two-player minimax game in an adversarial setup, in which G and D try to minimize and maximize a value function based on binary cross-entropy function $V(G, D) = E_{x \sim p_x} [\ln D(x)] + E_{y \sim p_y} [\ln (1 - D(G(y)))]$ respectively
6. Training of GAN: optimize the value function as $\min_G \max_D V(G, D)$
 - a) Fix the learning of G : update generator parameters (θ_D) by gradient ascent using m data samples and m fake samples as
$$\frac{\partial}{\partial \theta_D} \frac{1}{m} [\ln D(x) + \ln (1 - D(G(y)))]$$

For fixed G , $V(G, D)$ will be maximum for $D(x) = \frac{p_x(x)}{p_x(x) + p_g(y)}$
 - b) Fix the learning of D : update discriminator parameters (θ_G) by gradient descent using m fake samples as

$$\frac{\partial}{\partial \theta_G} \frac{1}{m} [\ln (1 - D(G(y)))]$$

For fixed D , $\min_G V = 2JS(p_x || p_g) - 2\ln 2$ is evaluated, where $JS(.)$ is known as Jensen-Shannon divergence.

7. Training of GAN ends when $JS(p_x || p_g)$ becomes 0, i.e., $p_x = p_g$ and $\min_G \max_D V(G, D) = -2\ln 2$

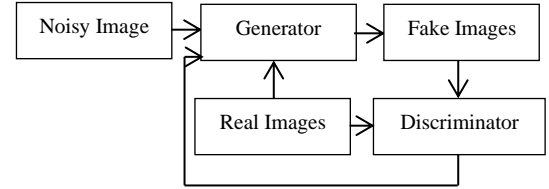


FIGURE 5: Basic GAN Architecture for image restoration

II. MACHINE LEARNING BASED GAUSSIAN DE-NOISERS

The Gaussian noise de-noisers are used in many important applications such as MRI de-noising, optical coherence tomography images, natural images, etc. The CNN-based machine learning model has shown tremendous improvement in image quality assessment metrics as compared to other state-of-the-art networks. The CNN based de-noisers have excelled in Gaussian noise de-noising but real world image de-noising is still a challenging problem. The Gaussian de-noisers are being designed for adaptive white Gaussian, spatially variant Gaussian noise, and blind Gaussian noise. The two important benchmark datasets BSD-68 and Set-12 are used for comparative analysis for Gaussian de-noisers. The CNN-based models follow discriminative learning while GAN's are generative learning models. CNN models require supervised learning, i.e., the availability of noisy and clean image pairs in training datasets. In the recent works, we are progressing towards unsupervised learning due to the lack of clean-noisy image pairs for real-world applications. The training of deep learning models involves optimization of the loss function, which consists of data fidelity term and regularizer. The different de-noiser variants are designed by changing the loss function, the number of layers, training dataset size, activation functions, and so on. The following section describes some benchmark machine learning models which are used for image de-noising:

A. METHODOLOGIES OF DICTIONARY LEARNING MODELS (GAUSSIAN NOISE)

The dictionary learning models achieve sparse representation by updating the dictionary with the training images. The fixed dictionary is limited to a specific type of images, whereas atoms of the learned basis dictionary are empirically learned for any family of images. The learned dictionary provides more efficient image priors for Bayesian estimation as compared to the fixed dictionary. The Gaussian dictionary learning techniques are K-SVD

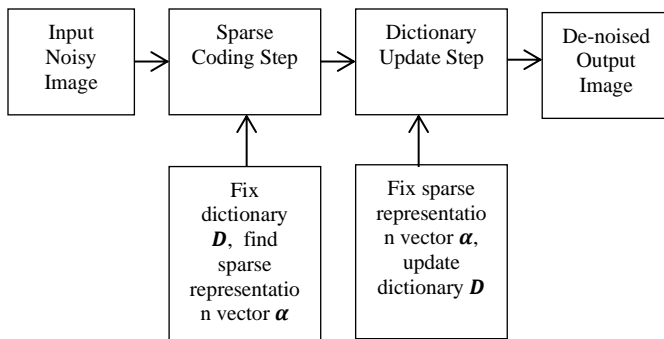


FIGURE 6: Flow Chart of K-SVD algorithm [32]

[32], locally learned dictionary (KLLD) [43], non-local hierarchical learning with wavelets [44], mean-corrected atoms dictionary learning [45]. The K-SVD algorithm finds the best dictionary for the N image samples by solving the following sparsity equation in which dictionary D is initialized with l^2 normalized columns

$$\min_{D, \alpha} \|y - D\alpha\|_F^2 \text{ such that } \|\alpha_i\|_0 \leq T_0 \quad (11)$$

In the above equation, T_0 is the no. of non-zero entries in representation vector α . The two iterative steps are the sparse coding stage and the codebook or dictionary update stage. In the sparse coding step, pursuit algorithm is used for the computation of the representation or sparse vector α for each input image y , by solving the following equation

$$\min_{\alpha_i} \|y_i - D\alpha_i\|_2^2 \text{ such that } \|\alpha_i\|_0 \leq T_0 \quad (12)$$

The next step is the dictionary update stage. Out of K columns of the dictionary, i.e., K atoms, each atom is updated considering one at a time. The image input examples that use a particular atom are retained, the rest of the examples are discarded. The contribution of other atoms is also subtracted from the representation vector. Now, the overall representation error is minimized by singular vector decomposition (SVD) to update the dictionary. The flowchart of K-SVD algorithm is given in Fig. 6.

KLLD algorithm involves clustering, dictionary selection, and coefficient calculation. In clustering local, features capture the local structures of the image data. The next step that is dictionary selection is the optimization according to the clustering done in the first step. In the final step, coefficients are calculated for dictionary atoms, which are linearly combined subjected to the kernel weights. The non-local hierarchical dictionary learning is achieved by sparsity and multiresolution analysis of wavelets in each decomposition level. Recently, K-SVD algorithm is modified based on dictionary learning with mean corrected atoms, which outperformed K-SVD in terms of PSNR. In [46], the authors propose to break up the noisy image into patches and treat the vectorized version of each patch as signals, thereby restricting the dimensionality of each atom in the dictionary. The size of the patch is chosen to allow for enough details of the underlying signal. Overlapping patches are chosen to reduce blocking artifacts that might result at the boundaries. Dealing with patches as signals, the K-SVD algorithm can be effectively scaled to de-noise large images. The algorithm is given in Algorithm 5.

Algorithm 5: De-noising algorithm of sparsity-based dictionary learning models [46]

1. Input image $Y = A = \{A_1, A_2, \dots, A_n\}$ (Y is expressed as a set of A signals)
2. De-noise Y via an ill-posed problem of finding patches Z , where $A = Z + v$
3. De-noised image patches Z are obtained with the help of an optimization problem, which aims at minimizing the following cost function:

$$\hat{X}, \hat{Z} = \arg \min_{X, Z} \|Z - A\|_2^2 + \beta \|DX - Z\|_F^2 + \sum_i \mu_i \|x_i\|_0$$

4. It is solved in terms of smaller optimization problems defined as:
 $\hat{x}_i, \hat{Z} = \arg \min_{x_i, Z} \|Z - A\|_2^2 + \beta \|Dx_i - R_i Z\|_F^2 + \sum_i \mu_i \|x_i\|_0$, where R_i selects i^{th} particle from Z , i.e., $z_i = R_i Z$, and β is a parameter that depends on noise variance
5. The cost function minimizes the error between the restored image and the input noisy image, under the assumption that each patch in the input image can be represented as a sparse linear combination of patches in the dictionary D
6. The closed form solution of the above optimization problem:

$$\hat{Z} = \frac{\beta A + \sum_i R_i D x_i}{\beta I + \sum_i R_i^T R_i}$$

B. METHODOLOGIES OF CNN-BASED MODELS (GAUSSIAN NOISE)

The de-noising CNN (DnCNN) is the benchmark de-noiser which is being used for image restoration tasks like image super-resolution and JPEG image de-blocking apart from the Gaussian image de-noising [38]. DnCNN model overcomes the disadvantage of trainable non-linear reaction-diffusion (TNRD) and a cascade of shrinkage fields [47], which uses specific priors based on the analysis model. So, the priors fail to capture image structures effectively. Moreover, many handcrafted parameters are used during stage-wise greedy training in combination with joint fine-tuning.

The additive noise (v) is combined with the clean image (x) to form the noisy image (y). The DnCNN model uses residual learning with the batch normalization module. In residual learning, the CNN learns the noise component instead of a de-noised image. The residual learning model is given by $R(y) \approx v$, and the desired output image is $y - R(y)$. The batch normalization achieves faster training by mini-batch stochastic gradient by reducing the internal co-variate shift. It is implemented by normalization and scale + shift step before non-linearity in each layer. In the l depth network, the first layer is a convolutional layer with ReLU activation, which uses sixty-four filters ($3 \times 3 \times$ no. of image channel) to give sixty-four feature maps as output. The intermediate layers are repeated units of convolution (sixty-four filters of size $3 \times 3 \times 64$) and

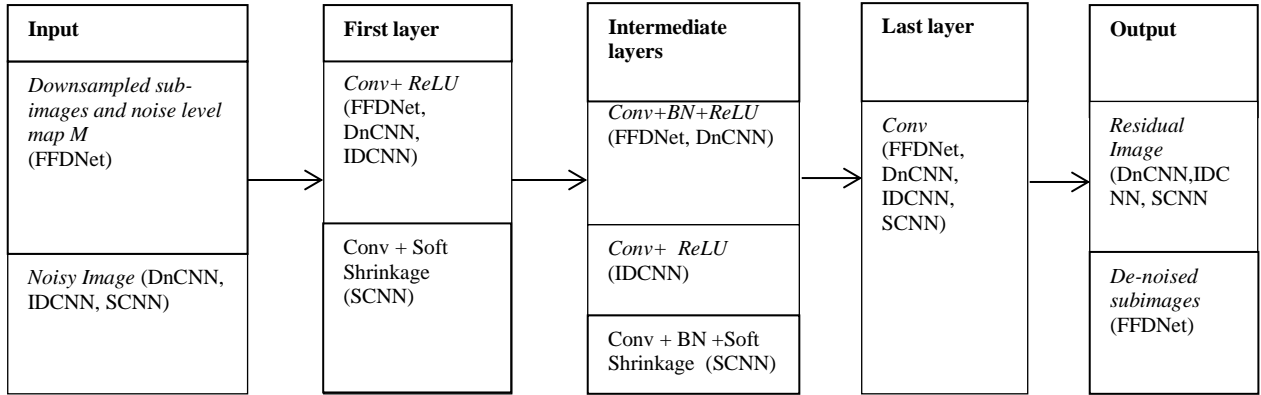


FIGURE 7: Block diagram depicting the architecture of DnCNN, IDCNN, SCNN and FFDNet

ReLU activation with batch normalization. The concluding layer is a convolutional layer which uses same number of filters as the number of image channels of size $3 \times 3 \times 64$. The model gives same de-noising results with both stochastic gradient descent and Adam's algorithm by optimizing the following loss function:

$$l(\theta) = \frac{1}{2N} \sum_{i=1}^N \|R(\mathbf{y}_i, \theta) - (\mathbf{y}_i - \mathbf{x}_i)\|^2 \quad (13)$$

where θ denotes DnCNN parameters, N are the number of images in training dataset, \mathbf{y} and \mathbf{x} represent a noisy and clean image, R is residual learning. There are other denoisers variants whose basic architecture resembles with DnCNN network. Wavelet de-noising CNN i.e., WDnCNN [48] uses residual learning in the novel feature space of the wavelet domain. In this method, the network is trained with four decomposed wavelet sub-bands, and the architecture is the same as that of DnCNN. SCNN [49] is residual learning-based model which uses soft shrinkage activation function for varying noise levels of the input image.

IDCNN [40] is another deep convolutional neural network that follows the same residual learning architecture as that of DnCNN without incorporating batch normalization. This network fails to converge with stochastic gradient descent because of the gradient explosion. So, this network clips the gradient in the specific pre-defined interval, i.e., gradient clipping procedure. It has been observed that network performance improves as the depth of the network increases from four to ten. In this model, a non-fixed noise mask is used during the process so that a single model can be used for different noise levels. The loss function of IDCNN is given by

$$l(\theta) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (14), \text{ where } \mathbf{x} \text{ and } \hat{\mathbf{x}} \text{ denote clean and estimated images respectively.}$$

The ECNDNet [50] is a residual learning model which follows the loss function given in equation (13). The architecture is the same as that of DnCNN. The main feature of the ECNDNet network is the usage of dilated convolution to increase the receptive field size. It reduces

the computational cost and enhances the extraction of more context information. Batch Normalization Residual Network (BRDNet) also uses residual learning, batch re-normalization, and dilated convolution to address the problem of internal co-variate shift for extraction of more features [51]. Deep iterative down-up CNN (DIDN), densely connected hierarchical denoising network (DHDN) [52], and multi-level wavelet CNN (MWCNN) [53] are based on UNet [54] architecture which was designed for the semantic segmentation. The deep iterative down-up CNN (DIDN) [55] is also based on receptive field size variation for improving de-noising results. It consists of four stages: initial feature extraction, down-up block, reconstruction, and enhancement. The initial feature maps are extracted by convolution followed by iterative up and down sampling of feature maps by down-up block. The outputs of all the down-up blocks are fed into the reconstruction block, which has convolutional and parametric rectified linear units. The concatenated output of the reconstruction block is fed into an enhancement block with a convolution unit. DHDN network uses modified Unet architecture to learn large number of parameters, solves vanishing gradient by residual learning and dense connectivity to convolution layers. In MWCNN, multiwavelet transform is integrated into UNet architecture to increase the receptive field size by reducing the resolution of feature maps.

