SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 21, No. 3, October 2024, 429-466
UDC: 004.932.4 **DOI: https://doi.org/10.2298/SJEE2403429K**

Salt and Pepper Denoising Filters for Digital Images: A Technical Review

Abhishek Kumar[1], Sanjeev Kumar[1], Asutosh Kar[2]

Abstract: Noise in images refers to random variations in pixel intensities that alter the original pixel intensities of the image. Among the various noises present in the image, salt and pepper noise corrupts images due to a defect in the device's hardware or the camera's faulty sensor. This leads to misinterpretation of pixels and deterioration of image quality during visualization of natural images and diagnosis of medical images. Up until now, researchers have presented several cutting-edge filters to overcome and lessen the impact of this noise. This article presents a comprehensive investigation into three different domains of impulse denoising of digital images. These domains are based on the spatial domain, the fuzzy logic domain, and the deep learning-based category. In this study, many techniques of image denoising were categorized and analyzed, along with their respective motivations, principles of execution, and comparative analysis. We carefully explain and implement a few significant approaches, considered state-of-the-art in each subject, in MATLAB. When doing simulations, the filters are analyzed and quantitatively evaluated using three metrics that are frequently utilized. These parameters are the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM). Finally, we provide a comparison of each study category to enhance our comprehension of each domain. We conclude by outlining the challenges each domain poses and providing a detailed explanation of the rationale for future research.

Keywords: Salt and pepper noise, Non-linear filter, Fuzzy logic, Convolution neural networks, Peak signal-to-noise ratio, Structural similarity index measure.

1 Introduction

Digital images have become increasingly popular in image photography, medical, remote sensing, criminal investigation, banking and finance, security, and more in recent decades. These fields need accurate and appealing graphics to be successful and applicable. However, environmental noise, transmission channels, capture mediums like camera sensors, analog to digital conversion, and

[1]Department of Electronics and Communication Engineering, Sarala Birla University, Ranchi, Jharkhand, India; E-mails: a.kumar1049@gmail.com; sanjeevsingh.ece@gmail.com
[2]Department of Electronics and Communication Engineering, Dr. B R Ambedkar National Institute of Technology Jalandhar, Punjab-144008, India; E-mail: kara@nitj.ac.in

other factors impair visual image quality, necessitating appropriate processing [1 – 3]. Edge blurring, image texture preservation, artifacts, misreading of pixels as noise, etc., during image analysis and processing require denoising to obtain high-quality images that challenge the community [4].

The denoising scheme is an analytically inverse issue, hence it rarely has a unique solution, leaving a gap in this sector. Finding an image-denoising strategy that reduces noise without compromising image quality is difficult [5]. Images are mostly contaminated by impulsive, Gaussian, or Speckle noise. Timing mistakes, such as camera sensor pixel element failure, erroneous memory locations, etc., create impulse noise (IN) during digitization. Fixed value IN (FVIN) values of 0 (lowest for pepper noise) or 255 (maximum for salt noise) give an 8-bit image a "salt and pepper (SPN)" appearance. Random valued IN (RVIN) ranges from 0 to 255.

Regarding the mathematical notation of SPN, the 8-bit gray level original image $X = [x_{i,j}]$ of dimension $M \times N$ and SPN corrupted image $Y = [y_{i,j}]$ are represented respectively, where $x_{i,j}$ and $y_{i,j}$ represents the gray value of the pixel values at the location (i, j) of X and Y respectively and $(i, j) \in \{1, 2, \ldots, M\} \times \{1, 2, \ldots N\}$

The pixel values $y_{i,j}$ of SPN noise corrupted image Y is defined by:

$$y_{i,j} = \begin{cases} x_{i,j} & \text{with probablity } \alpha; \\ v_{i,j} & \text{with probability } 1-\alpha, \end{cases} \tag{1}$$

where $v_{i,j}$ is the identically distributed random process with α as the probability density.

In various impulsive noise models, $v_{i,j}$ is replaced by $v_{\min}$ and $v_{\max}$ which is the dynamic range of the image.

Most researchers remove SPN noise from images by considering it minimum and maximum pixels. But SPN noise detection may have a chance of mistake when 0 or 255-pixel image information is used. Due to its inverse nature, image denoising remains a difficult unresolved subject despite previous research. Fig. 1 classifies image-denoising algorithms in this review work as spatial domain, fuzzy logic, and deep learning. In spatial domain filtering, pixels are directly filtered. Linear and non-linear filters exist. Image pixel values and fuzzy logic are used in the procedure. Designing the fuzzy membership function with fuzzy rules. Today, the most prominent deep learning methods and their subsets are employed extensively in machine learning.

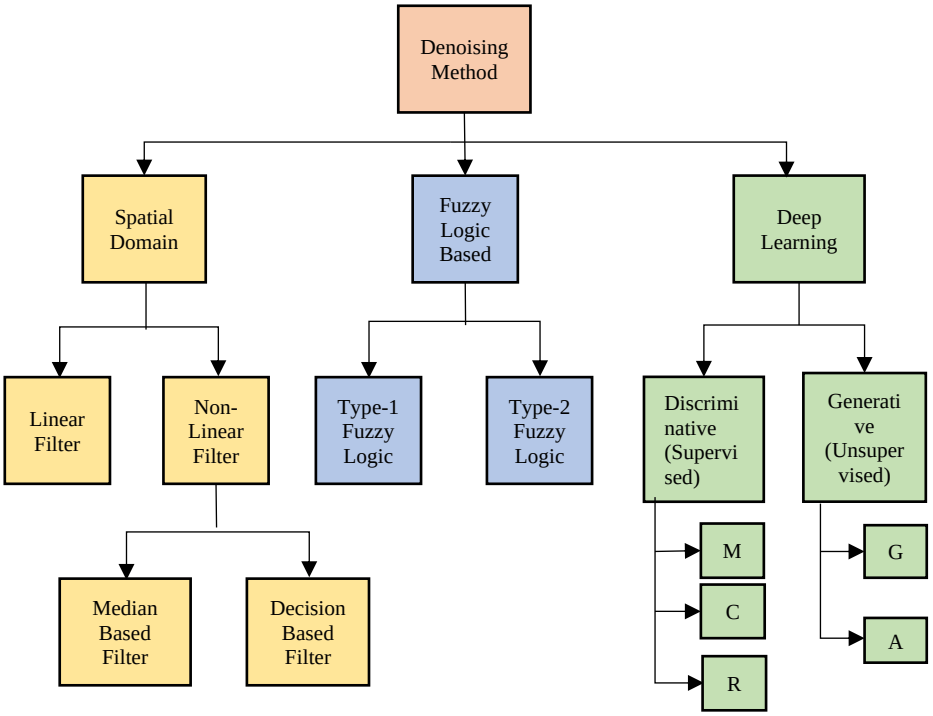


Fig. 1 – *Denoising techniques of Impulse noise. M: Multilayer Perceptron, CNN: Convolutional Neural Network, R: Recursive Neural Network, G: Generative Adversarial Network, A: Auto encoder.*

The main image processing filters are linear, nonlinear, or adaptive. To decrease noise, the linear filter (LF) matches output pixels to neighbors, whereas the nonlinear filter (NLF) preserves edge data. LFs without edge data are less suitable for image processing. Fuzzy logic is famous for handling uncertainties, therefore applying it as an extension of classical filters has helped remove image noise. When applying fuzzy logic to an application, choose the fuzzy rule foundation and inference mechanism MFs. To see humans as physical traits, the human perspective establishes these criteria and restrictions. Fuzzy logic solves difficult data transformations. In a noisy image, the fuzzy knowledge base can determine the noise level of pixels and assign a noise level (between 0 and 1) to each pixel of famous images [6 – 7]. Several academics have used fuzzy logic and optimization methods to train image denoising data for better analysis [8].

Machine learning (ML) has been popular in research and applications such multimedia concept retrieval, text mining, spam detection, video recommendation, and image categorization. These apps use several machine learning (ML) methods, including deep learning (DL). Convolutional neural networks (CNNs) are popular DL networks. CNN boosted DL's popularity.

Recently, convolution neural networks (CNNs) have been popular because they can train deep neural networks as IN detectors to discover noisy pixels in images [9]. Together with the mean, median, edge-preserving filtering, etc., these networks can provide high-quality spatial images to find patterns. A residual learning-based deep convolution architecture learns an end-to-end map from noisy to noise-free images. The networks worked well in blind denoising, authentic noisy images, and more. Other CNN studies utilizing Bayesian-optimized SVM, nonconvex variational model, and simultaneous multi-scale extraction [78 – 80] are widely employed. Several papers shed light on image-denoising technologies for diverse applications. However, there remains scope for further investigation, which has been attempted here.

Our contribution is summarized as:

– Analyzing of different impulse image denoising methods.
– Comparing the commonly used methods with their principle of operation and evaluation methods.
– Simulation of different image noise filtering methods and comparing their results.
– Discussion on the various methods with their advantage and limitations.
– Potential challenges and road maps for image denoising.
– Road map to future research and scopes in this field.

The rest of the paper is sectioned as follows. Section 2 discusses various types of spatial domain filters. Section 3 presents the fuzzy-based filter. Section 4 describes CNN-based filters with some model architecture. Section 5 provides the details of the simulation setup and analyzes the result of the discussed filters. Section 6 briefly analysis the discussed method and future work to be carried and lastly section 7 concludes the work.

2 Spatial Domain Filter

2.1 Linear and non-linear filter

Spatial domain filtering is primarily used to directly modify pixel values linearly or nonlinearly. Commonly used LFs are mean, Gaussian, and Weiner filters, which perform well for additive noise but lose edges and smoothing. NLFs are preferred over LFs because their output is a nonlinear function of the input. Subcategories include median and switching filters. The median filter (MeF) in [10] uniformly moves the processing filtering window over the noisy image, restoring it well. It denoises and preserves edges well at low noise concentrations but not at high noise densities. A MeF with good de-noising restoration and edge preservation at low noise levels was proposed in [11]. MF modifications such weighted MeF (WMeF) [12] and center-weighted MeF (CWMeF] [13] have been successful in this direction. Weighting improves the window's center pixel proportion and lowers smoothness; however, noise can affect it. WMeF weights

filtering window pixels and usually contributes to the central pixel. Central pixels have greater weights in CWMeF, reducing smoothing. The repeated pixel substitutions in the MeF cause streaking at greater noise densities. MeF allows each pixel, noisy or not, to match the median pixel in the filter window, keeping filtered images fuzzy. Rank-order-based adaptive MeF (RAMeF) with variable window size has been proposed based on two-step testing to address this issue. It can eliminate confused impulses more often while preserving sharpness. The first step detects residual impulses in the MeF output, while the second stage checks for corrupted center pixels with IN. An updated MeF utilizes a min-max algorithm to pre-label and eliminate noise from input pixels [14]. MeF denoising and noise prevention are applied to the cumulative output, although performance is poor at high noise density.

2.2 Switching-based filters

Every pixel, damaged or not, is median-operated using traditional MeF algorithms. Image quality suffers and critical information is lost. Adding an IN detection mechanism before filtering fixes it. Standard median (SM) filters consider just defective pixels. Progressive switching median (PSM) filters apply impulse detector and noise filter sequentially [15 – 16]. Next iteration pixels benefit from noisy pixels. Large noise blotches' noisy pixels are identified better than a MeF. Since the filter skips many impulses and affects excellent pixels, visual blurring ensues. At high noise levels, SM filters misclassify pixels. The noise adaptive soft switching median (NASM) filter uses a three-level ranking soft switching noise detection method to overcome these issues [17]. At low noise concentrations, it removes IN better while preserving signal characteristics. Additionally, the MeF cannot distinguish narrow lines from impulses, eliminating them from the image. Zhang et al. [18] proposed a simple impulse detector for SM filters that convolves the input image with four one-dimensional Laplacian operator kernels for better filtering. These operators calculate the least absolute value of these convolutions to detect impulses and are sensitive to edges in all orientations. Restoring noisy images with information from all images makes the local mean filter more robust. The weighted average (WA) of all image pixels is based on neighborhood similarity. SM with boundary discriminative noise detection (BDND), a complex IN detection method, divides pixels in a local window centered on the current pixel into three groups [19]. Both low- and high-intensity IN and uncorrupted pixels can identify corrupted pixels. Two boundaries are needed to classify pixels with high precision, zero miss detection, and low false alarm. The complexity of selecting a threshold limits these SM filters. Thus, replacing a noisy pixel with its neighboring median without considering local features like edges is problematic. Srinivasan et al. [20] have developed a decision-based algorithm (DBA) filter to replace the corrupted pixels either by the low noise density median pixels or by high noise density neighbourhood pixels to achieve higher correlation for better edge preservation. Further, the DBA filter shows consistent and stable performance across a wide

range of noise densities at low computation time, leading to a streaking effect due to repeated replacements of neighbouring pixels. In another attempt, Ibrahim et al. [21] have hybridized the adaptive and switching MeF which has achieved image denoising up to 95% while remaining faster and adaptable to the local noise level. Similarly, a switching-based adaptive weighted mean (SAWM) filter that combines the directional difference-based noise detector with the adaptive weighted mean filter has removed the IN with better restoration and high computational efficiency [22]. It computes the lowest absolute value of the four mean differences between the current pixel and its neighbours in four directional windows with predetermined thresholds, thus eliminating noisy pixels. It uses an adaptive WA filter to estimate every noisy pixel with a WA of its noiseless neighbours in the filter window. An edge-preserving filter (EPF) has been investigated which employs an efficient IN detector to detect noisy pixels [23]. The filter reconstructs the noisy pixels with a standard MeF and uses a directional correlation-dependent filtering technique based on observing the correlation of samples in six different directions. An improved high-efficiency IN detection (HEND) mechanism based on the BDND detector that adopts a coarse-to-fine strategy has been investigated. This technique defines the decision rules using the properties of the IN model while setting the verification criterion based on a set of newly designed convolution kernels. The method provides a very low false alarm rate and miss detection rate with noise densities as high as 90%, as observed by the authors.