The fast and flexible de-noising convolutional neural network (FFDNet) is the fastest in terms of implementation time, and it can handle spatially variant Gaussian noise [56]. The unique feature of this model is that unlike other networks, the mapping function contains a noise level map in the input. The noise level map plays a crucial role in keeping the trade-off between noise reduction and detail preservation. Conventionally, the mapping function learns de-noised images from noisy images, CNN parameters, and Gaussian noise standard deviation. In FFDNet, the CNN parameters are not affected with variation in Gaussian noise level. It works on downsampled sub-images, which tend to increase the receptive field. The architecture of FFDNet has the same units as that of DnCNN, i.e., convolutional

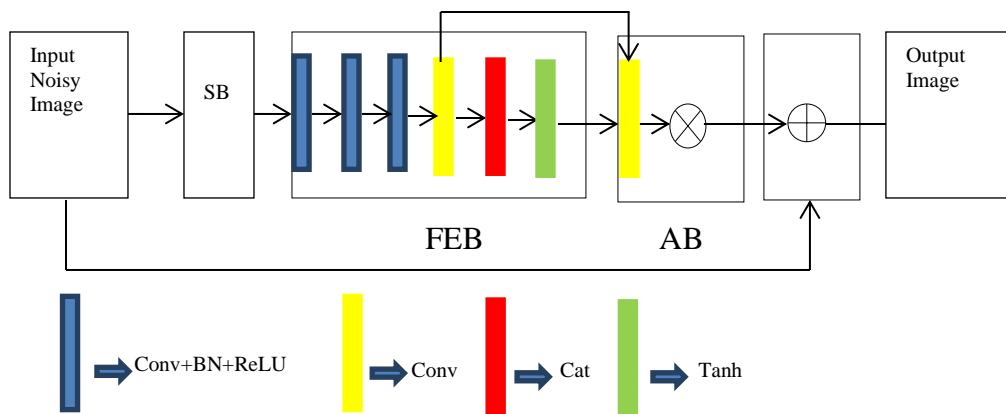


FIGURE 8: Architecture of ADNet [64]

operator in the first layer, repeated units of convolution, batch normalization, and ReLU activation, concluded with the convolutional layer. The Adam's algorithm [57] is used for training to minimize the following loss function.

$$l(\theta) = \frac{1}{2N} \sum_{i=1}^N \|F(y_i, \mathbf{M}_i, \theta) - (x_i)\|^2 \quad (15)$$

where F denotes FFDNet learning function and \mathbf{M} is noise level map.

Some models like NN3D [58] and graph CNN [59] which exploits non-local and local similarities through non-local filter and graphical signal processing. The NN3D uses standard pre-trained CNN in cascade connection with standard non-local filter. The DnCNN, IDCNN, and FFDNet focus towards local features with biased receptive field. The NN3D integrates non-local features in a single modular framework to further improve de-noising performance. Similarly, graph CNN also exploits the non-local similarities by incorporation a graphical convolutional layer. The graph CNN layer works on feature maps to aggregate similar spatially adjacent and spatially distant pixels. The averaging of local and non-local pixels is done to produce the desired feature map. Universal Denoising network (UNet and UNLNet) is another network that integrates convolution and non-local filtering layers for both gray and color image denoising [60]. Fig 7 depicts the block diagram of the combined architecture of DnCNN, IDCNN, SCNN, and FFDNet.

The models, namely PDNN[61], IRCNN [62] and DRUNet [63] integrate the observational model with deep CNN's discriminative learning. The model-based methods require several iterative steps to solve the optimization problem, but they are utilized to solve different image restoration tasks like de-blurring, super-resolution, and de-noising with the single model with the help of an image degradation matrix. They utilize the powerful de-noising capabilities of CNN and prior of the observational models in a single modular framework. In [61], [62], model based optimization is merged with robust image priors with variable splitting technique. The variable splitting reduces the number of CNN parameters and enhances the CNN training efficiency. The observation model is unfolded into discriminating CNN learning, which is composed of

multiple de-noisers modules interleaved with back-projection (BP) that ensure the observation consistencies. DRUNet [63] is the improved version of IRCNN and its methodology involves usage of CNN as deep denoiser prior accompanied by half quadratic splitting based iterative algorithm for solving deblurring, super-resolution, denoising and color image demosaicking.

Recently, the attention-guided de-noising convolutional neural network (ADNet) [64] has outperformed all previous CNN's. They are specifically designed to overcome the disadvantage of increment in network length. As the length of the network increases, the influence of shallow layers becomes weak in de-noising performance. It is divided into four major modules; sparse block (SB), feature enhancement block (FEB), attention block (AB) and reconstruction block (RB). The SB reduces the depth and improves the efficiency of the network with the usage of convolution and dilated convolution operator. It is twelve layer block with dilated Conv + BN + ReLU (second, fifth, ninth, and twelfth layer) and Conv + BN + ReLU in the rest of the layers. The next (13th to 16th) layers form FEB to create robust features by merging global and local features. The first three layer of FEB is Conv + BN + ReLU, and the fourth layer is Conv. The output of the Conv layer and input noisy is concatenated to improve the representation capability further. It is followed by the usage of tanh activation for the non-linearity. The AB consists of just one Conv layer, which compresses the features into the weights to modify the previous layer output. RB is the final stage, which incorporates subtractor for the residual learning process. The architecture of ADNet is given in Fig. 8. The fully convolutional encoder-decoder structure with skip connections is also used for Gaussian and speckle noise removal [65].

The PReLU (parametric rectified linear units) and edge aware based CNN de-noiser is one of the latest works that has produced good PSNR results for both BSD-68 and Set-12 compared to other networks [66]. It is improvised DnCNN network with PReLU as an activation function which learns the slope in negative direction as well. The inclusion on principal component analysis on the feature

TABLE I
ADVANTAGES AND DISADVANTAGES OF DIFFERENT MACHINE-LEARNING MODELS

Machine-learning model	Advantages	Disadvantages
Dictionary-learning model	<ul style="list-style-type: none"> Dictionary-learning models are powerful tools for many image restoration, de-noising and recognition tasks Most local image patches can be well approximated by a sparse linear combination of basis atoms Constructing dictionaries adaptive to the input image via some learning process helps achieve better sparsity than fixed orthogonal dictionaries like DCT, and wavelets Dictionary-based learning models help preserve minute details and texture of images that undergo noise addition 	<ul style="list-style-type: none"> Most dictionary-learning methods consider an over-complete dictionary and formulate the learning process as a minimization problem Minimization problems are very challenging and mostly non-convex Minimization in such a case is usually greedy and computational demanding If correlations among dictionary atoms are not well constrained, the redundancy of the dictionary does not necessarily improve the performance of sparse coding
Multi-layer perceptron model	<ul style="list-style-type: none"> Capability to learn non-linear models Capability to learn models in real-time (on-line learning) 	<ul style="list-style-type: none"> MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy MLP requires tuning a number of hyper-parameters such as the number of hidden neurons, layers, and iterations MLP is sensitive to feature scaling
CNN	<ul style="list-style-type: none"> Less number of parameters as compared to fully connected neural networks Apt for both known and blind Gaussian de-noising Concept of transfer learning i.e. the weights learned by CNN can be used by other network. Many methodologies are being designed owing to simple architecture and mathematical modelling 	<ul style="list-style-type: none"> Analytical approaches have advantage over CNN in merging prior into the inverse problem solution Non-availability of image databases for medical image de-noising and classification Difficulty in case of unsupervised learning in real world scenario.
GAN	<ul style="list-style-type: none"> Unsupervised learning method; can be trained using unlabelled image data as they learn internal representations of data Generate data similar to real image data; can generate images indistinguishable from the real data Learn complex distributions of image data Discriminator is a classifier that can classify objects 	<ul style="list-style-type: none"> Fail to model a multimodal probability distribution of data; suffer from mode collapse. Sometimes, suffer from complete collapse (generated samples are virtually identical) Suffer from the problem of vanishing gradients; training of the initial layers in the net is either extremely slow or effectively stops Internal covariance shift is induced by a change in the input distribution; this slows down the training Training of GANs can be comparatively slow owing to the above reasons

maps in sixteenth layer has led to the extraction of more features. The final step is cascading the network with an adaptive bilateral edge aware filter to further refine the edge and texture details.

C. METHODOLOGIES OF GAN-BASED MODELS (GAUSSIAN NOISE)

The GAN network given [67] in uses DenseNet CNN as the generator network to ease up the vanishing-gradient problem and Wasserstein-GAN as the loss function. The generator network outputs an estimated ground truth image from the noisy image, whereas the discriminator eliminates the difference between the generator output and the ground-truth image. The generator network follows the architecture of DenseNet with eight Dense Blocks, along with input, output, and bottleneck convolution block. The generator extracts both low-level and high-level features efficiently. The discriminator network uses leaky ReLU as the activation function and layer normalization instead of batch

normalization. It has eight convolutional layers and two fully connected layers, which assign a probability to generated images and ground-truth images. The value function of de-noising GAN network is given by

$$\min_{G \in \mathcal{D}} \max_{D \in \mathcal{D}} V(D, G) = E_{x \sim p_{data}(x)}[D(x)] - E_{y \sim p_y(y)}[D(y)] \quad (16)$$

where \mathcal{D} is the set of 1-Lipschitz functions. The objective is to make an approximation of $K \cdot W(p_{data}(x), p_y(y))$, in which K is a Lipschitz constant and W is a Wasserstein distance. The gradient penalty term is added so that the gradient of the discriminator network does not exceed K , and is given by

$$\lambda E_{y \sim p_y(y)}[\|\nabla_y D(y) - 1\|_2^2] \quad (17)$$

Loss function is the combination of content loss and adversarial loss given as

$$l = \lambda l_{GAN} + l_{content} \quad (18)$$

where, content loss is given by l_1 or l_2 norm, and

adversarial loss is given by Wasserstein-GAN critic function.

The GAN-CNN based blind de-noiser (GCBD) model [42] extracts noise blocks from the clean images. The GAN produces a noisy block instead of the de-noised image. The noisy blocks extracted from the GAN are used for the creation of a training dataset for CNN. The GCBD model is a cascade connection of GAN followed by CNN. The generated blocks by GAN along with extracted noise blocks are used for training the discriminative learning based CNN. The GCBD can be used when there is an absence of paired data for the supervised training of CNN. It gives promising results for Gaussian noise, mixed noise, and real-world noisy images. The limitation is that noise is taken only as additive white noise with zero-mean.

III. MACHINE LEARNING BASED IMPULSE DE-NOISERS

A. METHODOLOGIES OF DICTIONARY LEARNING MODELS (IMPULSE NOISE)

Wang et al. have proposed an adaptive dictionary-learning-based method to preserve image structure in impulse-contaminated images with the help of a robust l_1 -norm data-fidelity term to help impulse noise cancellation [68]. In this algorithm, the restoration problem is mathematically formulated into an $l_1 - l_1$ minimization objective and solved under the augmented Lagrangian framework through a two-level nested iterative procedure. The algorithm has high image restoration power to produce restored images with a high PSNR value. Guo et al. [69] have introduced a novel algorithm to enhance image sparsity to help remove salt and pepper noise removal with a fast multiclass dictionary learning, and then both the sparsity regularization and robust data fidelity are formulated as minimizations of $l_0 - l_0$ norm for impulse noise removal. Additionally, a numerical algorithm of modified alternating direction minimization is derived to solve the proposed de-noising model. This algorithm excels in image detail preservation. Deka et al. in [70] have proposed a novel two-stage de-noising method for removing random-valued impulse noise from an image. In the first stage, an impulse noise detection scheme is used to detect the pixels which are likely to be corrupted by the impulse noise, viz., noise candidates. In the second stage, the noise candidates are reconstructed by the image inpainting method based on sparse representation in an iterative manner until convergence is achieved. This algorithm works well in terms of both visual and quantitative aspects.

B. METHODOLOGIES OF CNN-BASED MODELS (IMPULSE NOISE)

Chen et al. have proposed a blind CNN architecture for random-value impulse noise (RVIN) removal [71]. This improvised de-noising mechanism for RVIN suppression works on the principle of flexible noise ratio prediction, which proved to be better than DnCNN-based RVIN suppression by eliminating unnecessary dependence on the exact perception of the noise ratio. Random patches are

selected from the RVIN-corrupted test image and feature vectors that indicate whether the centre pixel is contaminated or extracted by the predictor. These feature vectors are composed of numerous statistics, viz., the multiple rank-ordered absolute differences (ROADs), the clean pixel median deviation (CPMD), and the edge pixel difference (EPD). They are rapidly mapped to noisy/clean (1 for noisy, 0 for clean) labels by the pre-trained noise detector. According to the ratio of the obtained noisy labels to the total number of selected patches, the predictor provides the noise ratio of the whole image. From the output of the NRP, i.e., the predicted noise ratio, the most appropriate DnCNN specifically trained for this noise ratio is exploited for de-noising. Under the guidance of the NRP, the proposed method has the ability to handle unknown noise ratios. This method performs well in terms of execution efficiency and image restoration. Turkmen [72] has proposed an artificial neural network for de-noising RVIN-incorporated images by detecting the noisy pixels. The statistics used to detect the RVIN noisy centres are rank-ordered absolute differences (ROADs), and rank-ordered logarithmic difference (ROLD) values. These are the inputs to the ANN for the detection process. After the detection process is completed, the corrupted pixels are restored by the edge-preserving regularization (EPR) method, allowing edges and noise-free pixels to be preserved. This mechanism works well in the presence of high-density RVIN.