The adaptive switching MeF (ASWeM) uses image pixel weighted statistics in a sliding window for threshold-free local construction [24]. The threshold and weighted standard deviation are calculated after each window calculates the weighted mean. The impulse detection method uses local intensity extreme analysis to detect noise pixels with 99% accuracy. The approach finds each window's local minimum and maximum for several noise levels. Modified decision based unsymmetrical trimmed MeF (MDBUTMeF) uses the median at low noise density for noisy pixels [25]. Every window element’s mean value replaces noisy pixels above 80% noise density. The recovered image features several noisy patches with significant noise densities because the mean values and original pixel are uncorrelated. Drawback: DBA and MDBUTMeF only examine local image information and ignore global pixel correlation. Another method, Switching Non-Local Means (SNLM), denoises switching frames using global image data [26]. To denoise images non-locally using noise-free pixels and select the filter, noisy pixels are assessed at the highest grey levels. IBINR [27] eliminates fixed-value IN quickly and easily. The nonlinear interpolation approach removes all noisy pixels from an image. The non-destructive noisy pixel replacement mechanism in this Iterative Bilateral Noise Removal (IBINR) makes noisy pixels easy to spot. WITM, a rich class of filters [28], reduce extreme samples and converge to the weighted mean each iteration. The weighted mean, MeFs, and suitable stopping condition make the filters outperform comparable techniques in this field. The adaptive weighted median filter (AWMeF) removes

high-level SPN, whereas the advanced MDBUTMeF or AMDBUTMeF fills the image border with zeros, including sliding window boundary pixels [29 – 30]. This approach raises window size for each pixel until two succeeding windows have equal maximum and minimum values. In this condition, the chosen pixel is replaced with the weighted average of the current window; else, it remains unchanged as given by:

$$S_{i,j}^{mean}(w)=\begin{cases}\dfrac{\sum\limits_{(k,l)\in S_{i,j}(w)} a_{k,l}y_{k,l}}{\sum\limits_{(k,l)\in S_{i,j}(w)} a_{k,l}}, & \text{for } \sum\limits_{(k,l)\in S_{i,j}(w)} a_{k,l}\neq 0;\\ -1, & \text{otherwise,}\end{cases} \tag{2}$$

where $a_{k,l}$ is 1 for $y_{k,l}$ lies between the minimum and maximum value otherwise it is 0.

The one-stage noise detection method may label noise-free pixels as noisy in flat image regions, while the two-stage method [31] decreases this. Filtering broken pixels and narrow sliding windows reduce blurring. Damaged pixels are used as missing data for inpainting [32]. It adaptively selects the convolution mask based on local characteristics and fixes it iteratively to provide reliable information for damaged pixels. The filter has 3×3 convolution and directional masks. Noisy pixels in smooth regions are reduced by a $k\times k$ mask in four directions. Textured noisy pixels use directional masks. The DBUTMWMeF algorithm replaces median with modified winsorized mean. It fixes SPN-damaged color and grayscale images. Noisy pixels are replaced with the modified winsorized mean of the unsymmetrical trimmed array [33]. Non-noisy pixels remain untouched. Preserving fine features at high noise densities eliminates the "SPN" effect and reduces image noise. Replace the minimum and maximum values with the array's closest observations to get the winsorized mean. A simple arithmetic mean computation after adjusting these numbers yields the average. Its key issue is that it only detects noisy pixels using local image information like viewport directional gray scale discrepancies and ignores global information. They cannot be utilized for noiseless gray scale 0 or 255 image graphs. It cannot reduce noisy false positives and missed detections. To tackle this problem, a new filter that accounts for local and global image information provides a patch-based noise density estimation technique with sufficient precision [34]. Data are utilized to detect and correct noise in the proposed filter. Global image information eliminates the difficulty of applying filters on extreme gray images without noise.

Morphological mean filters reduce high-density noise and improve image quality. Unlike previous filters, it uses the noise-free pixel counter (NPC) module to locate and count uncorrupted pixels [35]. Morphological pixel dilation (MPD) replaces neighbor noisy pixels with noise-free pixels. NPC and MPD are repeated until the stopping criteria are fulfilled. ASWMeF, which detects and removes noise, removes SPN effectively. A noise detection step classifies pixels as

noiseless or noisy after assessing noise candidates with local averages. Following detecting the noisy pixels, an adaptive weighted MeF with a window size of 3x3 or 5x5 replaces the detected pixels with their WM values. This approach offers several benefits and doesn't require any threshold settings or prior training [36] and is given by:

$$C(i,j) = M(i,j)\,\beta(i,j) + [1-\beta(i,j)]\,F(i,j)\,, \tag{3}$$

where $C(i,j)$ is the output, $F(i,j)$ is the input, $M(i,j)$ is the weighted median, and $\beta(i,j)$ is 1 and 0 for noisy pixels and noise-free pixels respectively.

Traditional approaches use local neighborhood information statistics to estimate noisy pixels and image detail corruption from local image diversity singularities. This is solved by an INLM using image non-local similarity [37]. The core pixels' gray value distribution fits similar blocks with repeated patterns. INLM replaces the WA of all central pixels in analogous patterns for the investigated pixel in three stages. Beginning with position data and a SW, a noise map is created. INLM patch change only impacts noise pixel-centered local windows. Finally, it approximates the corrected pixel iteratively. A three-value weighting method that examines extreme pixels with variable-sized local windows reduced SPN noise [38]. All local window non-extreme pixels are large, medium, or small. Number of non-extreme pixels in largest or smallest group determines intermediate group centroid. Non-extreme pixels' maximum, average, and minimum values are weighted by distribution scale. To restore noisy pixels, weighted values replace extreme intermediate pixels.

Image high-density SPN can be eliminated using decision-based adaptive kriging interpolation [39]. In constrained environments, semi-variance between corrupted and non-noisy pixels generates interpolated weights for corrupted pixels. The processing window must have three noise-free pixels or be adaptively resized. Adaptive filters outperform fixed window size filters at higher noise densities but take longer to install in image acquisition systems. Fast adaptive and selective mean filters (FASMFs) minimize time and improve results for higher noise densities better than adaptive weighted mean filters [40]. It gives high priority to those noise-free pixels that are nearer to noisy pixels, also known as shorter D8 or chessboard distance given by:

$$D8\ distance = \max(|x_2 - x_1|, |y_2 - y_1|)\,, \tag{4}$$

where (x_1, y_1) and (x_2, y_2) are two-pixel positions.

The effective hybrid genetic algorithm (EHGA) to remove SPN in grey images utilizes the output of three of the best filters such as DAMeF, AWMeF, and NAFSM, to create a set of an initial population [41]. To choose the best individual among the created population individuals a fitness function as defined has been used.

$$fitness(F) = \lambda\,|F - X| + \left(\sum_{\Omega}\sqrt{1+\beta^2\,|\nabla F|^2}\right). \tag{5}$$

The term $|F - X|$ is called the fidelity term, which guarantees a certain degree of idealist between the original image and the evaluated image, where F is the evaluated image, which is an approximation of the original image and X is the noisy image. The ∇F is a fully variable adjustment element, and β and λ are equilibrium parameters. **Table 1** tabulates the various switching-based filters with their advantages and limitations.

Table 1
Comparison of different Switching Based Filters.

Author	Filter Type	Advantages	Limitations
Eng *et al.* [17]	NASMeF	• Effective at low noise density.	• Fails to preserve edges beyond 50% noise densities.
Zhang *et al.* [18]	SMeF	• Distinguish thin lines from impulses.	• Operators are not applied in a diagonal orientation
Ng *et al.* [19]	BDND	• High accuracy, low false alarm rate, zero miss detection rate for noise levels up to 70%.	• Boundaries selection for clustering.
Srinivasan [20]	DBA	• Improves edge preservation. • Low computation time.	• Fails to handle highly corrupted images. • Produces streaking effect.
Zhang *et al.* [22]	SAWMeF	• Better restoration performance and high computational efficiency.	• A predefined threshold value leads to image degradation.
Duan *et al.* [23]	HEIND	• False alarm rate and miss detection rate are remarkably low.	• Miss detection and false rate can be improved by applying adaptive and iterative processes.
Akkoul *et al.* [24]	ASWM	• Weighted statistics determine the threshold value locally.	• Leads to uncertainty problems and gives incorrect detection results.
Esakkirajan *et al.* [25]	MDBUTMeF	• Better restoration at low noise density.	• Less effective noise densities greater than 70%.
Nasri *et al.* [26]	SNLMF	• When reducing IN, take into account the image's global information.	• Tuning parameters are not optimized. • Blurring effect.
Kalyoncu *et al.* [27]	IBINR	• There is no damage caused by the noisy pixel replacement method.	• Edges suffered from the jittering effect.
Miao *et al.* [28]	WITM	• The weighted mean and MFs' inherent advantages are shared by WITM filters.	• Truncating samples leads to a smoothing effect at the edges.
Zhang *et al.* [30]	AWMeF	• Better result at high noise densities. • Low miss detection and low false alarm rate.	• It consumes more time, making it impractical to implement in image acquisition devices.

Table 1
Comparison of different Switching Based Filters-continued.

Author	Filter Type	Advantages	Limitations
Zhang *et al.* [32]	Image inpainting	• The visual quality of the restored image is efficiently improved.	• Smoothing effect at high noise densities due to non-uniform of masks selection.
Vasantha *et al.* [33]	DBUTWMeF	• Image's fine details even with extreme noise levels.	• Fixed window size • Edges have a smoothing effect
Li *et al.* [34]	Image block-based	• Global image information and better estimation accuracy.	• Due to the multiple noise detection scheme, the method is more complex.
Lin *et al.* [35]	Morphological mean	• Remove high-density noise and improve image quality.	• The image is blurred at high noise densities.
Faragallah *et al.* [36]	Adaptive switching weighted	• Both threshold parameters and prior training are not necessary.	• Weight is given more to only horizontal and vertical directions. • Window size is limited to only two sizes.
Wang *et al.* [37]	INLMF	• Exploit the image non-local similarity feature.	• Execution time is more. • Edges got blurred due to global information considered at high noise densities.
Varatharajan *et al.* [39]	Kriging interpolation	• Since all errors have the same mean, this interpolation is impartial. The goal of this interpolation is to reduce error variation.	• It requires more computing time and more inputs. • Required special care in correlating the pixels.
Beagum *et al.* [68]	FASMF	• Execution time is less.	• Restored result image is more blurred.

3 Fuzzy Based Filter

The MF adjusts degraded pixel intensity while preserving local features. However, damaged pixels in noisy images are hard to spot, and even basic threshold approaches cannot distinguish them. Some uncorrupted pixels have these two values. Fuzzy logic has been added to MF to improve pixel classification [6 – 8]. This can be used to measure pixel IN corruption or perform a blur degradation measure-based method. Some methods use fuzzy logic to select the appropriate filter for an input image. Fuzzy logic involves fuzzification and defuzzification. The noisy image must be blurred first. Blurring uses pixel intensity, or the difference between the present pixel and the surrounding area. The fuzzy values are used to filter noise by the system. The outcome is determined by defuzzification.

3.1 Type-1 fuzzy filter

An operator based on two-step fuzzy inference consisting of two cascading subunits is designed to improve the images corrupted by IN [42]. The first subunit is the action detection module, which is used to detect noisy pixels by considering the pixel difference between adjacent pixels and thus selecting the possible correction terms. The second subunit is an action adjustment module, which changes the value of this correction term to further improve detail retention. The adaptive fuzzy switching filter (AFSF) comprises three sub-units. The first subunit convolves the image among four one-dimensional Laplacian operators before detecting the noisy pixel by considering the pixel distribution among nearby pixels. By removing pixels from the specified window whose neighbouring pixels equal one-fourth of the maximum or minimum values, the second subunit calculates the pixels.