Li et al. [73] have improvised the usage of densely connected convolutional networks (DenseNet) to de-noise images corrupted by impulse noise with the help of CNN to learn pixel-distribution features from noisy images. The proposed method, viz., a densely connected network for impulse noise removal (DNINR), captures the pixel-level distribution information using wide and transformed network learning. This mechanism shows significantly better results in terms of edge preservation and noise suppression.

Khaw et al. [74] have used an efficient CNN with particle swarm optimization (PSO) for high-density impulse noise removal. This high-density impulse noise detection and removal model mainly consists of two parts: impulse noise removal and impulse noisy pixel detection for restoration. The deep CNN architecture facilitates the de-noising procedure to filter out noise from the noisy images. The PSO algorithm optimizes the threshold values for detecting impulse noisy pixels. The method is robust and works well on both gray and colour images in terms of both qualitative and quantitative aspects.

The RVIN can also be removed with the combination of classifier and regression CNN [75]. Classifier network separates noisy and noise-free pixels. Thereafter, the regression network uses noise-free pixels along with the original noisy input image to predict the output image. Batch Normalization is embedded in both classifier and regression network to accelerate the de-noising performance.

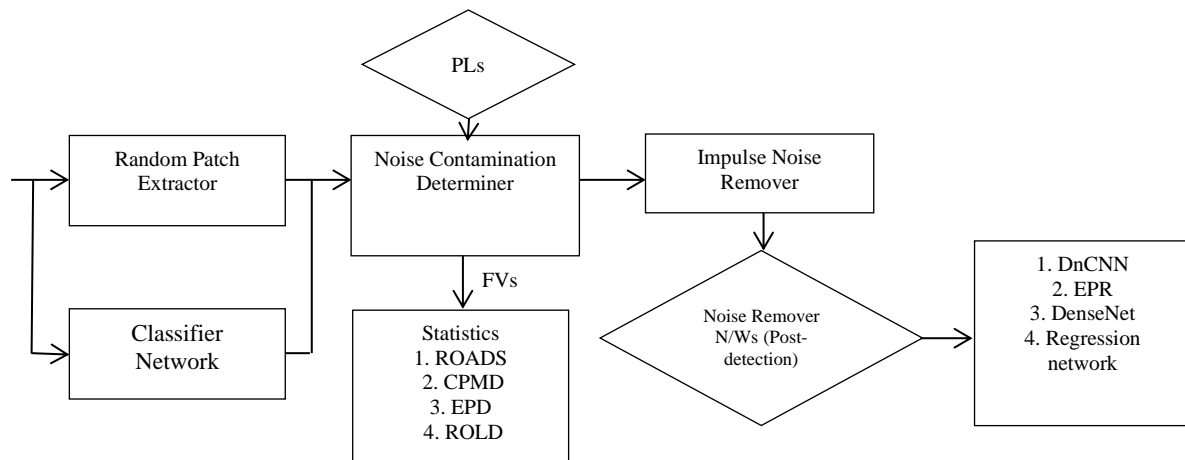


FIGURE 9: Block Diagram depicting the basic concept of impulse de-noisers

Fig.9 shows the overall architecture of the Impulse-noise removal model. The first step is the extraction of a random patch from the noise-corrupted image, and then a classifier N/W is used to predict noisy labels. Thus, the noise-contamination determiner determines the predicted labels (PLs) in case of Jin et al., and extracts feature vectors (FVs) in case of Chen and Turkmen. These feature vectors are composed of numerous statistics, viz., ROADS, CPMD, EPD, ROLD, etc. Finally, a de-noiser N/W is used to de-noise the contaminated image based of the identified noisy centres.

IV. MACHINE LEARNING BASED POISSON DE-NOISERS

Poisson noise is a special type of noise that is not additive in nature. Unlike the Gaussian noise, the noise power is measured by the peak value as its strength is dependent on the image intensity. It is natural to define the noise power in an image by the maximal value in the image, i.e., its peak value. Thus, Poisson de-noisers are described in terms of the peak value as the strength of the noise power.

A. METHODOLOGIES OF DICTIONARY LEARNING MODELS (POISSON NOISE)

Giryes et al. [76] have proposed a novel method to apply the sparse-representation technique to image patches extracted, adopting the same exponential idea. The proposed algorithm uses greedy pursuit with boot-strapping based stopping condition and dictionary learning within the de-noising process. The stopping criterion is novel in its nature. The paper effectively migrates from the Gaussian Mixture model (GMM) to a dictionary-learning based model by resolving the difficulties involved in the conversion process. The reconstruction performance of the proposed scheme is competitive with leading methods in high SNR, and achieving state-of-the-art results in cases of low SNR.

B. METHODOLOGIES OF CNN-BASED MODELS (POISSON NOISE)

Kumwilaisak et al. [77] have proposed a method (CNN+LSTM) based on Deep Convolutional Neural and

Multi-directional Long-Short Term Memory Networks to de-noise images of Poisson noise. CNN layers are used to extract image features and to estimate noise bases in the images, and the multi-directional LSTM layers are efficiently used to memorise the statistics of residual noise components, which possess long-range correlations and are sparse in the spatial domain. The Blahut-Arimoto algorithm is used to numerically derive a distortion-mutual information function for the image de-noising algorithm. The algorithm shows state-of-the-art performance in terms of objective and subjective qualities. Su at al. [78] have proposed a novel method to tackle the problems caused due to Poisson noise in the low-light imaging field. This proposal is that of a deep multi-scale cross-path concatenation residual network (MC2RNet) which incorporates cross-path concatenation modules for de-noising. MC2RNet learns the remnant residue between the noisy and the latent clean image to facilitate the model training procedure. This method opts for blind Poisson training over discriminative de-noising algorithms to train a single model for handling Poisson noise with different levels. The algorithm shows a better performance in terms of peak signal-to-noise ratio and visual effects. Ramez et al. [79] have proposed a flexible and data-driven method to de-noise Poisson-corrupted images, which reduces the heavy ad hoc engineering load occurring due to computational post-processing in the contemporary de-noising procedures. They have used a powerful framework of deep CNNs and a training mechanism that trains the same network with images having a specific peak value. Thus, by using a supervised approach and the representation capabilities of deep CNNs, and using a specific class of images for training, the authors have presented a comparatively simple method that shows state-of-the-art performance both qualitatively and quantitatively and is an order of magnitude faster than other methods. Ramez et al. [80] have introduced a methodology that exploits the architecture of a fully convolutional CNN that uses shallow layers to handle local noise statistics and deeper layers to recover edges and enhance textures. The de-noiser is made

class-aware by exploiting semantic class information that boosts performance, and enhances textures, and reduces artifacts. The residual learning based Gaussian de-noiser (DnCNN) [38], discussed in section 2.2 can also be trained for Poisson noise removal with training with Poisson noise corrupted data patches and relevant hyper-parameter settings [77].

V. MACHINE LEARNING BASED MIXED NOISE DE-NOISERS

The extraction of the clean image from mixed noise corrupted image is very complex problem because of the high-level of non-linearity in the noise distribution. The combination of Gaussian and impulse noise is present in many practical applications. The CNN based transfer learning models, dictionary learning model, and variational based mixed noise model are the machine learning models developed for mixed-noise removal. The comparative analysis of various mixed noise models is difficult as mixed noise can be modeled in different ways. The expression of noisy image pixel y obtained by corruption with the Gaussian and SPIN noise is given by

$$y = \begin{cases} d_{min} & \text{with probability } p/2 \\ d_{max} & \text{with probability } p/2 \\ x + v & \text{with probability } 1 - p \end{cases} \quad (19)$$

where d_{min} and d_{max} are minimum and maximum value in the entire image dynamic range with probability of $p/2$. The AWGN noise v is added with the probability $1 - p$. Similarly, the expression of noisy image pixel y obtained by the corruption with the SPIN, RVIN and Gaussian noise is given by:

$$y = \begin{cases} d_{min} & \text{with probability } p/2 \\ d_{max} & \text{with probability } p/2 \\ d & \text{with probability } r(1 - p) \\ x + v & \text{with probability } (1 - r)(1 - p) \end{cases} \quad (20)$$

where d is random pixel value with the probability $r(1 - p)$.

A. METHODOLOGIES OF DICTIONARY LEARNING MODELS (MIXED NOISE)

The dictionary learning models are designed for mixed Gaussian noise of different standard deviation and mixed Gaussian-impulse noise [84]. The energy minimization model with the weighted $l^2 - l^0$ norm is being used for mixed noise removal such as Gaussian-Gaussian mixture, impulse noise, and Gaussian-impulse noise. It integrates maximum likelihood estimation and sparsity over the learned dictionary. Modified-SVD is used for low rank approximation. In the recent structured dictionary learning model [87], two structured dictionary learning models are combined together. The data-fidelity term uses l_p -norm fidelity to fit image patches and l_q -norm regularizer for the sparse coding. The authors in [32] propose a novel algorithm to tackle mixed Gaussian noise, i.e., the K-SVD algorithm, which generalizes the K-means clustering process for adapting dictionaries in order to achieve sparse signal representations on a given set of training signals. A dictionary is sought for that leads to the best representation for each member in the set, under strict sparsity constraints.

B. METHODOLOGIES OF CNN-BASED MODELS (MIXED NOISE)

The CNN-based transfer learning, four-stage convolutional filtering model is mixed noise de-noiser designed for a mixture of Gaussian and impulse noise [82]. It uses a rank order filter in the preprocessing step, which is Cai's filter in case of Gaussian and SPIN, whereas the combination of adaptive median filter and adaptive center weighted median filter is used in the case of Gaussian, SPIN, and RVIN. The bilinear interpolation is performed on rank order filter output to get a slightly smoother version of the noisy image. The purpose of bilinear interpolation is to suppress high-frequency components that occurred due to rank order

TABLE II
COMPARISON OF PSNR (IN dB) VALUE OF DIFFERENT MACHINE LEARNING MODELS ON SET-12 DATASET

Images	C.Man	Hous e	Pepper s	Starfis h	Monar ch	Airpla ne	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Gaussian noise level $\sigma=15$													
K-SVD [32]	-	34.30	-	-	-	-	30.97	-	-	-	30.46	-	-
K-LLD [43]	-	33.81	-	-	-	-	30.91	-	-	-	30.65	-	-
CSF [47]	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	32.08	31.98	32.32
TNRD [37]	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.50
DnCNN [38]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
IDCNN1[40]	32.54	34.87	33.24	-	35.49	32.79	-	33.75	33.15	31.81	-	-	-
IDCNN2 [40]	32.24	34.83	33.11	-	35.38	32.68	-	33.70	32.98	31.73	-	-	-
FFDNet [56]	32.42	35.01	33.10	32.02	32.77	31.58	31.77	34.63	32.50	32.35	32.40	32.45	32.75
PDNN[61]	32.44	35.40	33.19	32.08	33.33	31.78	31.48	34.80	32.84	32.55	32.53	32.51	32.91
GraphCNN [59]	32.58	35.13	33.27	32.42	33.25	31.84	31.89	34.57	32.84	32.41	32.42	32.40	32.917
ECNDNet[50]	32.56	34.97	33.25	32.17	33.11	31.70	31.82	34.52	32.41	32.37	32.39	32.39	32.81
IRCNN [62]	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.77
BRDNet[51]	32.80	35.27	33.47	32.24	33.35	31.85	32.00	34.75	32.93	32.55	32.50	32.62	33.03
ADNet [64]	32.81	35.22	33.49	32.17	33.17	31.86	31.96	34.71	32.80	32.57	32.47	32.58	32.98
ADNet-B [64]	31.98	35.12	33.34	32.01	33.01	31.63	31.74	34.62	32.55	32.48	32.34	32.43	32.77
PReLU [66]	33.18	35.59	33.54	33.17	34.20	32.65	32.73	35.21	32.25	32.90	32.76	32.78	33.41
MWCNN [53]	-	-	-	-	-	-	-	-	-	-	-	-	33.20
Gaussian noise level $\sigma=25$													
K-SVD [32]	-	32.12	-	-	-	-	28.12	-	-	-	27.59	-	-

K-LLD [43]	-	31.77	-	-	-	-	28.40	-	-	-	28.17	-	-
MLP [35]	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.03
CSF[47]	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.83
TNRD[37]	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.06
DnCNN[38]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
IDCNN1[40]	30.06	32.94	30.79	-	32.85	30.06	-	31.13	30.45	29.18	-	-	-
IDCNN2[40]	29.98	32.87	30.71	-	32.77	30.00	-	31.08	30.18	29.12	-	-	-
FFDNet [56]	30.06	33.27	30.79	29.33	30.14	29.05	29.43	32.59	29.98	30.23	30.10	30.18	30.43
PDNN[61]	30.12	33.54	30.90	29.43	30.31	29.14	29.28	32.69	30.30	30.34	30.15	30.24	30.54
GraphCNN[59]	30.12	33.22	30.87	29.76	30.51	29.30	29.48	32.42	30.28	30.17	30.10	29.99	30.516
ECNDNet[50]	30.11	33.08	30.85	29.43	30.30	29.07	29.38	32.38	29.84	30.14	30.03	30.03	30.39
IRCNN [62]	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.38
BRDNet[51]	31.39	33.41	31.04	29.46	30.50	29.20	29.55	32.65	30.34	30.33	30.14	30.28	30.61
ADNet[64]	30.34	33.41	31.14	29.41	30.39	29.17	29.49	32.61	30.25	30.37	30.08	30.24	30.58
ADNet-B[64]	29.24	33.38	30.99	29.22	30.38	29.16	29.41	32.59	30.05	30.28	30.01	30.15	30.46
PReLU [66]	31.20	34.37	31.86	31.06	32.02	30.60	30.73	33.03	30.68	30.82	30.73	30.74	31.49
MWCNN[53]	-	-	-	-	-	-	-	-	-	-	-	-	30.84

Gaussian noise level $\sigma=50$

MLP [35]	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.78
TNRD [37]	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.81
DnCNN[38]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
IDCNN1[40]	26.80	29.89	27.22	-	29.23	26.52	-	27.69	26.50	25.87	-	-	-
IDCNN2[40]	26.82	29.76	27.20	-	29.18	26.45	-	27.68	26.51	25.87	-	-	-
FFDNet [56]	27.03	30.43	27.43	25.77	26.88	25.90	26.58	29.68	26.48	27.32	27.30	27.07	27.32
PDNN [61]	27.12	31.04	27.44	25.95	27.00	25.97	26.42	29.85	27.21	27.42	27.32	27.23	27.50
GraphCNN [59]	27.00	30.16	27.40	25.92	26.89	25.93	26.43	29.32	26.56	27.05	27.19	26.75	27.217
ECNDNet[50]	27.07	30.12	27.30	25.72	26.82	25.79	26.32	29.29	26.26	27.16	27.11	26.84	27.15
IRCNN [62]	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.14
BRDNet [51]	27.44	30.53	27.67	25.77	26.97	25.93	26.66	29.73	26.85	27.38	27.27	27.17	27.45
ADNet [64]	27.31	30.59	27.69	25.70	26.90	25.88	26.56	29.59	26.64	27.35	27.17	27.07	27.37
ADNet-B[64]	27.22	30.43	27.70	25.63	26.92	26.03	26.56	29.53	26.51	27.22	27.19	27.05	27.33
PReLU [66]	28.00	31.75	28.74	27.52	28.57	27.54	27.56	29.62	28.13	27.59	27.65	27.40	28.34
MWCNN[53]	-	-	-	-	-	-	-	-	-	-	-	--	27.79

TABLE III

COMPARISON OF PSNR (IN dB) VALUE OF DIFFERENT MACHINE LEARNING MODELS ON BSD-68

BSD-68							
σ	15		25		50		
DnCNN [38]	31.72		29.23		26.23		
IRCNN [62]	31.63		29.15		26.16		
SCNN [49]	31.48		29.03		26.08		
PDNN [61]	32.29		29.88		-		
FFDNet [56]	31.62		29.19		26.30		
UNLNet [60]	31.95		29.51		26.41		
ECNDNet[50]	31.71		29.22		26.23		
ADNet [64]	31.74		29.25		26.29		
BRDNet [51]	31.79		29.29		26.36		
DRUNet [63]	31.91		29.48		26.59		
MWCNN[53]	31.91		29.46		26.58		
PReLU [66]	32.60		30.48		27.20		
σ	10	20	30	40	50	75	
DnCNN-S [38]	33.88	30.27	28.36	27.11	26.23	24.64	
IRCNN [62]	33.74	30.16	28.26	27.08	26.19	-	
NN3D(WDnCNN) [58]	-	-	28.56	-	26.42	24.91	
FFDNet [56]	33.76	30.23	28.38	27.18	26.29	24.79	
DIDN [55]	33.98	30.44	28.58	27.37	26.47	-	
NN3D(DnCNN) [58]	-	-	28.41	-	26.27	24.71	
NN3D(FFDNet) [58]	-	-	28.37	-	26.29	24.80	
DIDN+ [55]	34.01	30.47	28.61	27.40	26.50	-	
WDnCNN [48]	-	-	28.56	-	26.39	24.85	
DHDN [52]	34.04	-	28.58	-	26.43	-	
Kodak-24							
σ	10	20	30	40	50		
DnCNN-S [38]	34.90	31.47	29.62	28.37	27.49		
IRCNN [62]	34.76	31.38	29.52	28.37	27.45		
FFDNet [56]	34.81	31.47	29.69	28.51	27.62		
DIDN [55]	35.16	31.83	30.04	28.84	27.96		
DIDN+[55]	35.20	31.87	30.08	28.88	28.01		

DHDN [52]	35.24	-	30.11	-	28.01
σ	15	25	35	50	75
BRDNet [51]	34.88	32.41	30.80	29.22	27.49
ADNet [64]	34.76	32.26	30.68	29.10	27.40

TABLE IV
PSNR (in dB) VALUES OF COLOR DE-NOISING NETWORKS

CBSD-68					
σ	10	20	30	40	50
CBM3D [81]	35.89	31.89	29.72	28.08	27.36
DnCNN-S [38]	36.12	32.37	30.32	28.95	27.92
IRCNN [62]	36.06	32.27	30.22	28.85	27.86
FFDNet [56]	36.14	32.34	30.31	28.96	27.96
DIDN [55]	36.48	32.73	30.71	29.36	28.35
DIDN+ [55]	36.52	32.77	30.75	29.40	28.40
DHDN [52]	36.48	-	30.54	-	28.01
σ	15	25	35	50	75
DRUNet [63]	34.30	31.69	28.51		
Kodak-24					
σ	10	20	30	40	50
CBM3D [81]	36.57	32.92	30.89	29.17	28.62
DnCNN-S [38]	36.58	33.20	31.28	29.95	28.94
IRCNN [62]	36.70	33.19	31.24	29.91	28.92
FFDNet [56]	36.80	33.32	31.39	30.08	29.10
DIDN [55]	37.32	33.88	31.97	30.68	29.72
DIDN+ [55]	37.37	33.94	32.03	30.75	29.80
DHDN [52]	37.37	-	32.01	-	29.74
σ	15	25	35	50	75
DRUNet [63]	35.31	32.89	29.86		

TABLE V
PSNR(in dB) COMPARISON OF DIFFERENT METHODS ON LIVE1 DATASET ON POISSON NOISE

Peak Value	DnCNN [38]	DenoiseNet[79]	CNN+LSTM[77]
0.1	18.90	19.15	19.61
1	22.68	22.59	22.87
10	27.77	27.49	27.77
40	30.92	30.72	31.01
80	32.65	32.31	32.71

TABLE VI
AVERAGE PSNR (IN dB) VALUES ON DIFFERENT DATASETS FOR POISSON NOISE

Method	Peak	Boats	Airplane	Baboon	Barbara	Flower	Goldhill	lenna	Monarch	Pens	Pepper	Average
DnCNN [38]	0.1	19.63	18.60	17.34	18.84	19.50	20.89	19.30	18.39	19.92	18.66	19.11
DenoiseNet [79]	0.1	20.10	19.35	17.29	18.98	19.79	21.16	19.26	18.36	19.89	19.23	19.35
CNN+LSTM [77]	0.1	20.56	19.66	17.52	19.20	20.21	21.63	20.40	19.03	20.18	19.62	19.80
DnCNN [38]	1	24.37	22.98	19.30	22.03	24.52	24.46	24.33	23.70	24.30	23.60	23.36
DenoiseNet [79]	1	24.24	22.94	19.35	21.89	24.18	24.47	24.28	23.52	23.99	23.42	23.23
CNN+LSTM [77]	1	24.74	23.54	19.44	22.23	25.22	24.79	24.72	23.82	24.45	23.62	23.66
DnCNN [38]	10	29.76	28.40	22.95	2.20	30.74	28.62	29.27	29.96	29.64	28.32	28.48
DenoiseNet [79]	10	29.44	28.15	22.90	26.57	30.39	28.45	29.03	29.61	29.21	28.19	28.19
CNN+LSTM [77]	10	29.80	28.43	23.01	27.15	30.73	28.68	29.38	29.94	29.56	28.49	28.52
DnCNN [38]	40	32.78	31.23	25.38	30.43	33.66	31.01	31.27	33.23	32.48	30.35	31.18
DenoiseNet [79]	40	32.58	31.14	25.40	30.00	33.48	30.88	31.25	32.98	32.18	30.62	31.05
CNN+LSTM [77]	40	32.91	31.64	25.56	30.49	33.86	31.11	31.58	33.32	32.56	30.90	31.40
DnCNN [38]	80	34.29	32.57	26.78	31.95	35.07	32.34	32.33	34.85	34.10	31.47	32.56
DenoiseNet [79]	80	33.90	32.61	26.80	31.50	34.52	32.16	32.11	34.35	33.46	31.63	32.30
CNN+LSTM [77]	80	34.37	33.16	26.98	32.02	35.16	32.40	32.50	34.97	34.03	31.99	32.76

TABLE VII
AVERAGE PSNR (in dB) VALUE ON SET-10 DATASET FOR POISSON NOISE

Method	Peak	Flag	House	Camera	Man	Bridge	Saturn	Peppers	Boat	Couple	Hill
DenoiseNet [79]		19.45	22.87	21.59	22.49	19.83	26.26	21.43	22.38	22.11	22.82
MC2RNet6-B [78]	1	19.97	23.04	21.95	22.52	19.88	26.93	21.69	22.48	22.15	22.80
MC2RNet6-S [78]		20.04	23.17	21.97	22.54	19.94	27.35	21.70	22.60	22.20	22.94
DenoiseNet [79]		21.38	24.77	23.25	23.64	20.80	28.37	23.19	23.66	23.30	23.95
MC2RNet6-B [78]	2	21.87	24.99	23.42	23.66	20.86	29.01	23.20	23.69	23.40	23.99
MC2RNet6-S [78]		21.92	25.23	23.66	23.76	20.87	29.22	23.35	23.80	23.57	24.07
DenoiseNet [79]	4	23.18	26.59	24.87	24.77	21.81	30.02	24.83	24.86	24.60	25.01

MC2RNet6-B [78]		24.13	26.78	24.84	24.84	21.81	30.94	24.96	24.29	24.76	25.10
MC2RNet6-S [78]		24.24	27.18	25.03	24.95	21.91	31.29	24.97	25.08	24.93	25.25
DenoiseNet [79]		25.73	28.42	26.35	26.10	22.91	32.28	26.45	26.23	26.11	26.26
MC2RNet6-B [78]	8	26.61	28.56	26.39	26.06	22.95	33.28	26.66	26.24	26.26	26.34
MC2RNet6-S [78]		26.71	28.75	26.45	26.24	23.01	33.71	26.68	26.46	26.41	26.43
DenoiseNet [79]		28.94	31.67	29.21	28.74	25.42	36.20	29.77	29.06	29.13	28.71
MC2RNet6-B [78]	30	30.57	31.52	29.04	28.71	25.37	36.40	29.71	28.99	29.16	28.77
MC2RNet6-S [78]		30.95	31.90	29.23	28.83	25.48	36.99	29.84	29.21	29.34	28.93

TABLE VIII
AVERAGE PSNR(in dB) VALUE ON BSD-68 DATASET FOR POISSON NOISE

Method	Peak=1	Peak=2	Peak=4	Peak=8
IRCNN [62] [80]	21.66	22.86	24.00	25.27
DenoiseNet [79]	21.79	22.90	23.99	25.30
MC2RNet6-B [78]	21.92	23.00	24.13	25.39
MC2RNet6-S [78]	22.00	23.08	24.25	25.51

TABLE IX
AVERAGE PSNR (in dB) COMPARISON OF MIXED NOISE (RVIN+AWGN) ON DIFFERENT NATURAL IMAGES

Image	Noise level	Lena	Barbara	Bridge	Boat	Airplane	Pepper	Hill
Mixed CNN [82]	$\sigma=15$,	32.28	25.67	28.86	29.12	32.52	34.07	31.61
[83]	$\rho=0.15$	32.56	29.43	29.16	30.30	32.87	33.49	31.73
Mixed CNN [82]	$\sigma=15$,	29.10	24.17	26.54	27.02	28.81	29.90	29.04
[83]	$\rho=0.30$	31.71	28.32	28.09	29.19	31.88	33.66	30.98
Mixed CNN [82]	$\sigma=15$,	24.87	21.67	22.86	23.62	23.32	24.54	24.69
[83]	$\rho=0.45$	30.36	26.25	26.53	27.60	30.44	32.09	29.86
Mixed CNN [82]	$\sigma=25$,	29.87	24.52	26.61	27.61	30.34	31.65	29.48
[83]	$\rho=0.15$	30.46	27.28	26.72	28.31	30.73	31.76	29.55
Mixed CNN [82]	$\sigma=25$,	27.93	23.36	25.10	26.08	28.00	28.70	27.80
[83]	$\rho=0.30$	29.79	26.35	25.97	27.39	29.90	31.30	28.99
Mixed CNN [82]	$\sigma=25$,	24.88	21.65	22.62	23.45	23.97	24.87	24.59
[83]	$\rho=0.45$	28.54	24.75	24.81	26.10	28.40	30.04	28.01

TABLE X
DE-NOISING RESULTS FOR MIXED GAUSSIAN NOISE ON BARBARA IMAGE (PSNR=19.02 dB)

Method	PSNR (dB)	Method	PSNR (dB)
WDL (case 1) [84]	32.39	K-SVD (known parameters) [32]	27.66
WDL (case 2) [84]	30.07	W-KSVD (unknown parameters) [84]	29.35
K-SVD [85]	26.95	IRCNN [62]	28.95
		EM-CNN [86]	30.68

TABLE XI
RESULTS FOR BLIND DE-NOISING

Noise Type	Gaussian noise		
Mode	Non-blind	Blind	
Method	BM3D [20]	DnCNN-B [38]	GCBD[42]
$\sigma = 15$	31.07	31.61	31.59
$\sigma = 25$	28.57	29.16	29.15
Noise Type	Mixed noise		
Mode	Non-blind	Blind	
Method	BM3D [20]	DnCNN-B[38]	GCBD[42]
$s = 15$	41.08	40.75	42.00
$s = 25$	37.85	37.54	39.87

filtering on the Gaussian noise. It is followed by the four-stage convolutional filtering. The first stage consists of the the conv layer and ReLU activation function followed by the max-pooling layer. The second and third stages consist of the conv layer and ReLU activation function. The fourth stage is the conv layer. The squared Frobenius norm is used as the loss function, and training is done by the back

propagation algorithm. The other CNN model for mixed Gaussian and impulse noise involves two parts: the first half for impulse noise removal and the second half for the Gaussian noise removal [83]. It consists of the input layer, intermediate layers of convolution, batch normalization, and leaky ReLU followed by the convolutional output layer. The second part of Gaussian noise removal has a skip

connection for residual learning. CNN model given in [85] has conv+ReLU+BN as basic building block and shows best structural metrics results for both known and unknown noise level of mixed Gaussian-Impulse noise [88]. The CNN is used as a regularizer in traditional variational based methods for mixed noise removal [86]. The mixed noise parameters are iteratively estimated by variational method followed by noise classification according to the statistical parameters. The methodology is implemented by optimization of sub-problem involving four steps which are regularization, synthesis, parameters estimation and noise classification.