In comparison to previous more complex detectors, the new IN reduction algorithm [3], which is based on a histogram and fuzzy impulse detection technique, has superior noise removal capacity. There are two steps to it, such as fuzzy impulse detection and IN cancellation, which require no prior training. In the fuzzy impulse detection stage, a fuzzy flag map is created, indicating how much a pixel looks like an impulse pixel. The MF $f_{i,j}$ for noisy pixels is defined as:

$$f_{i,j} = \begin{cases} 0, & M_{i,j} \le T_1; \\ \dfrac{M_{i,j} - T_1}{T_2 - T_1}, & T_1 \le M_{i,j} \le T_2; \\ 1, & M_{i,j} \ge T_1, \end{cases} \tag{6}$$

where T_1 and T_2 are two predetermined parameters and high-quality restored images are obtained when T_1 lies in [10, 20] and T_2 lies in [20, 30]. In the impulse noise cancellation stage, the restored pixels $y_{i,j}$ is defined as:

$$y_{i,j} = (1 - f_{i,j}) x_{i,j} + f_{i,j} m_{i,j}, \tag{7}$$

where $m_{i,j}$ is the median of the processing window.

In [43], a two-stage noise adaptive fuzzy switching median (NAFSM) filter hybridizes the fuzzy-switched MF and a basic adaptive MF to detect and eliminate SPN. The distorted image's histogram is used to identify noisy pixels, and fuzzy reasoning is used to address the uncertainty of the recovered local information's noise. The adaptive NAFSM filter widens its filtering window in response to local noise level, filtering high-density SPN. The inherited switching median characteristic accelerates filtering while preserving image attributes. Fuzzy Based IN Reduction (FBINR) removes IN [7] from images more

effectively, saving time on source image learning. FBINR uses the novel fuzzy filter (NFF) and histogram-based fuzzy filter to compute fuzzy sets. An intelligent fuzzy controller for IN reduction, noise detection, and a detail preservation module preserve image details while removing noise. A new adaptive fuzzy switching weighted mean filter that uses maximum absolute luminance difference (ALD) to remove SPN more accurately [44]. To better maintain edges, the ALD used a fuzzy method to calculate the weighted mean utilizing distance-relevant. A pixel is identified as belonging to one of the three groups by the fuzzy flag $f_{i,j}$.

$$f_{i,j} = \begin{cases} 0, & M - ALD \le T_1; \\ \dfrac{M - ALD - T_1}{T_2 - T_1}, & T_1 \le M - ALD \le T_2; \\ 1, & M_{i,j} \ge T_1, \end{cases} \tag{8}$$

where T_1 is 10 and T_2 is 30 are two predetermined parameters. If, $f_{i,j} = 0$ means the pixel is uncorrupted, between 0 and 1 means a lightly corrupted pixel, and 1 implies a heavily corrupted pixel. The pixel is restored by

$$y_{i,j} = (1 - f_{i,j}) x_{i,j} + f_{i,j} x_{mean} \tag{9}$$

and helps in better edge preservation than [3].

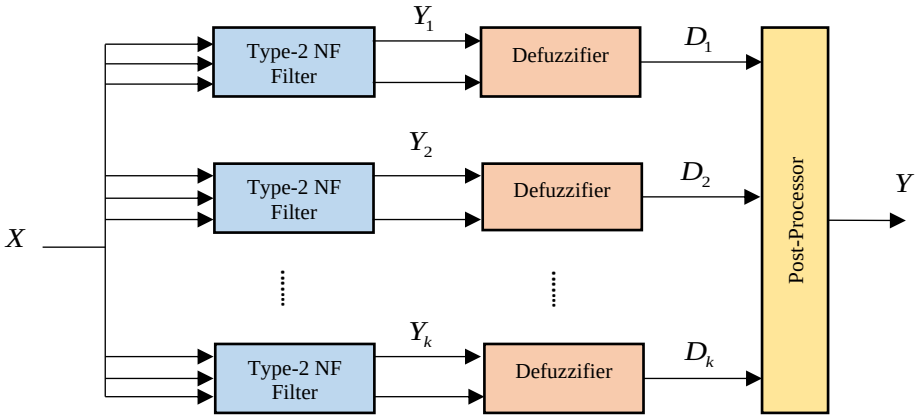
3.2 Type-2 Fuzzy filter

Fuzzy logic for image denoising is complicated by noisy pixels when assigning member values, which generates model uncertainty in noisy situations. The type 2 fuzzy logic set (T2FLS) handles greater ambiguity and supplies more missing elements, improving decision-making [45, 46]. This has second-order MFs for each first-order MF. Yildirim et al. [47] presented a T2FLS-based filter to preserve thin edges, textures, and other image characteristics. Fig. 2 shows a Type 2 neuro-fuzzy (NF) structure and NF pixel neighborhood topologies using the pixel position and its neighbour.

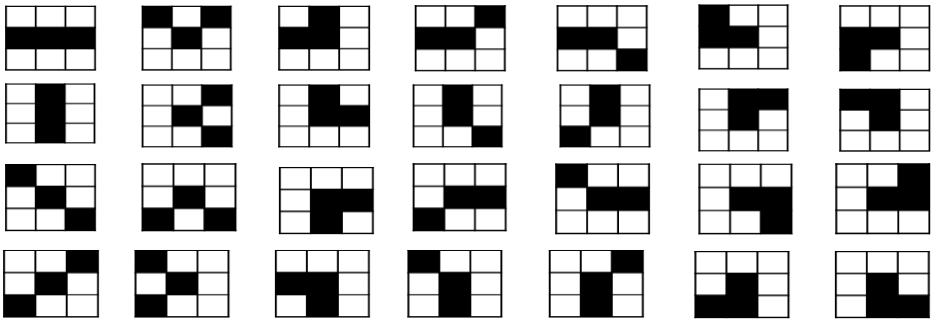
A type 1 fuzzy set representing the source pixel's lower and upper bound segment uncertainty is output. The NF filter blurs and scalarizes the input to produce the fuzzified output. The fuzzified scalar value output is the filter window pixel position's positive value contender. After evaluating these values, post-processing outputs the requested operator for the filter window pixel position. For each type 2 NF filter, the number of fuzzy rules N in the base rule is 30. For an NF filter, the input-output relationship is given by

$$\begin{aligned} &\text{if } (X_1^k \in M_{N1}^k) \quad \text{and} \quad (X_2^k \in M_{N2}^k) \quad \text{and} \quad (X_3^k \in M_{N3}^k), \\ &\text{then} \quad R_N^k = c_{N1}^k X_1^k + c_{N2}^k X_2^k + c_{N3}^k X_3^k + c_{N4}^k \end{aligned} \tag{10}$$

where X_1^k, X_2^k and X_3^k denote the inputs of the k^{th} NF filter and Y_k denote its output. Each combination of inputs and their associated membership functions is represented by a rule in the rule base of the k^{th} NF filter, N is the number of fuzzy rules in the rule base, M_{ij}^k denotes the i^{th} membership function of the j^{th} input, and R_N^k denotes the output of the i^{th} rule.



(a)



(b)

Fig. 2 – (a) *Structure of the Type-2 NF operator,*
(b) *Possible pixel neighbourhood topologies* [54].

Khanesar et al. [48] have implemented the tuning rules of the T2FLS. The authors have derived MFs in equation (11) from the learning descent with whole values at both ends of the support.

$$\mu(x)=\begin{cases}\left(1-\left|\dfrac{x-c}{d}\right|^{a}\right)^{\frac{1}{a}}, & \text{if } c-d<x<c+d;\\ 0, & \text{else;}\end{cases} \tag{11}$$

where c and d are the center and the width of the membership function and x is the input vector.

The variables a_1 and a_2, $(a_2<a<a_1)$ determine the width of the MF uncertainty using values $a_1<2$ and $a_2>0.5$. Two FLSs were used to demonstrate T2FLS's non-ideal features. Both entries in the initial T2FLS have two MFs to decide identification, prediction, and control applications. An alternative T2FLS using the Sugeno intermittent to train and detect impulses was proposed by Yüksel et al. [49]. The proposed detector has two T2FLS sub-detectors, two defuzzifiers, and a postprocessor. The pulse noise filter and practical pulse detector cooperate. A predetermined IN concept does not make the detector independent. The connected device receives the filter window pixel center and its horizontal or vertical neighbors. T2FLS and Type-1 FLS are used by the sub-detector on input. The lower and upper bound uncertainty ranges for the positive pixel position are shown. The interactive defuzzifier converts type-1 FLS to a scalar value for centroid defuzzification using detector fuzzy outputs. The authors introduce a new performance measure parameter for colour images as mean pixel distance (MPD), which is defined as:

$$MPD=\frac{1}{LC}\sum_{i=1}^{L}\sum_{c=1}^{C}\left\|s[l,c],y[l,c]\right\|, \tag{12}$$

where $s\,[l,c]$ and $y\,[l,c]$ are vector quantities in the RGB colour space at the line l and column c of the original and restored colour test image and $\left\|s[l,c],y[l,c]\right\|$ is the Euclidean distance between two points in the colour model defined as:

$$\left\|s[l,c],y[l,c]\right\|=\sqrt{\sum_{i\in(R,G,B)}\left(s_i[l,c]-y_i[l,c]\right)^2}\,. \tag{13}$$

Zhai et al. [8] demonstrate the Quantum-based PSO-based NS-IT2 FLS for mixed Gaussian IN (MGIN) elimination. By modeling equipment as FSs instead of real numbers, users can add measurement parameter uncertainty and design freely. The post-processor does not train neuro-fuzzy filters because each filter is learnt separately using one image set. NS-IT2 FLS tunes neuro-fuzzy filter parameters for optimal solution using a PSO algorithm instead of a gradient-based technique, which is harder to implement. QPSO outperforms PSO and provides a decent search space solution, but not global optimization. NS-IT2 FLS adjusting once for numerous image graphs saves time processing noisy images.

Singh et al. proposed an adaptive Type-2 fuzzy filter [50] to identify good and bad pixels using a membership function with varied mean and variance. Adding a Type-1 fuzzy filter removes noise without changing the setting at 90% noise levels. Singh et al. [51] cut complexity by employing an adjustable threshold to detect noisy pixels and two membership functions instead of five [50]. The first phase's noisy pixels are denoised using noise-free filter window pixels. The system preserves image corners and edges. Only two Gaussian MFs (GMF), upper and lower, provided below, are required to determine an adaptive threshold instead of multiple MFs as required in other filters.

$$\overline{\mu}(p_r)=\begin{cases}\mu_{P_{i,j}^{(H,1)}}(m_1^H,\sigma^H), & \text{if} \quad p_r^H < m_1^H;\\ \vee(\mu_{P_{i,j}^H}(p_r)), & \text{if} \quad m_1^H \le p_r^H \le m_2^H;\\ \mu_{P_{i,j}^{(H,2)}}(m_2^H,\sigma^H), & \text{if} \quad p_r^H > m_2^H\end{cases} \tag{14}$$

and

$$\underline{\mu}(p_r)=\begin{cases}\mu_{P_{ij}^{(H,2)}}(m_2^H,\sigma^H), & \text{if} \quad p_r^H \le \dfrac{m_1^H+m_2^H}{2};\\ \mu_{P_{ij}^{(H,1)}}(m_1^H,\sigma^H), & \text{if} \quad p_r^H > \dfrac{m_1^H+m_2^H}{2},\end{cases} \tag{15}$$

where $\mu_{P_{i,j}^{(H,k)}}$ is the mean of the Gaussian membership function of the neighbourhood pixel set P_{ij}^H for pixel p_{ij} and σ^H is the variance.

For denoising, a classical mean is used to determine the mean of the Gaussian membership function instead of the inverse distance-based method. The Gaussian MF $\mu_G(g_i)$ for the classical mean is defined by:

$$\mu_G(g_i)=\exp\left[-\frac{1}{2}\left(\frac{g_i-m}{\sigma_G}\right)^2\right], \tag{16}$$

where m and σ_G are the mean and variance of noise-free pixels g_i. A modification to [51] was done in [52] by combining the process of type-2 fuzzy for the detection of noisy pixels by eliminating the noisy pixels for determining k-mean and using the linear regression-based method for estimating the correct pixel value of noisy pixels. Liu et al. [53] present a novel T2FLS-matrix completion approach for image inpainting. The membership identifies and clips noisy pixels from the noise image, solving the matrix completeness problem. A new full-matrix method without a priori rank information for image noise reduction with parameter-less tuning features outperforms established category

methods. The suggested approach removes noise reliably in test images, and the PSNR is superior than noisy SPN filters.

4 Deep Learning Based Filter

Traditional filters depend on fixed mathematical concepts to remove noise from the image (like mean, Gaussian, median, Wiener filters, etc.). All the above discussed methods are based on assumptions of predefined rules and the noise distribution, which particularly aim to reduce noise while trying to preserve essential image features like edges. These filters often struggle with complex or non-standard noise patterns which greatly affect their effectiveness and its performance is limited by the need for careful parameter tuning. In contrast, deep learning methods, take a data-driven approach to learn. A large dataset of paired noisy and clean images for training, which enables them to extract features and understand complex noise patterns automatically. Deep learning models use multiple layers as input layer, hidden layer, output layer as shown in Fig. 3 to learn various aspects of an image, from simple details like edges in the initial layers to greater details in deeper layers. The input layer accepts noisy image as input. The hidden layer consists of convolutional layer (extract features like edge and texture), batch normalization (for stabilization) and pooling layers (for minimizing computation) and the output layer produces the noise free output image. The hierarchical structure of deep learning adapts more effectively to various types of noise and characteristics while preserving image details, which make them suitable for real-world scenarios where noise patterns are unpredictable.