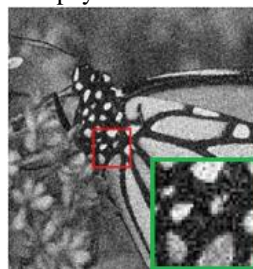
VI. REAL WORLD-DENOISERS

Xu et al. have constructed a benchmark dataset to de-noise real-world images [89]. The authors have used different cameras with different camera settings. They have evaluated different de-noising methods on the new proposed dataset as well as previous datasets for a proper comparison and subsequent analysis. Extensive experimental results demonstrate that the methods designed specifically for realistic noise removal based on sparse or low rank theories, achieve good de-noising performance and are robust. Another observation made by the authors suggests that the proposed dataset is more challenging for the state-of-the-art methods. In Kim et al. [90], a grouped residual dense network (GRDN) is proposed, which is an extended and generalized architecture of the state-of-the-art residual dense network (RDN) [91]. The core part of RDN is the grouped residual dense block (GRDB) and used as a building module of GRDN. Cascading GRDNs aids the de-noising performance significantly. Inspired by the GAN modeling technique, the authors have made their own generator and discriminator for real-world noise modeling. Lin et al. [92] have constructed a new dataset to solve the problem of low availability of proper datasets and obtained the corresponding ground truth by averaging, and then they extended them through noise domain adaptation. Furthermore, they went on to propose an attentive generative network by injecting visual attention into the generative network. During the training, the visual attention map learns noise regions. The generative network pays

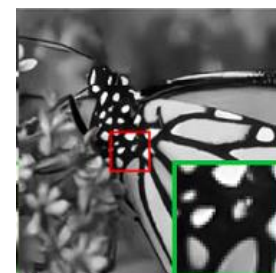
more attention to noise regions, which contributes in balancing between noise removal and texture preservation. Extensive experiments show that this method performs well both qualitatively and quantitatively. Chen et al. have proposed a Deep Boosting Framework (DBF) [93] for real-world image denoising by combining the deep learning into the boosting algorithm. The DBF replaces conventional boosting units by elaborate convolutional neural networks. The outcome is a lightweight Dense Dilated Fusion Network (DDFN) as the boosting unit, which addresses the vanishing gradient problem during training due to the cascading of networks while promoting the efficiency of limited parameters. This method reduces the domain-shift issue with the one-shot domain transfer scheme. This is a strong technique in terms of real-world de-noising. Real-world de-noising has been tested and evaluated on different datasets like DND and NIGHT. DND is a novel benchmark dataset which consists of realistic photos from 50 scenes taken by 4 consumer cameras. The NIGHT dataset is divided into 20 images (denoted as NIGHT-A) and the other 5 images (denoted as NIGHT-B). Another dataset used is RID. It has 20 representative scenes, which are captured under different shooting conditions. The problems faced in real-world de-noising are as follows: (1) The noise in real-world noisy images is very complex, which cannot be described by simpler distributions like Gaussian or Poisson. (2) The inherent practicality of real-world noisy images makes the de-noising more difficult than the synthetic case. (3) the noise distribution may change along with the in-camera imaging pipeline [94]. It thus makes the noise distribution in a captured RGB image different from its Gaussian assumption in the RAW space. (4) The problem of domain shift cannot be neglected in the practical scenario. It can neither be neglected between the synthetic and the real-world noise, but the characteristics of real-world noise can also exhibit differences pertaining to different camera settings (viz., sensor or aperture size), shooting conditions (viz., light, environment, and temperature), and imaging pipelines (viz., smartphone and



(a) Original Image



(b) Noisy Image ($\sigma = 25$)



(c) DnCNN (PSNR 30.28 dB) [38]

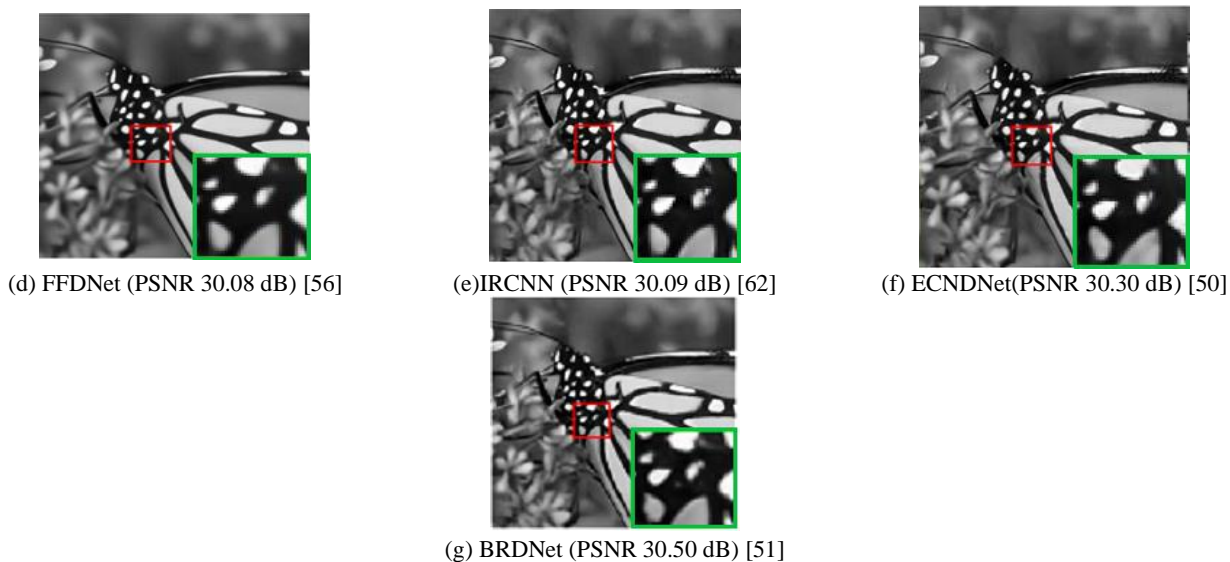


FIGURE 10: Butterfly image (Set-12) de-noising results at $\sigma = 50$ Random-Valued Impulse Noise

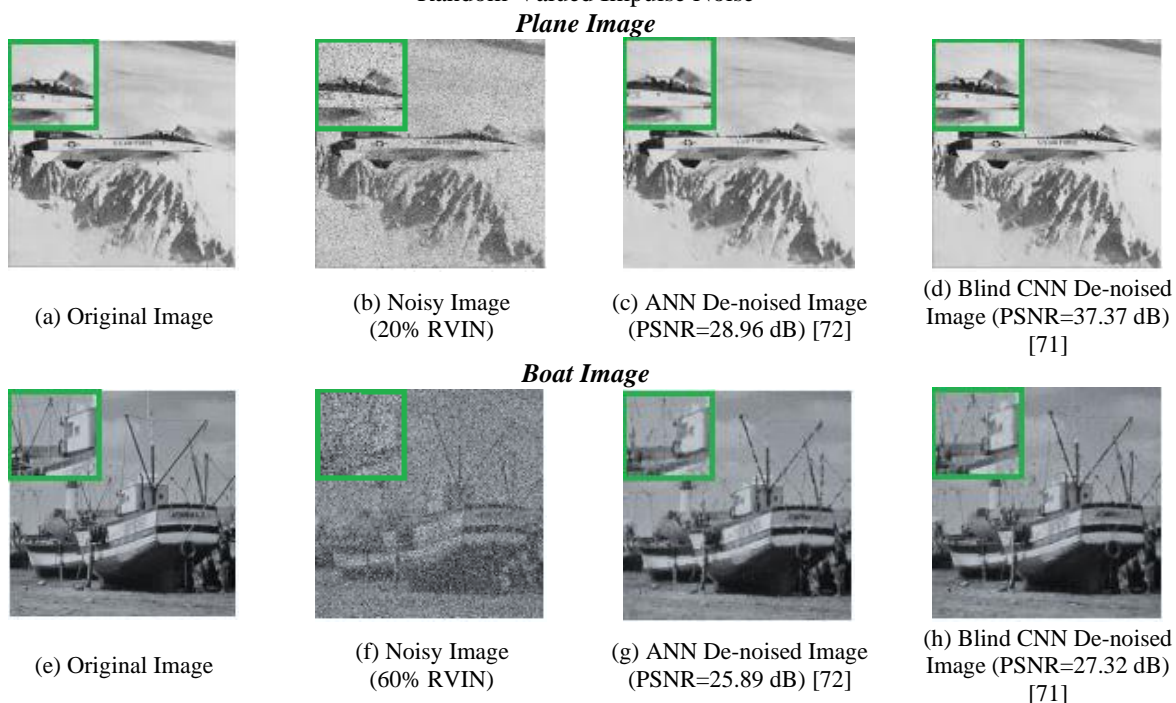
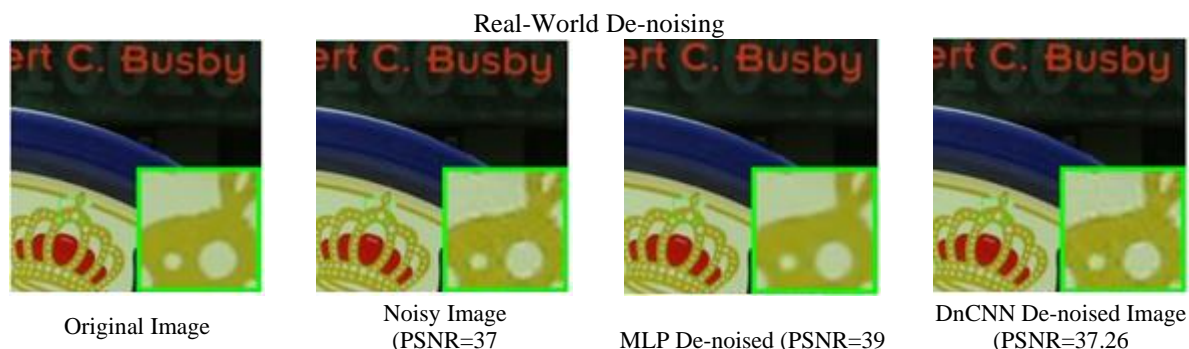


FIGURE 11: RVIN de-noising results for Plane and Boat image



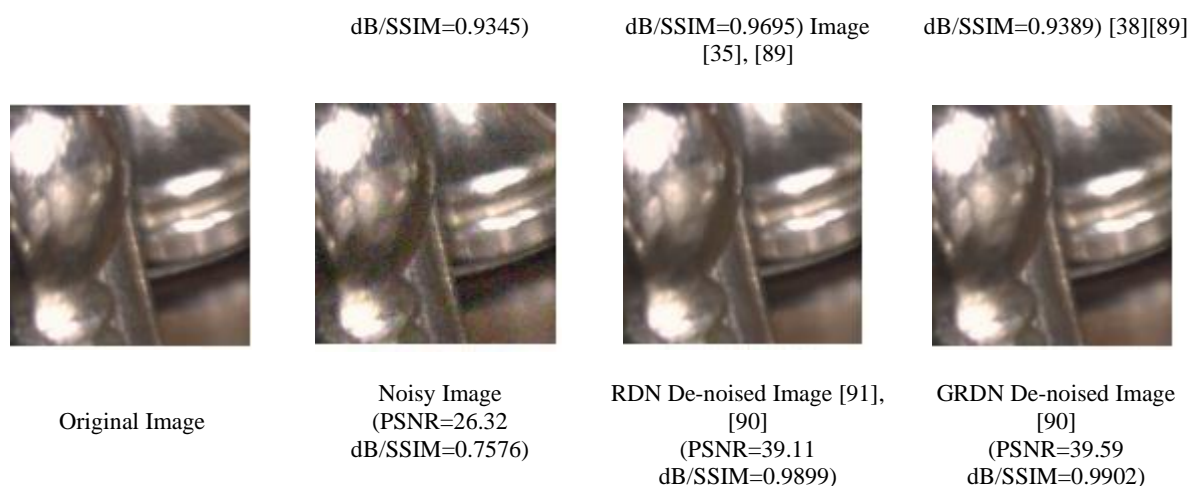
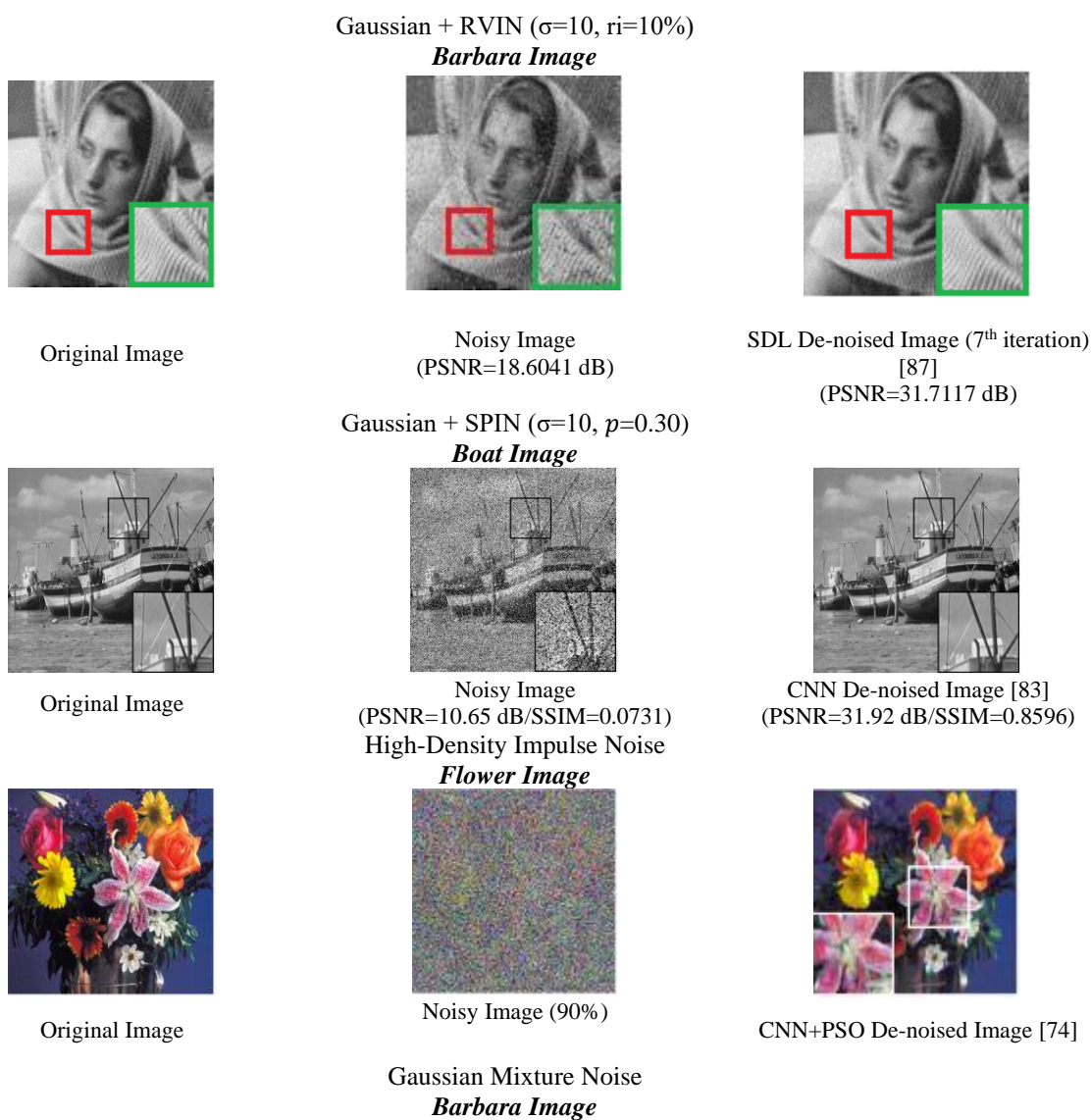


FIGURE 12: De-noising results for real-world noisy images



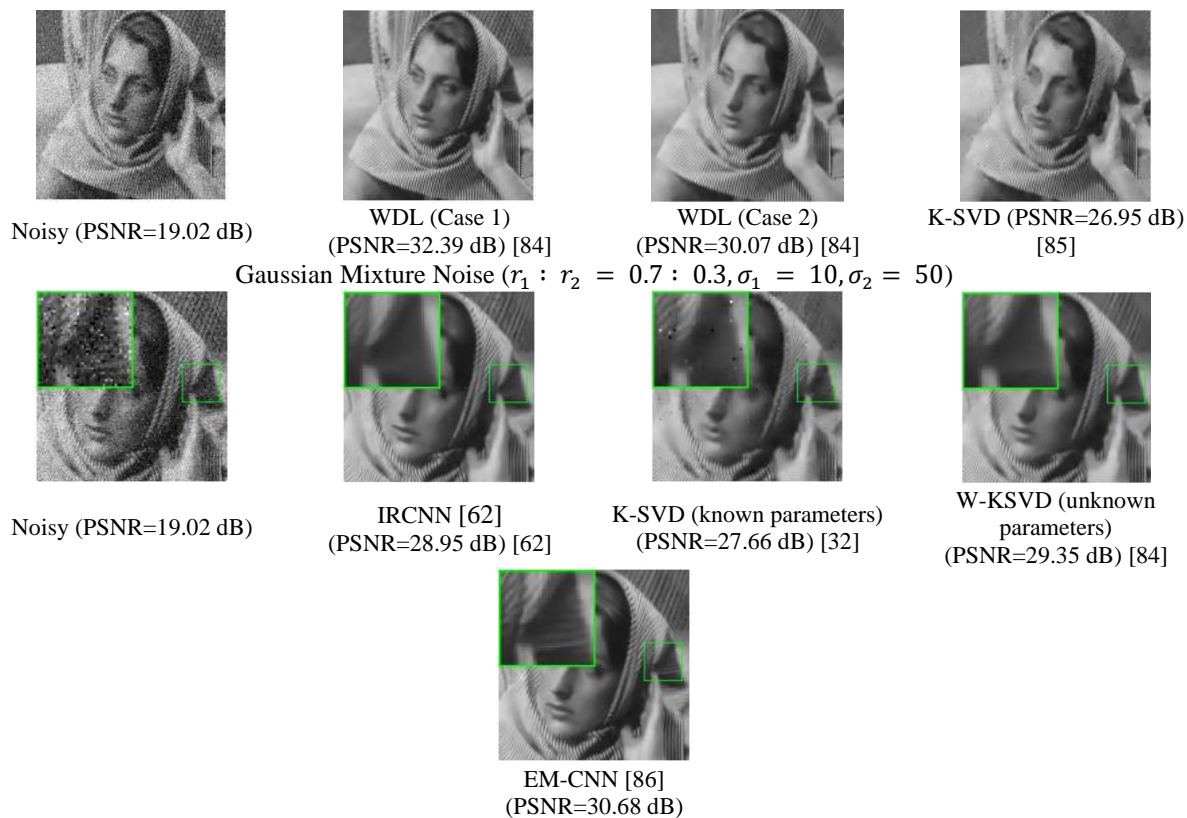


FIGURE 13: De-noising results for mixed noise

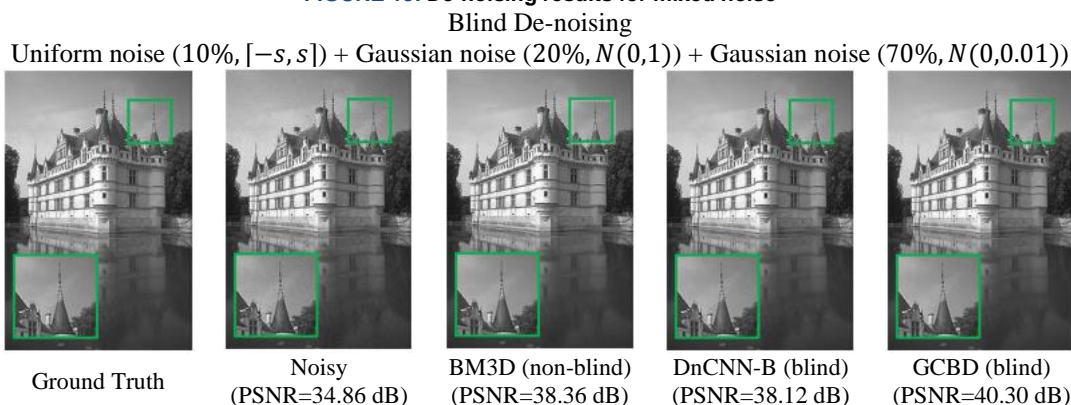


FIGURE 14: De-noising results for mixed noise on BSD68

professional camera) even under the same ISO values [93]. These problems make real-world image de-noising difficult and still a challenging task.

VII. BLIND IMAGE DE-NOISERS

The noise models are defined for particular noise type with known probability distribution function. For example, in case of Gaussian noise the standard deviation of the noisy image is known and the corresponding de-noised images are calculated. However, in real life scenario, the noise can be due to combined effect of various sources and noise modelling input parameters may not be well-defined. So, de-noisers which produce de-noised image even when noise level of input image is not defined are termed as blind de-noisers. The models are trained in such a way that it can incorporate wide range of unknown noise levels. The another approach is to estimate noise levels of the input

image, which does not produce accurate de-noised images due to inaccurate noise approximation. BM3D is the non-learning blind denoiser based on leveraging self-similarity by joint filter application on self-similarity image patches.

The DnCNN [38] model based on residual learning and batch normalization is the blind Gaussian denoiser in which single model is trained with varying noise levels from zero to fifty-five, down-sampled images with different upscaling factors and images (JPEG) with multiple quality factors. So, the same network can be used as blind Gaussian denoiser, JPEG image de-blocker and single-image super-resolution. Although FFDNet [56] improves DnCNN performance at noise level fifty and seventy-five on BSD-68 dataset, it is a non-blind de-noiser as it requires noise level map at the input. Similarly [60] involves two sub-networks which are trained separately based on noise-level

choice at inference time making it inappropriate for blind level denoising. Blind Universal Image Fusion Denoiser[95] is the network that extracts features to learn an image prior and intermediate noise level values, which is fed into the fusion part of model for final de-noising. The latest de-noisers such as ADNet[64], BRDNet[51], SCNN [49], PReLU [66], [78], [42] are designed for blind denoising. The blind de-noiser for mixed Gaussian impulse noise is also being designed [83]. The recent research trend in the field of computer vision is progressing towards the development of universal and blind de-noiser for real-world de-noising.

VII. DESCRIPTION OF DATASET AND SOFTWARE TOOLS

Software: The tremendous success of machine learning particularly deep learning is because of the parallel computing of GPU. TABLE XII describes popular machine-learning libraries used for various computer vision tasks.

TABLE XII
POPULAR SOFTWARE PACKAGES

Software Packages	Developer	Description
Caffe	Berkeley AI Research	C++, Python and Matlab interfaces Widely used for object detection tasks.
Matconvnet	Oxford Visual Geometry Group	Matlab interface, C++ compiler Pre-trained models for computer vision tasks.
Pytorch	Facebook AI Research Lab	Python interface, Open source software Python library for fast numerical computation
Theano	Montreal Institute of Learning Algorithms	Used for image de-noising, classification, super-resolution Faster than Theano
Tensorflow	Google Brain Team	compiler C++ and Python interfaces
Keras	ONEIROS project	Python interface Acts as interface for Tensorflow library
MXNet	Apache Software Foundation	Scalable, Flexible, Multiple programming languages

Datasets: The machine learning based methods have shown significant progress due to availability of open access benchmark datasets. The datasets are available for gray scale de-noising, color image de-noising, medical image de-noising and real-world de-noising. The training dataset is used for training the model, whereas testing dataset images are used to assess the de-noising results. The peak signal to noise ratio (PSNR) and structural similarity index (SSIM) are most commonly used image quality assessment metrics. However, there are many image quality assessment metrics which are given in [39]. The performance comparison of de-noisers can be done if they use common testing dataset. In case of Gaussian de-noisers, Set-12 dataset comprises of twelve scenes and BSD-68 dataset, i.e. Berkeley

Segmentation Dataset comprises of sixty eight natural images is commonly used. Kodak-24, LIVE and McMaster are also being used for synthetic de-noising. RENOIR, NAM, DND, SIDD and Xu are datasets for real world de-noising [89]. Some of the benchmark datasets are given in [96].