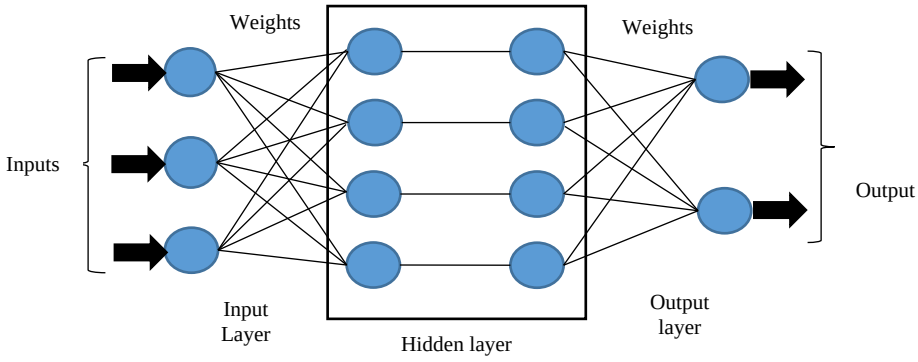


Fig. 3 – *Basic architecture of Deep Learning.*

But on the other side, this enhanced performance comes at the cost of higher computational complexity and resources. For training deep learning models, substantial computational resources are required, but once its trained, these models can be optimized. On the other hand, traditional filter is simpler and faster

to implement, and require less resources and often used in real-time. In a broader sense, traditional filters are useful for simpler tasks and environments, deep learning approaches offer superior performance and versatility, especially in handling complex noise scenarios.

Deep neural network approaches use hierarchical layers of information-processing stages to learn. DL, a subtype of ML, is inspired by human brain information processing. DL uses massive data sets to label input instead of human-designed rules. AI is used to create deep learning. These multi-layer algorithms interpret data uniquely. DL approaches are divided into discriminative (supervised) and generative (unsupervised) learning. Here, we briefly describe various algorithms employing deep learning model as: CNN, MLP, RNN, GAN, and AE.

4.1 Convolutional Neural Network (CNN) based filter

CNN machine learning techniques have advanced recently. One of the deep neural network methods that improves classification, identification, segmentation, denoising, etc. This network's weight replication minimizes network complexity by reducing trainable parameters and improving generalization over ANNs [54]. Due to its increased representation capability, the network outperforms sparse representation in image resolution [55]. Sparse representation loses 2D-structural information because sparse dictionaries are vectorized images. The network may preserve 2D structure development during training and testing while convolution considers local pixels of the 2D mask [56]. Turkmen [57] developed a multi-layer perceptron-based ANN detector and decision maker employing rank-ordered absolute differences (ROAD) and rank-ordered logarithmic difference (ROLD) as input variables, one-tenth pixels of randomly picked training image, 70% random valued IN density, and 150 epochs of training. Neural detector output is used by the decision-maker. The output of the decision-maker (i, j) is calculated as follows:

$$DM(i,j) = \begin{cases} L_{\min}, & \text{if} \quad DO(i,j) < \dfrac{L_{\min} + L_{\max}}{2} \\ L_{\max}, & \text{if} \quad DO(i,j) \geq \dfrac{L_{\min} + L_{\max}}{2} \end{cases}, \tag{17}$$

where $L_{\min}$ and $L_{\max}$ are minimum and maximum values of the pixel range of an image.

The pixels are denoised by the edge-preserving regularization which preserves the noise-free pixels and edges. The deep network CNN (DnCNN) [58] is a nonlinear mapping model used to denoise RVIN by changing training data. For the accurate detection of noise, the noise ratio predictor (NRP) can determine the severity of corruption. An adaptive blind CNN was proposed by Chen *et al.* [59] as shown in Fig. 4, is the most suitable DnCNN model exploited for image restoration rapidly and efficiently. The noise detector of the key components of

NRP was trained by the Berkeley segmentation dataset (BSD) at six CNN denoising models by ROAD, which provides suitable characteristics of noise pixels and edge pixels.

A hybrid model comprising the CNN and PSO has been applied to detect noisy pixels and denoising, wherein the former detects the IN using a Gaussian denoiser [60]. The PSO searches and finds the optimal threshold T1 and T2 for detecting noisy pixels while the MeF handles the false detections. The model is trained with various natural images for high noise densities. For training the network, the Stochastic Gradient Descent (SGD) corresponds to a particular noise level. It achieves an overall best performance at $T1 = -0.0714$ and $T2 = -0.1121$, corresponding to different noise levels. Fig. 5 provides the Hybrid Impulse Noise Denoiser using PSO and CNN.

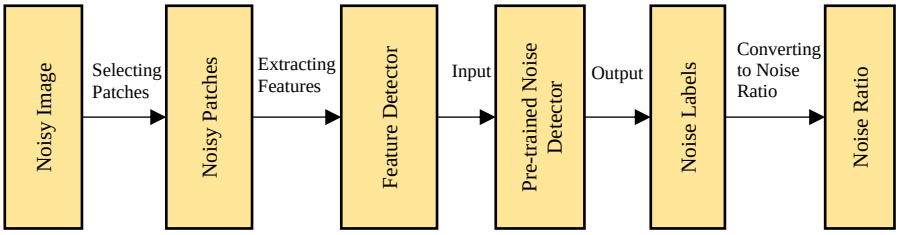


Fig. 4 – *Block diagram of noise ratio predictor.*

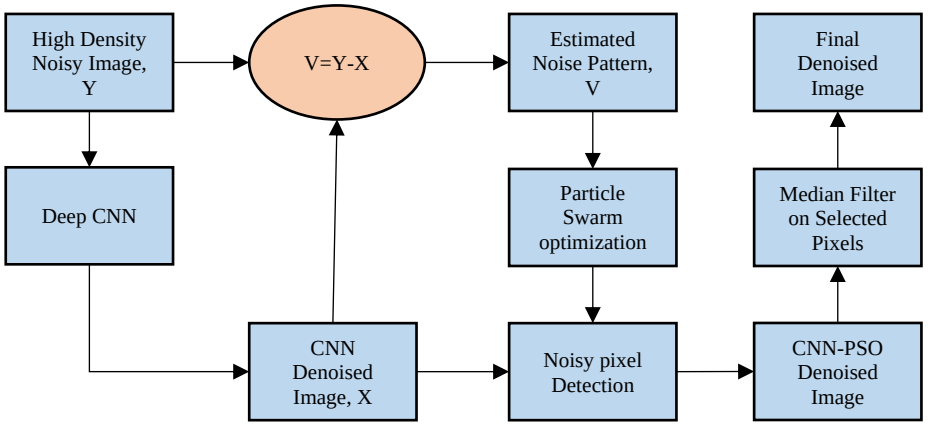


Fig. 5 – *Block diagram of CNN-PSO.*

A modified CNN network model comprising two deep CNNs as a classifier to detect IN and a regression network to restore the noisy pixels has been proposed [61]. In the testing stage, each noisy or noise-free pixel is labeled by the classifier network, and a sparse, clean image is obtained based on the noise detection results. To reconstruct the denoised image the sparse clean image is applied along with the original noisy image. To verify the training performance,

400 images of size 180×180 and sixty-eight 321×481 sized natural images from the BSD68 dataset have been used. For better outcomes, the adaptive moment estimation rather than the SGD has been used for training. It uses a learning rate of 0.001 and 0.0002 for the first 30 and the last 20 epochs, respectively. The modified CNN structure is shown in Fig. 6.

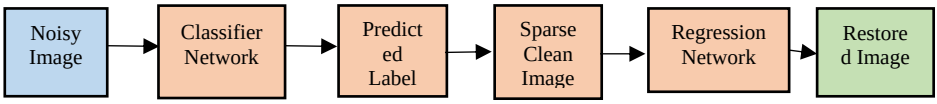


Fig. 6 – *Block diagram of DCNN.*

In view of some CNN methods, especially the DenseNet, Li *et al.* [62] have proposed a densely connected network for impulse noise removal (DNINR). In contrast to the traditional deep CNN model, DNINR is based on nonlinear learning and residual learning principles which utilizes CNN to learn pixel distribution features. A total of 12 layers is included and a lightweight network is generated for better estimation by designing a multi-scale, with the potential for bidirectional information flow between each layer. The DNINR structure utilizes pre-activation batch normalization. The network consists of three layers, namely, convolution layers, batch normalization layers, and rectified linear unit (ReLU) only and it is chosen as the activation function in the whole architecture. The model was trained with stochastic gradient descent (SGD) and it achieved better performance than other methods. The first layer is expressed as an operation:

$$G_0 = Et(x), \tag{18}$$

where $Et(x)$ denotes the feature extraction operation. G_l receives the feature maps of previous preceding layers, and

$$G_l = C_l(G_{l-1}) = C_l(C_{l-1}(\ldots(C_1 G_0))\ldots), \tag{19}$$

where C_l denotes the l^{th} function and G_{l-1}, G_l are input and output of l^{th} layer respectively.

The DNINR model is expressed as:

$$Y = f_{dt}(G_l), \tag{20}$$

where f_{dt} denotes the distribution transformation function.

Contrary to traditional CNNs for denoising in which noisy images directly act as an input to estimate the denoised image, the non-local switching filter CNN (NLSFCNN) uses a pre-processing step before training the network. The pre-processing step detects the noisy pixels while a non-local switching filter smoothens them. The network generates the overlapping patches from pre-processed images and determines the optimal parameters. The n^{th} operation $f_n(P)$ using a base filter weight W_n and bias B_n is defined as:

$$F_n(P) = \max(0, W_n F_{n-1}(P) + B_n), \quad (21)$$

where $F_n(P)$ are the operators defined for the nth layer, W_n are the bases filter set, and B_n are the biases of the respective layer.

Sadrizadeh et al. [63] have proposed an end-to-end fully blind CNN that requires no pre-processing or noise detection algorithm to remove the IN. The authors consider a customized dataset with different noise densities to train the network while emphasizing higher noise density, thus reducing the response time. The new multi-term loss function used in the process is given by

$$L_{Total} = L_{MSE} + 0.1L_{MAE} + 0.075L_{DSSIM} + 0.015L_{NSC}, \quad (22)$$

where MSE is a mean squared error, MAE is the mean absolute error, DSSIM is the dissimilarity structural index metric and NSC is noise sparsity constraint. An iterative inpainting method is used as a post-processing algorithm where an exponential smoothing kernel is used to denoise the image. According to [64], the learning-based network for removing SPN has some drawbacks, such as affecting CNN function because there is no correlation between noisy pixels and noise-free pixels, and a residual learning strategy, which is used in some of these networks like DnCNN, is ineffective and introduces visual artifacts. To tackle this, a multi-layer deep convolutional neural network model called SeConvNet was proposed to eliminate SPN from images. To ensure a uniform representation of noisy pixels, all 255-valued pixels are converted to 0. SeConvNet's seven blocks with rising kernel sizes (3 to 15) selectively recover noisy pixels while preserving non-noisy pixels. The 8th to the layers use typical convolutional layers with 64 filters (3×3), ReLU activation, and batch normalization. The last layer reconstructs the denoised image using convolution with filters matching the image channels (1 for grayscale, 3 for color). The result is multiplied by a noisy pixel's map and added back to the input to modify just noisy pixels and preserve clean pixels. This design better removes noise and preserves image information.

4.2 Multilayer Perceptron (MLP)

A supervised learning method known as the multi-layer perceptron (MLP) uses multiple intermediate hidden layers to transfer the input image vector to the output image vector [65 – 66]. It is based on a feed-forward model. A fully linked network with an input layer that receives input data, an output layer that decides or predicts something based on the input signal, and one or more hidden layers acting as the network's computational engine in between are the components of a typical MLP. Many activation functions, often referred to as transfer functions, like ReLU (Rectified Linear Unit), Tanh, Sigmoid, and Softmax [67] are used to calculate an MLP network's output. The most widely used algorithm, "Backpropagation", is used to train MLP networks. Several optimization techniques, including Adaptive Moment Estimation (Adam), Limited Memory

BFGS (L-BFGS), and Stochastic Gradient Descent (SGD), are used during the training phase. The number of hidden layers, neurons, and iterations are only a few of the hyperparameters that need to be tuned for MLP, which could increase the computational cost of solving a complex model. Because vanishing gradients limit the number of layers in MLPs, compared to convolutional neural networks, which limit their performance.

4.3 Recurrent Neural Network (RNN)

Another well-liked neural network is the Recurrent Neural Network (RNN), which uses time-series or sequential data and feeds the output of the previous stage as input to the current one. Recurrent networks, like feedforward and CNN, learn from training input [68 – 69]; however, they differ in that they include a "memory" that enables them to influence current input and output by utilizing data from earlier inputs. The output of an RNN depends on earlier items in the sequence, in contrast to a normal DNN, which operates under the assumption that inputs and outputs are independent of one another. However, the problem of vanishing gradients in typical recurrent networks makes learning large data sequences difficult. Several well-liked recurrent network variations that reduce problems and function effectively in numerous real-world application domains include Gated Recurrent Units (GRUs), Bidirectional RNN/LSTM, and Long Short-Term Memory (LSTM) [70]. All things considered, the fundamental characteristic of a recurrent network is the presence of at least one feedback connection, which permits activations to repeat. As a result, the networks can perform temporal processing and sequence learning tasks including temporal association or prediction, sequence reproduction or recognition, etc.

4.4 Generative Adversarial Network (GAN)

One kind of neural network architecture called a Generative Adversarial Network (GAN) was created to get around the challenge of deep generative models learning complicated probabilistic distributions. To enable the model to create or output new instances from the original dataset, it entails automatically identifying and learning regularities or patterns in the input data. The generator and discriminator sub-models of generative modeling are used by the generative adversarial network (GAN) [71]. The discriminator D forecasts the possibility that a subsequent sample will be taken from real data rather than data supplied by the generator, and generator G generates new data with attributes comparable to the original data. As a result, in GAN modeling, the discriminator and generator are educated to compete with one another [72]. Although GAN network deployment is often intended for unsupervised learning tasks, depending on the challenge, it has also shown to be a superior solution for semi-supervised and reinforcement learning. In addition, GANs are employed to enforce the alignment of the latent feature space in cutting-edge transfer learning research [73].

4.5 Auto Encoder (AE)

Neural networks are employed in an auto-encoder (AE) [74], a prominent unsupervised learning technique, to learn representations. High-dimensional data is typically worked with auto-encoders, and dimensionality reduction describes the representation of a set of data. An auto encoder consists of three parts: the encoder, the code, and the decoder. The input is compressed by the encoder, which also produces the code that the decoder needs to reconstruct the input. Generative data models have recently been learned using the AEs. In numerous unsupervised learning tasks, such as dimensionality reduction, feature extraction, effective coding, generative modeling, denoising, anomaly or outlier detection, etc., the auto-encoder is frequently utilized. Large data sets can also be de-dimensionalized via principal component analysis (PCA), which functions similarly to a single-layered AE with a linear activation function. Variational auto encoders can be utilized as generative models, and regularized auto encoders, including auto encoder [75], k-sparse [76] denoising, contractive [77], denoising, and sparse, help learn representations for subsequent classification tasks. Comparison of discussed deep learning methods are shown in **Table 2**.

Table 2
Comparison of Deep Learning Methods.

Method	Architecture	Advantages	Disadvantages
MLP	Fully connected layers	• Simple to implement. • Ability to learn non-linear network	• Doesn't have spatial domain information. • Huge data is required for better results. • Different random weights give different accuracy due to the non-convex loss function.
CNN	Convolutional layers	• Learning is very effective. • Powerful for removing the noise with known statistics. • Effective in blind learning. • Fewer parameters are required. • Weight learning concepts can be used by other networks. • Can be useful to deal with real-world noise.	• The noise detectors are not completely free from false detections and missed detections which results in blurring of the image, while missed detections lead to residual noise pixels. • High Computational complexity. • Efficiency is low in unsupervised learning.
AG	Generator + Discriminator	• Learn complex mapping effectively. • Can be trained using unlabelled data. • Better classification of data.	• Training is extensive. • Can suffer from the collapse of architecture. • Vanishing gradients problem. • Training can be slow.
AE	Encoder-Decoder structure	• Better balance of implementation and performance. • Designed for dimensionality reduction and reconstruction	• Less effective at high noise.

5 Results and Discussion

For a comparative study and analysis of the considered methods, we separately consider the spatial domain filter, fuzzy-based filter, and CNN-based filter. In each of these, the analysis is carried out by quantitative parameters and visual perception. For visual perception, edge determination, texture preservation, and artifacts in the restored image are generally considered. The simulation is performed in MATLAB R2021a with an Intel Core i5 processor and 8GB memory on Windows 11. For the quantitative analysis, the parameters used are Peak signal to noise ratio (PSNR) and Structural similarity index measure (SSIM).

PSNR is the ratio of the maximum possible power of a signal to the power of corrupting noise that affects the fidelity of its representation.

$$PSNR = 10\log_{10}\frac{(255)^2}{MSE} \quad [\text{dB}] \tag{23}$$

and

$$MSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[z_{ij} - x_{ij}]^2}{M \times N}}\,. \tag{24}$$

SSIM (Structural Similarity Index Measure)

$$SSIM = \frac{\left(2\mu_x\mu_z + C_1\right)}{\left(\mu_x^2 + \mu_z^2 + C_1\right)} \times \frac{\left(2\sigma_{xz} + C_2\right)}{\left(\sigma_x^2 + \sigma_z^2 + C_2\right)}, \tag{25}$$

where x_{ij} is the original image, y_{ij} refers to a noisy image, z_{ij} refers to a denoised image, and μ_x, μ_z, σ_x and σ_z are the mean and variances of original image *x* and denoised image Z respectively, σ_{xz} is the covariance between *x* and *z*, C_1 and C_2 are constant values.

For comparing spatial domain filters, four testing standard images Lena, Cameraman, Barbara, and Goldhill of dimension 512×512 are considered which are being extensively used in literature. The parameters on which the models are evaluated are PSNR and SSIM. For analyzing the different CNN models as discussed in the literature, the learning model training dataset consists of 91 test images and the training parameters are set as described in the respective paper and as shown in **Table 3**.

The parameters of the different models are best tuned as in their work and some adjustments are done during simulation. For a fair comparison, we used 12 standard testing images, such as Lena, Cameraman, Boat, and more as shown in Fig. 11. The testing images are in the gray scale of dimensions 512×512. The parameters on which the models are evaluated are PSNR and SSIM.

Table 3
Training Parameters.

Parameters	ANN	Blind CNN	CNN-PSO	DCNN	DNINR	NLSF-CNN	SeConvNet
Batch Size	64	64	128	64	64	64	128
Learning Rate	10^{-1}	10^{-2}	10^{-2}	10^{-3}	10^{-1}	10^{-3}	10^{-3}
Epochs	150	50	60	60	60	50	50

Table 4 shows the various spatial filter PSNR and SSIM values at different noise level densities for different test images.

Table 4
PSNR v/s Noise density of spatial domain filter.

Image	Noise Density (%)	Filter							
		AMF	NASMF	MDBUTMF	SNLM	AWMF	ASWMF	FASMF	EHGA
Lena	20	26.88	31.67	32.67	35.23	34.98	38.71	39.19	39.26
	40	23.87	28.47	29.02	31.98	32.04	35.48	35.18	36.33
	60	21.63	25.94	26.31	29.91	30.29	32.04	31.63	30.77
	80	19.31	22.47	22.94	24.18	26.96	27.98	27.26	28.31
	Average	22.17	26.1	26.69	29.41	30.34	33	32.22	33.69
Barbara	20	24.16	28.82	28.7	29.95	29.94	30.97	31.02	31.24
	40	21.68	25.91	25.33	27.18	27.24	28.38	27.66	27.78
	60	19.72	23.61	23.22	25.42	25.18	25.63	25.4	25.47
	80	17.65	20.45	20.46	20.55	22.76	22.38	22.94	23.06
	Average	20.18	23.75	23.56	25	25.64	25.89	25.95	26.12
Cameraman	20	27.01	31.04	32.59	34.53	33.84	37.94	40.06	40.13
	40	23.85	27.9	29.24	31.34	33.03	34.77	35.48	35.46
	60	21.39	25.42	26.06	29.31	30.4	31.4	31.25	31.3
	80	18.65	22.02	21.93	23.7	25.86	27.42	26.12	26.41
	Average	21.99	25.58	26.34	28.82	29.97	31.71	31.92	32.09
Goldhill	20	26.05	27.01	31.28	32.8	33.13	36.77	36.68	36.92
	40	22.99	24.23	28.25	30.71	32.09	34.77	33.06	33.24
	60	20.89	22.17	25.66	28.72	29.89	31.4	30.17	30.27
	80	18.97	19.47	22.67	23.23	26.54	27.42	26.77	26.97
	Average	21.57	22.59	26.02	28.12	29.71	31.57	30.71	30.88

AMF filter results have very low PSNR at high noise densities due to the direct replacement of noisy pixels with the median of processing windows regardless of considering local information. NASMF fails to preserve details due to mis-classification of pixels beyond 50% noise. AWMF has a better noise-

confirming method of two successive windows for minimum and maximum pixels and it reflects better PSNR values at high noise densities, but it takes more time to execute. MDBUTMF replaces the noisy pixels with the mean of window and at high noise the probability of mean of window to be noisy is high. Similarly, FASMF accounts for only the shorter chessboard distance pixels for restoration and it is reflected in the result. ASWMF leads to uncertainty problems due to local threshold value which lacks the global information at high noise. SNLM suffers at high noise density due to tuning parameters are not optimized and hence switching the window becomes difficult. A better PSNR is achieved by the EHGA filter due to the use of the initial population for the genetic algorithm and the denoised pixel is obtained by crossover and mutation operator applied to the initial population. **Table 5** shows the SSIM of the denoised image of various filters. SSIM of FASMF surpasses in comparison to other filters at high noise densities which is due to the consideration of lesser chessboard distance pixels. EHGA performs better at low noise densities but fails at high noise densities for better SSIM.

Table 5

SSIM v/s Noise density of spatial domain filter.

Image	Noise Density (%)	Filter							
		AMF	NASMF	MDBUTMF	SNLM	AWMF	ASWMF	FASMF	EHGA
Lena	20	0.91	0.82	0.86	0.91	0.93	0.94	0.89	0.91
	40	0.74	0.74	0.78	0.84	0.86	0.88	0.81	0.85
	60	0.70	0.68	0.71	0.75	0.76	0.75	0.70	0.74
	80	0.71	0.60	0.62	0.68	0.67	0.70	0.63	0.66
	Average	0.76	0.71	0.74	0.79	0.80	0.82	0.77	0.79
Barbara	20	0.84	0.76	0.79	0.85	0.87	0.87	0.80	0.82
	40	0.68	0.67	0.71	0.78	0.79	0.81	0.72	0.76
	60	0.62	0.57	0.61	0.67	0.70	0.70	0.62	0.65
	80	0.55	0.51	0.54	0.59	0.62	0.62	0.54	0.58
	Average	0.68	0.64	0.67	0.73	0.74	0.75	0.68	0.71
Cameraman	20	0.87	0.79	0.82	0.87	0.89	0.90	0.83	0.85
	40	0.71	0.71	0.74	0.81	0.82	0.84	0.75	0.80
	60	0.65	0.61	0.64	0.70	0.73	0.73	0.66	0.69
	80	0.58	0.54	0.58	0.63	0.65	0.65	0.59	0.62
	Average	0.72	0.67	0.71	0.76	0.77	0.78	0.72	0.74
Goldhill	20	0.89	0.80	0.84	0.89	0.91	0.92	0.80	0.79
	40	0.73	0.72	0.76	0.83	0.84	0.86	0.73	0.70
	60	0.67	0.62	0.66	0.72	0.74	0.75	0.63	0.60
	80	0.60	0.56	0.59	0.64	0.67	0.67	0.56	0.53
	Average	0.73	0.68	0.72	0.77	0.79	0.80	0.68	0.67

Figs. 7 and 8 show the PSNR and SSIM variations with noise densities of the Lena image. The visual results of Lena's image are shown at 80% noise densities for AMF, MDBUTMF, AWMF, ASWMF, FASMF, and EHGA in Fig. 9.