TABLE XIII
MODELS USED IN MULTI-DOMAIN IMAGE DE-NOISING

MODEL	SPECIALITY	NOISE SUPPRESSION IN VARIOUS MODELS
DnCNN [38]	Intuitively removes latent clean image in the hidden layers utilizing the residual learning strategy	1. Single DnCNN model for blind Gaussian denoising
		2. Single image super-resolution
		3. JPEG image deblocking
EM-CNN [86]	Uses an integration of CNN and EM-based mixed noise removal to give a variational method that can estimate the noise parameters iteratively to categorize noise types and levels in each pixel Uses maximum likelihood estimation framework and sparse representations over a trained dictionary and uses a self-determined weighting data fidelity function that detects noise in terms of different estimated noise parameters Uses variable splitting technique to bring strong image prior into model-based optimization methods and learned CNN de-noisers are used as modules in model-based optimization	1. Removal of Gaussian mixture noise
		2. Removal of Gaussian-Impulse noise
		1. Gaussian-Gaussian mixture
W-KSVD [84]	Constructs paired training data from the given noisy images, and then trains a deep denoising network for removing the noise; GAN is used to build the dataset Uses enhanced sparse representation in transform domain where enhancement of sparsity is achieved by grouping similar 2-D image fragments into	2. Impulse noise
		3. Gaussian-impulse noise
		4. Modified K-SVD for weighted rank-one approximation
IRCNN [62]		1. Gaussian noise (grey and color)
		2. Mixed Gaussian noise
		3. Various low-level vision applications
GCBD [42]		1. Blind Gaussian noise
		2. Blind Mixed noise
		1. Gaussian noise
BM3D [20]		2. Color image de-noising
		3. Mixed noise

VIII. RESULT AND DISCUSSION

Out of all machine learning methods, dictionary learning models performance is inferior in terms of PSNR. The disadvantages of dictionary learning are heuristic selection of the hyperparameters like sparsity level, number of atoms and iterations [97]. It fails to learn invariant features such as translational, rotation and scale invariance and it is apt for low dimensional signal only. The machine learning models have evolved from fully connected neural networks to CNN based de-noisers. CNN's have various advantages over fully connected neural networks such as multi-layer perceptron. The spatial information is intact in the case of CNN whose input is multi-dimensional image data. The parameters of CNN are reduced due to weight sharing as a fixed weight kernel is used. Therefore, reduction in number of learning parameters, translational invariance and locality due to convolutional operation has given edge to CNN over other fully-connected models [98]. Most of the CNN de-noisers require application oriented large datasets for supervised learning. The availability of medical image datasets is still challenging as it requires manual intervention for its annotation. Moreover, the de-noising results are almost stagnant after the network attains certain depth and there is no significant change by increasing the number of training images. The CNN methodologies involve change in activation function, network depth, loss function, training dataset, etc. In order to solve this problem there is a gradual shift from discriminative learning CNN model to generative learning model GAN model. It uses two neural networks generator and the discriminator, where generator model creates plausible images and discriminator model constantly evaluates the generator images as real or fake. Therefore, both networks work in synchronization and act as an adversarial for each other. The fundamental design of GAN is based on indirect training of the generator by the discriminator. This falls under the category of semi-supervised learning. The training efficiency of GAN is more than that of CNN as more features are learned in GAN in the same number of epochs as compared to CNN [99]. The GAN's achieve better results with less training images as compared to CNN.

TABLE II shows de-noising results in terms of PSNR for dictionary learning and CNN based networks. The models progressed from K-SVD, KLLD i.e., from dictionary learning models to CNN based models. The DnCNN model is the benchmark residual learning-based Gaussian de-noiser, which has led to the further development of many de-noisers. The methodologies involve a change in loss function, increase in receptive field size, the change in number of layers, integration of transform, spatial domain methods with CNN and inclusion of graph theory in CNN. It can be inferred from the TABLE II that PSNR values obtained by different CNN based methods are very close to each other. However, ADNet [64] network with four modules suppresses the effect of network length on shallow layers and gives good PSNR results on Set-12 dataset. In the CNN network, after an optimum number of layers,

PSNR values attain saturation. It implies that further increment in network length does not improve the de-noising performance. Apart from ADNet, BRDNet is the other network that integrates residual learning with batch renormalization and dilated convolutions to enhance de-noising performance. The de-noising performance of BRDNet can be attributed to an increase in receptive field size by dilated convolutions and an increase in network width by concatenation of two networks. Therefore it overcomes the disadvantages of the previous networks, such as (a) training difficulty and stagnation of results due to an increase in network length (b) mini-batch and internal co-variate shift problems. Further, PReLU [66] based edge aware filter has attained best PSNR results both on Set-12 and BSD-68 dataset at different sigma levels. It has used parameteric rectified linear units as activation, which overcomes the disadvantage of ReLU by learning in the negative direction. The success can be attributed to the fact that this is a hybrid methodology which has the inclusion of principal component analysis and edge aware bilateral filter. Moreover, CNN uses supervised learning which is becoming computationally demanding with an increase in dataset size. Therefore, the generative learning model of Generative Adversarial Network is being used. The GCBD model gives promising result even in the absence of supervised learning data. Its PSNR value is same as that of the DnCNN network on the BSD-68 dataset for noise level=15, and 25. TABLE III gives a comparative analysis of machine learning methods on the BSD-68 and Kodak-24 datasets. It has been observed that there is no significant difference in the PSNR values of different networks. The DIDN network designed with receptive field variation and modification of U-net architecture designed for semantic segmentation perform better for color images too as shown in TABLE IV. DRUNet [63] network which is based on deep learning CNN based image prior plugged into the half quadratic splitting-based de-noising iterative algorithm shows good results on both gray and color images.

The impulse de-noisers predict pixels affected by noise in the first step. It is followed by a noise contamination determiner and post noise detection processing. The dictionary learning and CNN-based models are designed for impulse noise removal. However, the noise ratios are varied in a very large range. To overcome the problem of less flexibility due to the unknown severity of contamination, [70] uses a noise ratio predictor that can measure the severity of corruption, i.e., the noise ratio of the image rapidly and efficiently. Fig. 11 (d) shows that blind CNN achieves a higher value of PSNR, compared to ANN [72]. Blind CNN removes the noise and retains image details, the reason being the NRP. It converts the noise mask into a noise ratio, and, according to this ratio, the most appropriate CNN model is selected for de-noising, rather than restoring image by removing the detected RVIN noise pixel-by-pixel.

The Poisson noise is modeled by its peak value and it is also categorized into dictionary learning models and CNN

based models. There is just one dictionary learning model which perform de-noising by greedy pursuit algorithm and boot strapping based stopping criterion. Gaussian de-noisers such as DnCNN and IRCNN are also being used for Poisson de-noising with different parameter settings. Although Gaussian de-noisers are being used for the poisson noise, the heuristic setting of network parameters is again a big challenge. TABLE V gives poisson de-noising performance on Live1 dataset. DenoiseNet is the first residual learning based CNN de-noiser designed for poisson noise. Later, CNN+LSTM and MC2RNet models have outperformed DenoiseNet. The CNN + LSTM Poisson de-noiser, which uses CNN for feature extraction and LSTM layers to store noise components, outperforms DenoiseNet as given in TABLE V and VI. The inclusion of Blahut-Arimoto algorithm to determine number of CNN layers and learning of residual noise statistics by LSTM improves the de-noising results of CNN+LSTM. Deep multi-scale cross-path concatenation residual network (MC2RNet) which incorporates cross-path concatenation modules for de-noising also outperforms CNN based DenoiseNet as given in TABLE VII and TABLE VIII on Set 10 and BSD-68 dataset respectively. Therefore, CNN+LSTM, DenoiseNet and MC2RNet are the available CNN based Poisson de-noisers which are less in number as compared to Gaussian de-noisers.

The mixed noise can be modeled mathematically in different ways. There are models designed for mixture of impulse and Gaussian noise. The four-stage residual learning-based mixed network [82] and de-noiser with two-stage cascade connection of impulse and Gaussian de-noiser are used for mixed Gaussian Impulse noise given in TABLE IX. TABLE X discusses mixed Gaussian noise. And, TABLE XI discusses blind de-noising. Figs. 10 to 14 depict qualitative results of different images with different noise types, i.e., Gaussian noise, impulse noise, mixed noise, real-world, and blind noise. TABLE XIII discusses the methods that extend into multiple domains of image de-noising.

IX. CONCLUSION AND FUTURE SCOPE

In this paper, comprehensive study and analysis of machine learning models for removal of different noises is provided. The categorization of different de-noisers is done into dictionary learning models such as CNN based models and GAN based models. The comparative analysis PSNR results of different de-noisers on some benchmark datasets are provided for better understanding of reader. It has been observed that integration of analytical methods in machine learning model can further improve the results. Although there are numerous networks designed for synthetic datasets, but real-world image de-noising is still a challenging problem. The GAN based de-noisers are still in primitive stage. However, the generative learning based GAN and deep belief networks can perform unsupervised learning to certain extent unlike CNN. The future prospects lie in design of real-world de-noisers with unsupervised learning framework for practical applications. The transfer

learning approach, graph theory inclusion in neural network, prior design, and receptive field enhancement are some of the areas for future research.

References

- [1] W. Meinel, J. C. Olivo-Marin, and E. D. Angelini, "Denoising of microscopy images: A review of the state-of-the-art, and a new sparsity-based method," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3842–3856, Aug. 2018, doi: 10.1109/TIP.2018.2819821.
- [2] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: From classical to state-of-the-art approaches," *Inf. Fusion*, vol. 55, pp. 220–244, Mar. 2020, doi: 10.1016/j.inffus.2019.09.003.
- [3] Y. E. Gökdağ, F. Şansal, and Y. D. Gökdel, "Image denoising using 2-D wavelet algorithm for Gaussian-corrupted confocal microscopy images," *Biomed. Signal Process. Control*, vol. 54, p. 101594, Sep. 2019, doi: 10.1016/j.bspc.2019.101594.
- [4] R. Gonzalez and R. Wood, *Digital Image Processing*, 3rd ed., London, U.K., Pearson Education, 2008.
- [5] S. W. Hasinoff, "Photon, Poisson Noise," in *Computer Vision*, Springer US, pp. 608–610, 2014, doi: https://doi.org/10.1007/978-0-387-31439-6_482.
- [6] I. Rodrigues, J. Sanches, and J. Bioucas-Dias, "Denoising of medical images corrupted by poisson noise," in *Proceedings - International Conference on Image Processing, ICIP*, 2008, pp. 1756–1759, doi: 10.1109/ICIP.2008.4712115.
- [7] H. Yu, M. Ding, X. Zhang, and J. Wu, "PCANet based nonlocal means method for speckle noise removal in ultrasound images," *PLoS One*, vol. 13, no. 10, pp. e0205390, 2018, doi: 10.1371/journal.pone.0205390.
- [8] E. E. Kuruoğlu and J. Zerubia, "Modeling SAR images with a generalization of the Rayleigh distribution," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 527–533, Apr. 2004, doi: 10.1109/TIP.2003.818017.
- [9] G. Kim, J. Cho, and M. Kang, "Cauchy Noise Removal by Weighted Nuclear Norm Minimization," *J. Sci. Comput.*, vol. 83, no. 1, pp. 1–21, Apr. 2020, doi: 10.1007/s10915-020-01203-2.
- [10] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "A multichannel order-statistic technique for cDNA microarray image processing," *IEEE Trans. Nanobioscience*, vol. 3, no. 4, pp. 272–285, Dec. 2004, doi: 10.1109/TNB.2004.837907.
- [11] Y. Norose, K. Mizutani, N. Wakatsuki, and T. Ebihara, "Noise reduction in ultrasonic computerized tomography by preprocessing for projection data," *Jpn. J. Appl. Phys.*, vol. 54, no. 7, p. 07HC12, Jul. 2015, doi: 10.7567/JJAP.54.07HC12.
- [12] M. Ye and Y. Qian, "Mixed Poisson-Gaussian noise model based sparse denoising for hyperspectral imagery," in *Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing*, 2012, pp. 1–4, doi: 10.1109/WHISPERS.2012.6874280.
- [13] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Vis. Comput. Ind. Biomed. Art*, vol. 2, no. 7, 2019, doi: 10.1186/s42492-019-0016-7.
- [14] M. Mafi, H. Martin, M. Cabrerizo, J. Andrian, A. Barreto, and M. Adjouadi, "A comprehensive survey on impulse and Gaussian denoising filters for digital images," *Signal Processing*, vol. 157, Elsevier B.V., pp. 236–260, 01-Apr-2019, doi: 10.1016/j.sigpro.2018.12.006.
- [15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE International Conference on Computer Vision*, 1998, pp. 839–846, doi: 10.1109/iccv.1998.710815.
- [16] R. G. Gavaskar and K. N. Chaudhury, "Fast Adaptive Bilateral Filtering," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 779–790, Feb. 2019, doi: 10.1109/TIP.2018.2871597.
- [17] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, pp. 1531–1549, 2010, doi: 10.1016/j.patcog.2009.09.023.