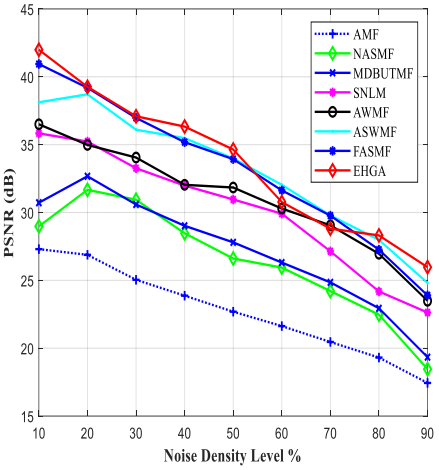


Fig. 7 – *PSNR v/s Noise density plot of Lena Image for spatial filter.*

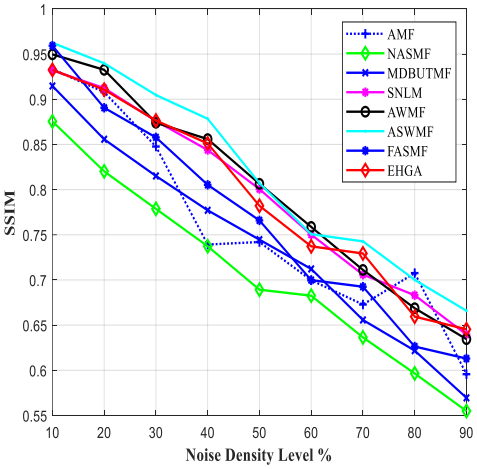


Fig. 8 – *SSIM v/s Noise density plot of Lena Image for spatial filter.*

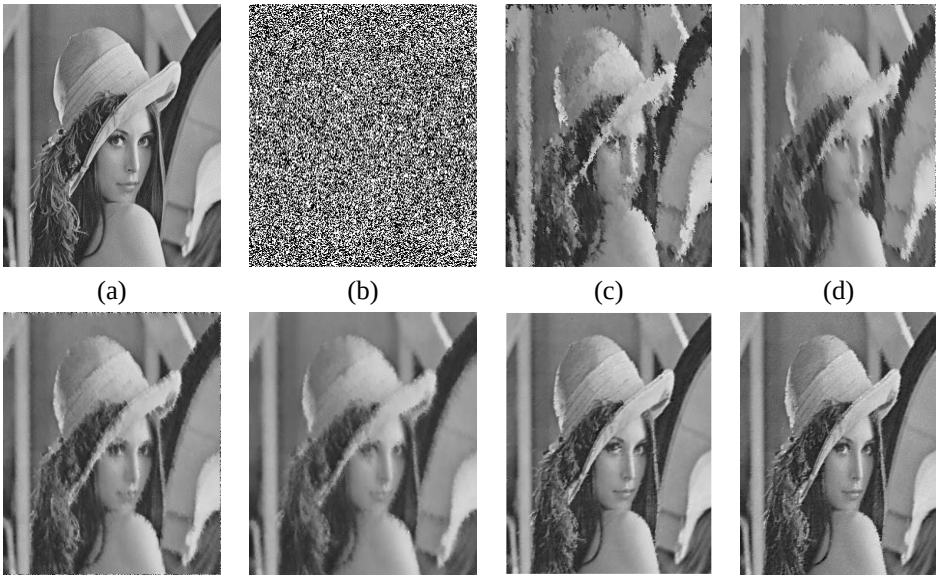


Fig. 9 – *Restoration of Lena Image at* 80% *noise density* (a) *Original Image*; (b) 80% *noisy image*; (c) AMF; (d) MDBUTMF; (e) AWMF; (f) ASWMF; (g) FASMF; (h) EHGA.

Table 6 shows the PSNR result of the Fuzzy domain filter for Boat and bridge images denoised for noise density 10 to 60% and it is plotted in Fig. 10. FBINR Shows a better result than earlier work as it uses an ALD with a fuzzy algorithm to compute weighted mean which helps in better edge restoration. In compared filter, PSNR is better, the reason being the forming of quadrant vector by dividing a large processing window and applying fuzzy rule-based inference for noise detection and using chained median for restoration.

Table 6

PSNR v/s Noise density of Fuzzy domain filter.

Image	Author / Method	Noise Density in %					
		10	20	30	40	50	60
Boat	Luo et al.	31.41	30.01	28.45	27.70	26.79	24.91
	NAFSM	31.46	30.12	28.99	27.60	25.87	24.20
	FBINR	33.56	31.80	29.09	28.57	27.13	24.52
	Nadeem	33.43	31.41	28.72	25.97	24.52	23.76
	Singh	33.54	32.29	30.17	29.00	28.37	28.35
	Liu	32.89	31.70	30.56	28.92	27.66	25.84
Bridge	Luo et al.	28.23	27.55	24.80	23.99	22.42	21.20
	NAFSM	28.63	27.34	25.74	24.19	22.85	21.22
	FBINR	28.54	27.40	25.34	24.14	22.90	21.56
	Nadeem	28.00	27.01	2499	23.52	22.21	20.93
	Singh	30.35	28.67	27.43	25.15	24.62	23.20
	Liu	28.26	27.37	26.15	24.95	23.89	22.48

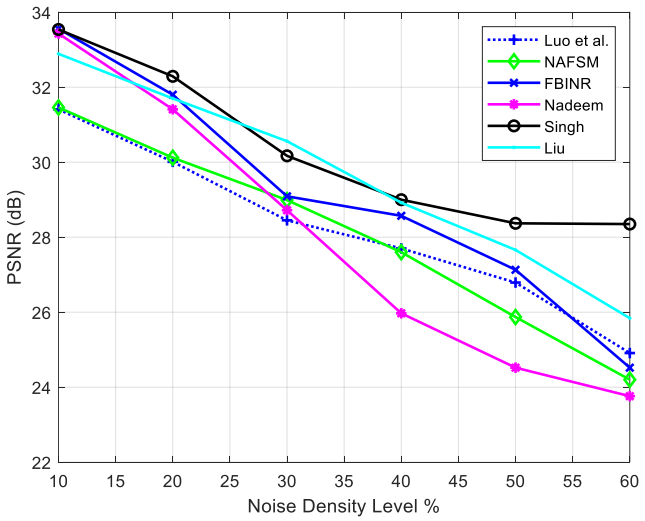


Fig. 10 – *PSNR v/s Noise density plot of Boat Image for fuzzy filter.*

Fig. 11 – *Set 12 image dataset.*

CNN-based methods including ANN, Blind CNN, CNN-PSO, DCNN, DNINR, NLSF-CNN, and SeConvNet are intensively investigated. Denoising Set12 test images provided PSNR and SSIM results for 10% to 90% noise levels in **Table 7** and **Table 8** respectively.

Table 7

PSNR v/s Noise density of CNN domain filter for 4 images of Set-12 data set.

Image	Noise Density (%)	Methods						
		ANN	Blind CNN	CNN-PSO	DCNN	DNINR	NLSF-CNN	SeConv Net
Lena	20	32.28	35.46	38.64	41.59	35.49	36.73	37.48
	40	30.92	29.67	36.89	37.71	33.62	32.69	34.63
	60	21.95	23.38	24.87	30.61	31.24	31.57	33.14
	80	19.6	20.91	22.28	25.52	26.1	25.35	27.48
	Average	26.2	27.95	31.19	33.37	31.16	31.36	32.97
Barbara	20	32.47	35.49	38.51	36.74	35.68	36.69	34.36
	40	31.17	29.99	36.85	32.29	30.84	32.86	27.15
	60	20.02	25.61	26.23	25.15	26.74	22.63	27.99
	80	17.91	24.23	21.83	22	23.42	19.72	24.38
	Average	25.58	29.27	31.11	29.42	29.07	28.4	28.35
Cameraman	20	32.65	35.52	38.38	39.17	33.87	36.66	35.82
	40	31.41	30.29	36.81	32.89	30.29	33.02	27.18
	60	11.15	28.9	24.27	26.59	28.25	30.58	35.34
	80	7.08	24.84	21.92	24.76	26.32	27.04	31.28
	Average	21.2	30.07	31.06	31.38	29.22	31.69	31.21
House	20	31.74	35.54	36.7	38.26	35.7	34.44	36.39
	40	27.47	30.58	32.48	36.77	31.1	27.59	30.61
	60	10.28	28	27.66	24.86	25.91	27.92	28.64
	80	6.3	24.96	23.61	22.37	23.3	25.3	26.25
	Average	19.62	29.96	30.56	31.08	29.01	28.71	30.47

Table 8
SSIM v/s Noise density of CNN domain filter for 4 images of Set-12 data set.

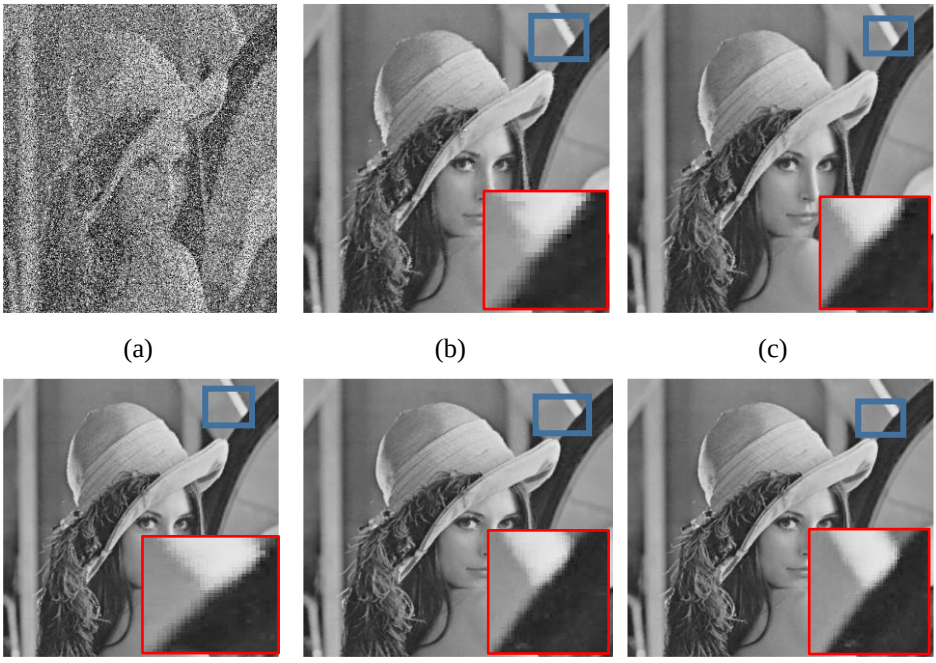
Image	Noise Density (%)	Methods						
		ANN	Blind CNN	CNN-PSO	DCNN	DNINR	NLSF-CNN	INR Net
Lena	20	0.9201	0.9607	0.9661	0.8847	0.9505	0.9565	0.9851
	40	0.9028	0.9318	0.9411	0.8559	0.9078	0.9186	0.9801
	60	0.69	0.67	0.7	0.74	0.75	0.7	0.74
	80	0.69	0.58	0.61	0.67	0.65	0.62	0.69
	Average	0.7963	0.7926	0.8059	0.7859	0.8168	0.8083	0.858
Barbara	20	0.8686	0.9285	0.9724	0.9485	0.9831	0.977	0.9852
	40	0.7574	0.8715	0.9491	0.9139	0.9791	0.958	0.9694
	60	0.6	0.56	0.6	0.65	0.68	0.65	0.68
	80	0.53	0.49	0.52	0.58	0.6	0.57	0.6
	Average	0.7069	0.7301	0.7789	0.7876	0.8204	0.8028	0.8251
Cameraman	20	0.8527	0.958	0.977	0.984	0.975	0.9605	0.9653
	40	0.7685	0.926	0.958	0.973	0.9542	0.9449	0.9496
	60	0.64	0.59	0.63	0.69	0.71	0.68	0.72
	80	0.57	0.53	0.56	0.61	0.64	0.6	0.64
	Average	0.709	0.7682	0.7994	0.83	0.8316	0.8097	0.829
House	20	0.9515	0.9574	0.931	0.9643	0.9555	0.9535	0.9435
	40	0.9231	0.9388	0.8976	0.9535	0.9349	0.9369	0.9256
	60	0.65	0.61	0.65	0.7	0.73	0.7	0.73
	80	0.58	0.54	0.57	0.63	0.65	0.62	0.65
	Average	0.7919	0.7798	0.7765	0.8258	0.8274	0.8154	0.8269

Performance of ANN degrades when noise densities of training and testing image are unequal. PSO algorithms to optimize parameter potential and a median filter to correct pixel false detection give CNN-PSO, DCNN, and SeConvNet greater PSNR than previous methods. Their performance improves at 60% noise density, but performance degrades for images having symmetry facial features which contain a repetitive pattern. In case of blind CNN, the model is complex for noise detection as it is composed of ROAD, CPMD, and EPD statistics. The regression network for noisy pixel reconstruction is trained using noise-free images, and the classifier network utilizes adaptive moment estimation (Adam) instead of stochastic gradient descent, giving DCNN a high PSNR. SeConvNet used a unique multi-term loss function and highly damaged images to train the network without noise density. Non-local information for image pre-processing

and CNN network training patches reduces NLSF-CNN PSNR at high noise density, but it has a limitation of image inpainting. Figs. 12 and 13, zoom Lena and Boat image for better comparison. Fig. 13 of the boat image zooms in on a squared pole to see tiny details. None perform better than NLSF-CNN and SeConvNet. CNN-PSO and SeConvNet handle high noise better than earlier methods (Fig. 14 shows the Lena image PSNR). **Table 9** compares the average PSNR and SSIM of set-12 data set images for different noise levels. The performance of deep learning model is better in comparison to traditional method but at the cost of complex architecture and high computational time.

Table 9
Average PSNR and SSIM of Set-12 Data set.

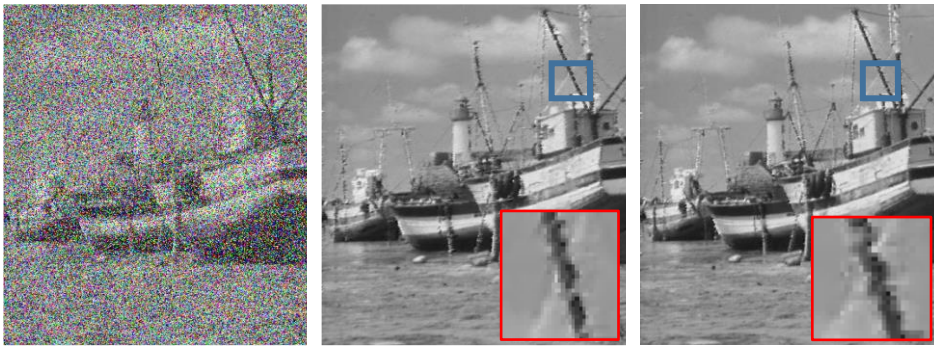
Image	Parameter	Methods						
		ANN	Blind CNN	CNN-PSO	DCNN	DNINR	NLSF-CNN	SeConvNet
Lena	PSNR	26.2	27.95	31.19	33.37	31.16	31.36	32.97
	SSIM	0.79	0.792	0.805	0.7859	0.8168	0.808	0.858
Barbara	PSNR	25.5	29.27	31.11	29.42	29.07	28.4	28.35
	SSIM	0.70	0.730	0.778	0.7876	0.8204	0.802	0.8251
Cameraman	PSNR	21.2	30.07	31.06	31.38	29.22	31.69	31.21
	SSIM	0.70	0.768	0.799	0.83	0.831	0.809	0.829
House	PSNR	19.6	29.96	30.56	31.08	29.01	28.71	30.47
	SSIM	0.79	0.779	0.776	0.8258	0.8274	0.815	0.8269
Boat	PSNR	25.9	27.70	30.91	33.07	30.88	31.08	32.67
	SSIM	0.81	0.814	0.827	0.8080	0.8386	0.830	0.8794
Starfish	PSNR	26.6	30.45	32.36	30.61	30.24	29.55	29.49
	SSIM	0.74	0.769	0.820	0.8291	0.8632	0.844	0.8681
Pepper	PSNR	20.8	29.52	30.49	30.80	28.69	31.11	30.64
	SSIM	0.74	0.802	0.833	0.8634	0.8650	0.843	0.8624
Airplane	PSNR	20.2	30.86	31.48	32.01	29.88	29.57	31.38
	SSIM	0.81	0.803	0.799	0.8506	0.8522	0.839	0.8517
Monarch	PSNR	26.2	28.00	31.24	33.42	31.21	31.41	33.02
	SSIM	0.84	0.842	0.856	0.8361	0.8670	0.858	0.9082
Man	PSNR	26.9	30.78	32.71	30.93	30.57	29.86	29.81
	SSIM	0.77	0.797	0.848	0.8574	0.8918	0.873	0.8968
Couple	PSNR	21.0	29.83	30.81	31.13	28.99	31.44	30.96
	SSIM	0.77	0.830	0.861	0.8920	0.8936	0.871	0.8910
Parrot	PSNR	20.4	31.19	31.81	32.35	30.20	29.89	31.72
	SSIM	0.84	0.831	0.827	0.8791	0.8807	0.8683	0.8802



(a) (b) (c)

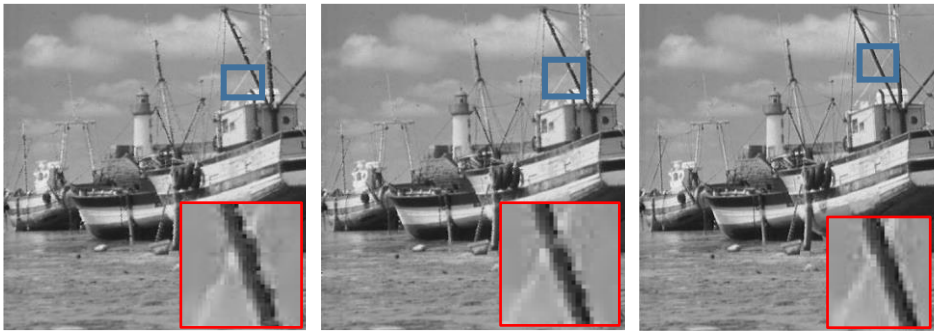
(d) (e) (f)

Fig. 12 – *Restoration of Lena Image:* (a) 50% *noisy image*; (b) *Blind CNN*; (c) *DCNN*; (d) *CNN-PSO*; (e) *NLSF-CNN*; f) *SeConvNet.*



(a) (b) (c)

Fig. 13 – *Restoration of Boat Image* (a) *50% noisy image;* (b) *Blind CNN;* (c) *DCNN.*



(d) (e) (f)

Fig. 13 – (d) *CNN-PSO*; (e) *NLSF-CNN*; (f) *SeConvNet.*

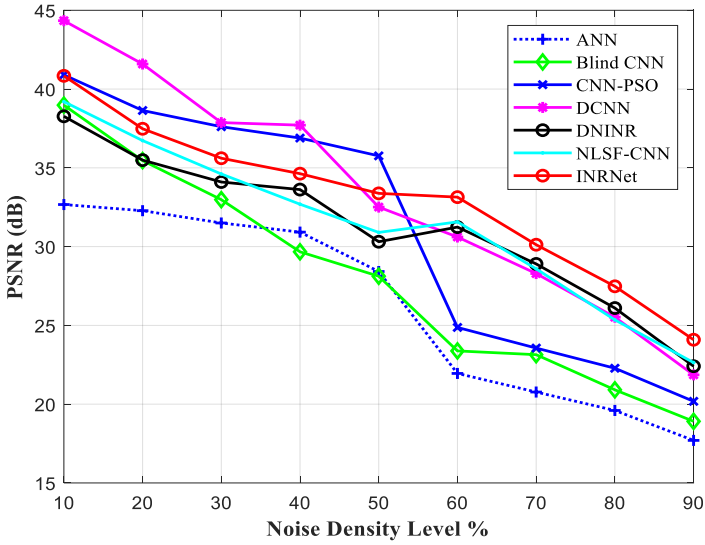


Fig. 14 – PSNR v/s *Noise density plot of Lena Image for CNN-based filter.*

6 Analysis and Future Work

Table 10 provides a comparison of different filter methods used in this study with various features. At last, the gaps associated with various methods and future work to be carried out are outlined.

It is well known that noise in nature is complex in real-world scenarios, and generally, it does not infer any rules. Although many techniques as discussed in the literature attained great success, there are still challenges in the field of image denoising. These include:

– Need to identify correct noisy pixels in any environment.
– The possibility of blurriness and loss of edges at high noise densities.
– The exploration of tuning and optimization problems for better results.
– Need to define a proper fuzzy set of rules for FIS for minimum false detection of pixels.
– Deep CNN requires more memory resources for training.
– To reduce the complexity of the network at the advanced level model.

Table 10
Comparison of different filter methods with various features.

Features	**Non-linear Filter**	**Fuzzy Logic Filter**	**CNN-based Filter**
Algorithm Examples	Median Filter, Min-Max Filter	Fuzzy Median Filter, Fuzzy Switching Filter	SeConvNet, DnCNN
Noise Handling	Effective for low to moderate noise levels	Adaptive to moderate to high noise levels	Superior performance at high noise levels due to data-driven learning
Detail Preservation	Potential blurring of edges and fine details	Enhanced edge preservation through fuzzy rule-based adaptation	Excellent detail preservation via learned feature extraction
Computational Complexity	Low to moderate	Moderate	High due to deep learning architecture
Adaptability	Fixed operation, limited adaptability	Adaptive, leveraging fuzzy rules to modify filtering behaviour	Highly adaptive, learning from training data
Implementation Ease	Simple, straightforward to implement	More complex implementation due to fuzzy rule integration	Complex implementation requires training and high computational resources
Real-time Application	Suitable for real-time applications	Real-time feasible with optimization	Challenging for real-time deployment due to computational demands
Training Requirement	No training required	Requires careful rule-setting	Extensive training on large datasets is necessary

The potential areas of further research in image denoising are but not limited to:

– Combining local and global information of an image to improve the correlation among the pixels.
– Reducing the vagueness in the fuzzy logic-based method by incorporating more degrees of freedom to FIS.
– The use of prior knowledge in deep CNN is an effective way of obtaining highly accurate features and it is implemented by designing or choosing the most favourable loss function for CNN training.
– Including a post-processing step in the training process of neural networks will seem to be an interesting step.

7 Conclusion

This paper conducts a comprehensive study of several denoising techniques of IN that often occur in digital images. Surveying on different denoising methods focusing on one platform will ascertain and provide a framework to understand the findings of other researchers in this field. For better and more comprehensive analysis, this work analyzes and illustrates three popular and prominent denoising models, such as spatial, fuzzy, and CNN models. Each categorization's comparative description and performance analysis are formalized quantitatively and qualitatively. The performance of the SD filter is analyzed in terms of PSNR and SSIM, which are the widely suggested traditional methods of image denoising. These methods often eliminate IN at the expense of blurring fine details and compromising sharp edges. Observation shows the performance of DBUTMWMeF, ASWMeF, FASM, and EHGA are similar at high noise density. Nevertheless, the visual aspect reveals a better performance provided by FASMeF and EHGAas compared to other discussed methods. The performance of CNN models is compared in terms of PSNR and SSIM on the Set12 dataset image. The challenges of the concerned domain are brSSIMed, and some potential areas for future research are suggested. Nevertheless, exploring efficient models that can perform optimally in a real-world scenario in the presence of unknown noisy images still challenges the community. The application of the blind CNN may remain a viable solution to this issue. However, its validity and optimization techniques in the current scenario require further exploration.

8 References

[1] S. Izadi, D. Sutton, G. Hamarneh: Image Denoising in the Deep Learning Era, Artificial Intelligence Review, Vol. 56, 2023, pp. 5929 – 5974.

[2] B. Goyal, A. Dogra, D. C. Lepcha, V. Goyal, A. Alkhayyat, J. S. Chohan, V. Kukreja: Recent Advances in Image Dehazing: Formal Analysis to Automated Approaches, Information Fusion, Vol. 104, April 2024, p. 102151.

[3] W. Luo: Efficient Removal of Impulse Noise from Digital Images, IEEE Transactions on Consumer Electronics, Vol. 52, No. 2, May 2006, pp. 523 – 527.

[4] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, A. Sharma: Image Denoising Review: From Classical to State-of-the-Art Approaches, Information Fusion, Vol. 55, March 2020, pp. 220 – 244.

[5] L. Fan, F. Zhang, H. Fan, C. Zhang: Brief Review of Image Denoising Techniques, Visual Computing for Industry, Biomedicine, and Art, Vol. 2, July 2019, p. 7.

[6] H. Xu, G. Zhu, H. Peng, D. Wang: Adaptive Fuzzy Switching Filter for Images Corrupted by Impulse Noise, Pattern Recognition Letters, Vol. 25, No. 15, November 2004, pp. 1657 – 1663.

[7] A. Hussain, S. M. Bhatti, M. Arfan Jaffar: Fuzzy Based Impulse Noise Reduction Method, Multimedia Tools and Applications, Vol. 60, Vol. 3, October 2012, pp. 551 – 571.

[8] D. Zhai, M. Hao, J. Mendel: Universal Image Noise Removal Filter Based on Type-2 Fuzzy Logic System and QPSO, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 20, No. supp02, October 2012, pp. 207 – 232.

[9] A. E. Ilesanmi, T. O. Ilesanmi: Methods for Image Denoising Using Convolutional Neural Network: A Review, Complex & Intelligent Systems, Vol. 7, June 2021, pp. 2179 – 2198.

[10] T. S. Huang: Two-Dimensional Digital Signal Processing II. Transforms and Median Filters, Springer, Berlin, 1981.

[11] G. Kalaimani, K. Manojkumar, S. S. Kumar: Median Filtering for Removal of Maximum Impulse Noise from Images with a Decision Based Model, Journal of Computational and Theoretical Nanoscience, Vol. 16, No. 2, February 2019, pp. 562 – 567.

[12] S.- J. Ko, Y. H. Lee: Center Weighted Median Filters and their Applications to Image Enhancement, IEEE Transactions on Circuits and Systems, Vol. 38, No. 9, September 1991, pp. 984 – 993.

[13] S. M. Mahbubur Rahman, M. Kamrul Hasan: Wavelet-Domain Iterative Center Weighted Median Filter for Image Denoising, Signal Processing, Vol. 83, No. 5, May 2003, pp. 1001–1012.

[14] J.- H. Wang, L.- D. Lin: Improved Median Filter Using Minmax Algorithm for Image Processing, Electronics Letters, Vol. 33, No. 16, July 1997, pp. 1362 – 1363.

[15] H. Hwang, R. A. Haddad: Adaptive Median Filters: New Algorithms and Results, IEEE Transactions on Image Processing, Vol. 4, No. 4, April 1995, pp. 499 – 502.

[16] Z. Wang, D. Zhang: Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 46, No. 1, January 1999, pp. 78 – 80.

[17] H.- L. Eng, K.- K. Ma: Noise Adaptive Soft-Switching Median Filter, IEEE Transactions on Image Processing, Vol. 10, No. 2, February 2001, pp. 242 – 251.

[18] S. Zhang, M. A. Karim: A New Impulse Detector for Switching Median Filters, IEEE Signal Processing Letters, Vol. 9, No. 11, November 2002, pp. 360 – 363.

[19] P.- E. Ng, K.- K. Ma: A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images, IEEE Transactions on Image Processing, Vol. 15, No. 6, June 2006, pp. 1506 – 1516.