- [18] R. Guhathakurta, "Denoising of image: A wavelet based approach," in *2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017*, 2017, pp. 194–197, doi: 10.1109/IEMECON.2017.8079587.
- [19] S. D. Ruikar and D. D. Doye, "Wavelet Based Image Denoising Technique," *International Journal of Advanced Computer Science and Application*, vol. 2, no. 3, pp. 49–53, 2011, doi: 10.14569/IJACSA.2011.020309.
- [20] 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, doi: 10.1109/TIP.2007.901238.
- [21] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, and K. E. Bm3d, "BM3D Image Denoising with Shape-Adaptive Principal Component Analysis," in *Signal Processing with Adaptive Sparse Structured Representations*, 2009, p. 00369582.
- [22] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, 2013, doi: 10.1109/TIP.2012.2210725.
- [23] Jean-Luc Starck, E. J. Candes and D. L. Donoho, "The curvelet transform for image denoising," in *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, June 2002, doi: 10.1109/TIP.2002.1014998.
- [24] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005, doi: 10.1109/TIP.2005.859376.
- [25] S. Schulte, M. Nachtgaele, V. DeWitte, D. Van der Weken, and E. E. Kerre, "A fuzzy impulse noise detection and reduction method," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1153–1162, May 2006, doi: 10.1109/TIP.2005.864179.
- [26] J. H. Wang, W. J. Liu, and L. Da Lin, "Histogram-based fuzzy filter for image restoration," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 2, pp. 230–238, Apr-2002, doi: 10.1109/3477.990880.
- [27] M. Pathak and G. R. Sinha, "A Survey of Fuzzy Based Image Denoising Techniques," *IOSR J. Electron. Commun. Eng.*, vol. 9, no. 4, pp. 27–36, 2014.
- [28] P. Rodriguez, "Total variation regularization algorithms for images corrupted with different noise models: A review," *Journal of Electrical and Computer Engineering*, 2013, doi: 10.1155/2013/217021.
- [29] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869, doi: 10.1109/CVPR.2014.366.
- [30] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006, doi: 10.1109/TIT.2005.860430.
- [31] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, "Review of noise removal techniques in ECG signals," *IET Signal Process.*, vol. 14, no. 9, pp. 569–590, Dec. 2020, doi: 10.1049/iet-spr.2020.0104.
- [32] 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, doi: 10.1109/TSP.2006.881199.
- [33] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," *Cs Tech.*, vol. 40, no. 8, pp. 1–15, 2008.
- [34] L. Ma, L. Moisan, J. Yu, and T. Zeng, "A dictionary learning approach for Poisson image Deblurring," *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1277–1289, 2013, doi: 10.1109/TMI.2013.2255883.
- [35] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2392–2399, doi: 10.1109/CVPR.2012.6247952.
- [36] X. Tang, L. Zhang, and X. Ding, "SAR image despeckling with a multilayer perceptron neural network," *International Journal of Digital Earth*, vol. 12, no. 3, Taylor and Francis Ltd., pp. 354–374, 04-Mar-2019, doi: 10.1080/17538947.2018.1447032.
- [37] Y. Chen and T. Pock, "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017, doi: 10.1109/TPAMI.2016.2596743.
- [38] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," in *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017, doi: 10.1109/TIP.2017.2662206.
- [39] R. S. Thakur, R. N. Yadav, and L. Gupta, "State-of-art analysis of image denoising methods using convolutional neural networks," *IET Image Process.*, vol. 13, no. 13, pp. 2367–2380, 2019, doi: 10.1049/iet-ipr.2019.0157.
- [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, 2018, doi: 10.1049/iet-ipr.2017.0389.
- [41] I. J. Goodfellow, J. P. Abadie, M. Mirza *et al.*, "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing System*, 2014, pp. 2672–2680, doi: 10.5555/2969033.2969125.
- [42] J. Chen, J. Chen, H. Chao, and M. Yang, "Image Blind Denoising with Generative Adversarial Network Based Noise Modeling," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3155–3164, doi: 10.1109/CVPR.2018.00333.
- [43] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, 2009, doi: 10.1109/TIP.2009.2018575.
- [44] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, 2013, doi: 10.1109/TIP.2013.2277813.
- [45] T. Xia, H. Dong, X. Wang, X. Yu, Y. Liu, and X. Li, "Image denoising based on dictionary learning of mean corrected atoms," in *Proceedings - 18th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2019*, 2019, pp. 202–207, doi: 10.1109/ICIS46139.2019.8940197.
- [46] A. Kumar and S. Kataria, "Dictionary Learning Based Applications in Image Processing using Convex Optimisation," in *International Conference on Signal Processing and Integrated Networks (SPIN)*, 2017.
- [47] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781, doi: 10.1109/CVPR.2014.349.
- [48] W. Bae, J. Yoo, and J. C. Ye, "Beyond Deep Residual Learning for Image Restoration: Persistent Homology-Guided Manifold Simplification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017, vol. 2017-July, pp. 1141–1149, doi: 10.1109/CVPRW.2017.152.
- [49] K. Isogawa, T. Ida, T. Shiadera, 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, doi: 10.1109/LSP.2017.2782270.
- [50] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, and N. Luo, "Enhanced CNN for image denoising," *CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 17–23, Mar. 2019, doi: 10.1049/trit.2018.1054.
- [51] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, Jan. 2020, doi: 10.1016/j.neunet.2019.08.022.
- [52] B. Park, S. Yu, and J. Jeong, "Densely connected hierarchical network for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019, vol. 2019-June, pp. 2104–2113, doi: 10.1109/CVPRW.2019.00263.

- [53] P. Liu, H. Zhang, W. Lian, and W. Zuo, "Multi-Level Wavelet Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 74973–74985, 2019, doi: 10.1109/ACCESS.2019.2921451.
- [54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
- [55] S. Yu, B. Park, and J. Jeong, "Deep iterative down-up CNN for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019, vol. 2019-June, pp. 2095–2103, doi: 10.1109/CVPRW.2019.00262.
- [56] 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, 2018, doi: 10.1109/TIP.2018.2839891.
- [57] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [58] C. Cruz, A. Foi, V. Katkovnik, and K. Egiazarian, "Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1216–1220, 2018, doi: 10.1109/LSP.2018.2850222.
- [59] D. Valsesia, G. Fracastoro, and E. Magli, "Image Denoising with Graph-Convolutional Neural Networks," in *Proceedings - International Conference on Image Processing, ICIP*, 2019, vol. 2019-September, pp. 2399–2403, doi: 10.1109/ICIP.2019.8803367.
- [60] S. Lefkimmiatis, "Universal Denoising Networks: A Novel CNN Architecture for Image Denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3204–3213, doi: 10.1109/CVPR.2018.00338.
- [61] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising Prior Driven Deep Neural Network for Image Restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, 2019, doi: 10.1109/TPAMI.2018.2873610.
- [62] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January, pp. 2808–2817, doi: 10.1109/CVPR.2017.300.
- [63] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-Play Image Restoration with Deep Denoiser Prior," *arXiv*, 31-Aug-2020. [Online]. Available: <http://arxiv.org/abs/2008.13751>.
- [64] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Networks*, vol. 124, pp. 117–129, Apr. 2020, doi: 10.1016/j.neunet.2019.12.024.
- [65] R. Couturier, G. Perrot, and M. Salomon, "Image denoising using a deep encoder-decoder network with skip connections," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11306 LNCS, pp. 554–565, doi: 10.1007/978-3-030-04224-0_48.
- [66] R. S. Thakur, R. N. Yadav, and L. Gupta, "PReLU and edge-aware filter-based image denoiser using convolutional neural network," *IET Image Process.*, vol. 14, no. 15, pp. 3869–3879, Dec. 2020, doi: 10.1049/iet-ipr.2020.0717.
- [67] Y. Zhong, L. Liu, D. Zhao, and H. Li, "A generative adversarial network for image denoising," *Multimed. Tools Appl.*, vol. 79, no. 23–24, pp. 16517–16529, Jun. 2020, doi: 10.1007/s11042-019-7556-x.
- [68] S. Wang *et al.*, "Dictionary learning based impulse noise removal via L1-L1 minimization," *Signal Processing*, vol. 93, no. 9, pp. 2696–2708, 2013, doi: 10.1016/j.sigpro.2013.03.005.
- [69] D. Guo, Z. Tu, J. Wang, M. Xiao, X. Du, and X. Qu, "Salt and Pepper Noise Removal with Multi-Class Dictionary Learning and L0 Norm Regularizations," *Algorithms*, vol. 12, no. 1, p. 7, Dec. 2018, doi: 10.3390/a12010007.
- [70] B. Deka and P. K. Bora, "Removal of random-valued impulse noise using sparse representation," in *2011 National Conference on Communications, NCC 2011*, 2011, doi: 10.1109/NCC.2011.5734762.
- [71] J. Chen, G. Zhang, S. Xu, and H. Yu, "A blind CNN denoising model for random-valued impulse noise," *IEEE Access*, vol. 7, pp. 124647–124661, 2019, doi: 10.1109/ACCESS.2019.2938799.
- [72] I. Turkmen, "The ANN based detector to remove random-valued impulse noise in images," *J. Vis. Commun. Image Represent.*, vol. 34, no. October, pp. 28–36, 2016, doi: 10.1016/j.jvcir.2015.10.011.
- [73] G. Li, X. Xu, M. Zhang, and Q. Liu, "Densely connected network for impulse noise removal," *Pattern Anal. Appl.*, vol. 23, no. 3, pp. 1263–1275, Aug. 2020, doi: 10.1007/s10044-020-00871-y.
- [74] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "High-density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation," *IET Image Process.*, vol. 13, no. 2, pp. 365–374, Feb. 2019, doi: 10.1049/iet-ipr.2018.5776.
- [75] L. Jin, W. Zhang, G. Ma, and E. Song, "Learning deep CNNs for impulse noise removal in images," *J. Vis. Commun. Image Represent.*, vol. 62, pp. 193–205, 2019, doi: 10.1016/j.jvcir.2019.05.005.
- [76] R. Giryes and M. Elad, "Sparsity-based poisson denoising with dictionary learning," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5057–5069, Dec. 2014, doi: 10.1109/TIP.2014.2362057.
- [77] W. Kumwilaisak, T. Piriyatharawat, P. Lasang, and N. Thatphithakkul, "Image Denoising with Deep Convolutional Neural and Multi-Directional Long Short-Term Memory Networks under Poisson Noise Environments," *IEEE Access*, vol. 8, pp. 86998–87010, 2020, doi: 10.1109/ACCESS.2020.2991988.
- [78] Y. Su, Q. Lian, X. Zhang, B. Shi, and X. Fan, "Multi-scale Cross-path Concatenation Residual Network for Poisson denoising," *IET Image Process.*, vol. 13, no. 8, pp. 1295–1303, Jun. 2019, doi: 10.1049/iet-ipr.2018.5941.
- [79] R. Tal, O. Litany, R. Giryes, and A. M. Bronstein, "[1701.01687] Deep Convolutional Denoising of Low-Light Images," <https://arxiv.org/abs/1701.01687>, 2017. [Online]. Available: <https://arxiv.org/abs/1701.01687>.
- [80] 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, doi: 10.1109/TIP.2018.2859044.
- [81] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proceedings - International Conference on Image Processing, ICIP*, 2006, vol. 1, doi: 10.1109/ICIP.2007.4378954.
- [82] M. T. Islam, S. M. Mahbubur Rahman, M. Omair Ahmad, and M. N. S. Swamy, "Mixed Gaussian-impulse noise reduction from images using convolutional neural network," *Signal Process. Image Commun.*, vol. 68, no. June, pp. 26–41, 2018, doi: 10.1016/j.image.2018.06.016.
- [83] R. Abiko and M. Ikehara, "Blind Denoising of Mixed Gaussian-impulse Noise by Single CNN," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019, vol. 2019-May, pp. 1717–1721, doi: 10.1109/ICASSP.2019.8683878.
- [84] J. Liu, X. C. Tai, H. Huang, and Z. Huan, "A weighted dictionary learning model for denoising images corrupted by mixed noise," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1108–1120, 2013, doi: 10.1109/TIP.2012.2227766.
- [85] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006, doi: 10.1109/TIP.2006.881969.
- [86] 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, 2020, doi: 10.1109/TIP.2019.2940496.
- [87] H. Zhu and M. K. Ng, “Structured Dictionary Learning for Image Denoising under Mixed Gaussian and Impulse Noise,” *IEEE Trans. Image Process.*, vol. 29, pp. 6680–6693, 2020, doi: 10.1109/TIP.2020.2992895.
- [88] M. Mafi, W. Izquierdo, H. Martin, M. Cabrerizo, and M. Adjouadi, “Deep convolutional neural network for mixed random impulse and gaussian noise reduction in digital images,” *IET Image Process.*, vol. 14, no. 15, pp. 3791–3801, Dec. 2020, doi: 10.1049/iet-ipr.2019.0931.
- [89] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, “[1804.02603] Real-world Noisy Image Denoising: A New Benchmark.” [Online]. Available: <https://arxiv.org/abs/1804.02603>. [Accessed: 30-Dec-2020].
- [90] D. W. Kim, J. R. Chung, and S. W. Jung, “GRDN: Grouped residual dense network for real image denoising and GAN-based real-world noise modeling,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019, vol. 2019-June, pp. 2086–2094, doi: 10.1109/CVPRW.2019.00261.
- [91] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image restoration,” *IEEE Trans. Pattern Anal.*, Dec. 2018, doi: 10.1109/tpami.2020.2968521.
- [92] K. Lin, T. H. Li, S. Liu, and G. Li, “Real photographs denoising with noise domain adaptation and attentive generative adversarial network,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2019-June, pp. 1717–1721, 2019, doi: 10.1109/CVPRW.2019.00221.
- [93] C. Chen, Z. Xiong, X. Tian, Z. J. Zha, and F. Wu, “Real-World Image Denoising with Deep Boosting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 12, pp. 3071–3087, Dec. 2020, doi: 10.1109/TPAMI.2019.2921548.
- [94] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, “A Holistic Approach to Cross-Channel Image Noise Modeling and Its Application to Image Denoising,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, pp. 1683–1691, doi: 10.1109/CVPR.2016.186.
- [95] 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, doi: 10.1109/TIP.2020.2976814.
- [96] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C. W. Lin, “Deep learning on image denoising: An overview,” *Neural Networks*, vol. 131, pp. 251–275, 2020, doi: 10.1016/j.neunet.2020.07.025.
- [97] T. Tong, J. Caballero, K. Bhatia, and D. Rueckert, “Dictionary learning for medical image denoising, reconstruction, and segmentation,” in *Machine Learning and Medical Imaging*, Elsevier Inc., 2016, pp. 153–181.
- [98] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, “Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods,” *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 20–36, Jan. 2018, doi: 10.1109/MSP.2017.2760358.
- [99] S. Barua, S. M. Erfani, and J. Bailey, “FCC-GAN: A Fully Connected and Convolutional Net Architecture for GANs,” *arXiv*, May 2019.