[20] K. S. Srinivasan, D. Ebenezer: A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises, IEEE Signal Processing Letters, Vol. 14, No. 3, March 2007, pp. 189 – 192.

[21] H. Ibrahim, N. S. P. Kong, T. F. Ng: Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images, IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, November 2008, pp. 1920 – 1927.

[22] X. Zhang, Y. Xiong: Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter, IEEE Signal Processing Letters, Vol. 16, No. 4, April 2009, pp. 295 – 298.

[23] F. Duan, Y.- J. Zhang: A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters, IEEE Signal Processing Letters, Vol. 17, No. 7, July 2010, pp. 647 – 650.

[24] S. Akkoul, R. Ledee, R. Leconge, R. Harba: A New Adaptive Switching Median Filter, IEEE Signal Processing Letters, Vol. 17, No. 6, June 2010, pp. 587 – 590.

[25] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, C. H. Premchand: Removal of High Density Salt and Pepper Noise through Modified Decision Based Unsymmetrical Trimmed Median Filter, IEEE Signal Processing Letters, Vol. 18, No. 5, May 2011, pp. 287 – 290.

[26] M. Nasri, S. Saryazadi, H. Nezamabadi-pour: SNLM: A Switching Non-Local Means Filter for Removal of High Density Salt and Pepper Noise, Scientia Iranica, Vol. 20, No. 3, June 2013, pp. 760 – 764.

[27] C. Kalyoncu, Ö. Toygar, H. Demirel: Interpolation-Based Impulse Noise Removal, IET Image Processing, Vol. 7, No. 8, November 2013, pp. 777 – 785.

[28] Z. Miao, X. Jiang: Weighted Iterative Truncated Mean Filter, IEEE Transactions on Signal Processing, Vol. 61, No. 16, August 2013, pp. 4149 – 4160.

[29] N. K. Chaitanya, P. Sreenivasulu: Removal of Salt and Pepper Noise using Advanced Modified Decision Based Unsymmetrical Trimmed Median Filter, Proceedings of the International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, February 2014, pp. 1 – 4.

[30] P. Zhang, F. Li: A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise, IEEE Signal Processing Letters, Vol. 21, No. 10, October 2014, pp. 1280 – 1283.

[31] Z. M. Ramadan: A New Method for Impulse Noise Elimination and Edge Preservation, Canadian Journal of Electrical and Computer Engineering, Vol. 37, No. 1, 2014, pp. 2 – 10.

[32] X. Zhang, F. Ding, Z. Tang, C. Yu: Salt and Pepper Noise Removal with Image Inpainting, AEU - International Journal of Electronics & Communications, Vol. 69, No. 1, January 2015, pp. 307 – 313.

[33] K. Vasantha, T. G. Manjunath, S. N. Raja: A Decision Based Unsymmetrical Trimmed Modified Winsorized Mean Filter for the Removal of High Density Salt and Pepper Noise in Images and Videos, Procedia Computer Science, Vol. 54, 2015, pp. 595 – 604.

[34] Z. Li, Y. Cheng, K. Tang, Y. Xu, D. Zhang: A Salt and Pepper Noise Filter Based on Local and Global Image Information, Neurocomputing, Vol. 159, No. 2, July 2015, pp. 172 – 185.

[35] P.- H. Lin, B.- H. Chen, F.- C. Cheng, S.- C. Huang: A Morphological Mean Filter for Impulse Noise Removal, Journal of Display Technology, Vol. 12, No. 4, April 2016, pp. 344 – 350.

[36] O. S. Faragallah, H. M. Ibrahem: Adaptive Switching Weighted Median Filter Framework for Suppressing Salt-and-Pepper Noise, AEU - International Journal of Electronics and Communications, Vol. 70, No. 8, August 2016, pp. 1034 – 1040.

[37] X. Wang, S. Shen, G. Shi, Y. Xu, P. Zhang: Iterative Non-Local Means Filter for Salt and Pepper Noise Removal, Journal of Visual Communication and Image Representation, Vol. 38, July 2016, pp. 440 – 450.

[38] C.- T. Lu, Y.- Y. Chen, L.- L. Wang, C.- F. Chang: Removal of Salt-and-Pepper Noise in Corrupted Image Using Three-Values-Weighted Approach with Variable-Size Window, Pattern Recognition Letters, Vol. 80, September 2016, pp. 188 – 199.

[39] R. Varatharajan, K. Vasanth, M. Gunasekaran, M. Priyan, X. Z. Gao: An Adaptive Decision Based Kriging Interpolation Algorithm for the Removal of High Density Salt and Pepper Noise in Images, Computers and Electrical Engineering, Vol. 70, August 2018, pp. 447 – 461.

[40] S. Beagum S. Fareed, S. S. Khader: Fast Adaptive and Selective Mean Filter for the Removal of High-Density Salt and Pepper Noise, IET Image Processing, Vol. 12, No. 8, August 2018, pp. 1378 – 1387.

[41] N. Alaoui, A. B. H. Adamou-Mitiche, L. Mitiche: Effective Hybrid Genetic Algorithm for Removing Salt and Pepper Noise, IET Image Processing, Vol. 14, No. 2, February 2020, pp. 289 – 296.

[42] F. Russo, G. Ramponi: A Fuzzy Filter for Images Corrupted by Impulse Noise, IEEE Signal Processing Letters, Vol. 3, No. 6, June 1996, pp. 168 – 170.

[43] K. K. Vin Toh, N. A. Mat Isa: Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction, IEEE Signal Processing Letters, Vol. 17, No. 3, March 2010, pp. 281 – 284.

[44] Y. Wang, J. Wang, X. Song, L. Han: An Efficient Adaptive Fuzzy Switching Weighted Mean Filter for Salt-and-Pepper Noise Removal, IEEE Signal Processing Letters, Vol. 23, No. 11, November 2016, pp. 1582 – 1586.

[45] L. A. Zadeh: The Concept of a Linguistic Variable and its Application to Approximate Reasoning – I, Information Sciences, Vol. 8, No. 3, 1975, pp. 199 – 249.

[46] A. T. Azar: Overview of Type-2 Fuzzy Logic Systems, International Journal of Fuzzy System Applications, Vol. 2, No. 4, October-December 2012, pp. 1 – 28.

[47] M. T. Yildirim, A. Basturk, M. E. Yuksel: Impulse Noise Removal from Digital Images by a Detail-Preserving Filter Based on Type-2 Fuzzy Logic, IEEE Transactions on Fuzzy Systems, Vol. 16, No. 4, August 2008, pp. 920 – 928.

[48] M. A. Khanesar, E. Kayacan, M. Teshnehlab, O. Kaynak: Analysis of the Noise Reduction Property of Type-2 Fuzzy Logic Systems Using a Novel Type-2 Membership Function, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 41, No. 5, October 2011, pp. 1395 – 1406.

[49] M. E. Yuksel, A. Basturk: Application of Type-2 Fuzzy Logic Filtering to Reduce Noise in Color Images, IEEE Computational Intelligence Magazine, Vol. 7, No. 3, August 2012, pp. 25 – 35.

[50] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, N. K. Verma: Adaptive Type-2 Fuzzy Approach for Filtering Salt and Pepper Noise in Grayscale Images, IEEE Transactions on Fuzzy Systems, Vol. 26, No. 5, October 2018, pp. 3170 – 3176.

[51] V. Singh, P. Agrawal, T. Sharma, N. K. Verma: Improved Adaptive Tpe-2 Fuzzy Filter with Exclusively Two Fuzzy Membership Function for Filtering Salt and Pepper Noise, Multimedia Tools and Applications, Vol. 82, No. 13, May 2023, pp. 20015 – 20037.

[52] A. Kumar, S. Kumar, A. Kar: Improved Adaptive Type-2 Fuzzy Detection and Simple Linear Regression-Based Filter for Removing Salt & Pepper Noise, Circuits, Systems, and Signal Processing, Vol. 43, No. 12, December 2024, pp. 7781 – 7819.

[53] Q. Liu, X. Li, J. Yang: Optimum Codesign for Image Denoising Between Type-2 Fuzzy Identifier and Matrix Completion Denoiser, IEEE Transactions on Fuzzy Systems, Vol. 30, No. 1, January 2022, pp. 287 – 292.

[54] S. S. Haykin: Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, Singapore, 1999.

[55] C. Dong, C. C. Loy, K. He, X. Tang: Image Super-Resolution Using Deep Convolutional Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, No. 2, February 2016, pp. 295 – 307.

[56] C. Cruz, A. Foi, V. Katkovnik: Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising, IEEE Signal Processing Letters, Vol. 25, No. 8, August 2018, pp. 1216 – 1220.

[57] I. Turkmen: The ANN Based Detector to Remove Random-Valued Impulse Noise in Image, Journal of Visual Communication and Image Representation, Vol. 34, January 2016, pp. 28 – 36.

[58] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, IEEE Transactions on Image Processing, Vol. 26, No. 7, July 2017, pp. 3142 – 3155.

[59] J. Chen, G. Zhang, S. Xu, H. Yu: A Blind CNN Denoising Model for Random-Valued Impulse Noise, IEEE Access, Vol. 7, September 2019, pp. 124647 – 124661.

[60] H. Y. Khaw, F. C. Soon, J. H. Chuah, C.- O. Chow: High-Density Impulse Noise Detection and Removal Using Deep Convolutional Neural Network with Particle Swarm Optimisation, IET Image Processing, Vol. 13, No. 2, February 2019, pp. 365 – 374.

[61] L. Jin, W. Zhang, G. Ma, E. Song: Learning Deep CNNs for Impulse Noise Removal in Images, Journal of Visual Communication and Image Representation, Vol. 62, July 2019, pp. 193 – 205.

[62] G. Li, X. Xu, M. Zhang, Q. Liu: Densely Connected Network for Impulse Noise Removal, Pattern Analysis and Applications, Vol. 23, February 2020, pp. 1263 – 1275.

[63] S. Sadrizadeh, H. Otroshi-Shahreza, F. Marvasti: Impulsive Noise Removal Via a Blind CNN Enhanced by an Iterative Post-Processing, Signal Processing, Vol. 192, March 2022, p. 108378.

[64] A. Ali Rafiee, M. Farhang: A Deep Convolutional Neural Network for Salt-and-Pepper Noise Removal Using Selective Convolutional Blocks, Applied Soft Computing, Vol. 145, September 2023, p. 110535.

[65] Y. Chen, T. Pock: Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, June 2017, pp. 1256 – 1272.

[66] X. Tang, L. Zhang, X. Ding: SAR Image Despeckling with a Multilayer Perceptron Neural Network, International Journal of Digital Earth, Vol. 12, No. 3, 2019, pp. 354 – 374.

[67] I. H. Sarker: Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective, SN Computer Science, Vol. 2, March 2021, p. 154.

[68] D. P. Mandic, J. A. Chambers: Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, 1st Edition, John Wiley & Sons, Ltd, Hoboken, 2001.

[69] I. H. Sarker: Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions, SN Computer Science, Vol. 2, August 2021, p. 420.

[70] S. Hochreiter, J. Schmidhuber: Long Short-Term Memory, Neural Computing, Vol. 9, No. 8, November 1997, pp. 1735 – 1780.

[71] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio: Generative Adversarial Networks, Proceedings of the 28th Annual Conference on Neural Information Processing Systems, Montreal, Canada, December 2014, pp. 2672 – 2680.

[72] J. Chen, J. Chen, H. Chao, M. Yang: Image Blind Denoising with Generative Adversarial Network Based Noise Modeling, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, USA, June 2018, pp. 3155 – 3164.

[73] A. Aggarwal, M. Mittal, G. Battineni: Generative Adversarial Network: An Overview of Theory and Applications, International Journal of Information Management Data Insights, Vol. 1, No. 1, April 2021, p. 100004.

[74] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio: Deep Learning, The MIT Press, Cambridge, London, 2016.

[75] D. P. Kingma, M. Welling: Auto-Encoding Variational Bayes, arXiv:1312.6114v11 [stat.ML], December 2022, pp. 1 – 14.

[76] A. Makhzani, B. Frey: *k*-Sparse Autoencoders, arXiv:1312.5663v2 [cs.LG], March 2014, p. 1 – 6.

[77] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio: Contractive Auto-Encoders: Explicit Invariance During Feature Extraction, Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue Washington, USA, June 2011, pp. 833 – 840.

[78] X. Lu, F. Li: Fine-Tuning Convolutional Neural Network Based on Relaxed Bayesian-Optimized Support Vector Machine for Random-Valued Impulse Noise Removal, Journal of Electronic Imaging, Vol. 32, No. 1, January 2023, p. 013006.

[79] B. Zhang, G. Zhu, Z. Zhu, H. Zhang, Y. Zhou, S. Kwong: Impulse Noise Image Restoration Using Nonconvex Variational Model and Difference of Convex Functions Algorithm, IEEE Transactions on Cybernetics, Vol. 54, No. 4, April 2024, pp. 2257 – 2270.

[80] Y. Li, T. Xie, D. Mei: Application of Convolutional Neural Networks for Parallel Multi-Scale Feature Extraction in Noise Image Denoising. IEEE Access, Vol. 12, July 2024, pp. 98599 – 98610.