International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 8 Issue 01 Series. III || Jan 2019 || PP 87-98

Satellie and Natural Image Denoising Using Singular Difference Butterworth and Sliding Deep Neural Network

D.Purushothaman[1], Dr.C.Chandrasekar[2]
[1]*Research scholar, Department of Computer Science, Dravidian University kuppam, A.P, India*
and
Assistant Professor, Arunai Engineering College Tiruvannamalai
[2]*Professor, Department of Computer Science, Periyar University, Salem, Tamil Nadu, India*
Corresponding Author: D.Purushothaman

ABSTRACT: *Butterworth Filter, with high adaptive ability, affords a continued fraction expansion method for the noisy image processing. However, the edge effect presents in its operation gives rise to issue, i.e. how to acquire genuine segmentation results to completely eradicate noises from the image. Accordingly, we propose a method to deal with the edge preservation caused by Butterworth Filter in the segmentation of an image signal and then enhanceits denoising performance. This method has two steps. The first step involves a diagonal operation through the Butterworth High Pass Filter model constructed by Singular Value Difference (SVD) with high generalization ability based on the original image pixel value. By applying, Singular Value Difference Butterworth High Pass Filter to the original satellite and natural images, the edge is said to be preserved and also the blurriness is said to be minimizedextracting region of interest with intense sharp images. The second step is the Sliding Neighborhood Deep Neural Network model for image denoising. This is an expansion by the Butterworth Filtering technique with respect to the neighborhood sliding operation to and beyond the edge of the data resulting from the first operation. Applications to remove the noise show that the edge preservation of the Butterworth Filteris improved by the SDB-SDNN methodto meet essential of the genuine segmentation results. They illustrate a good denoising effect of the Butterworth Filter by improving the edge preservation on the basis of the proposed method. Besides, the denoised image also preserves the image details in a sufficient manner by improving the peak signal-to-noise ratio.*

KEYWORDS*: Butterworth Filter, High Pass Filter, Singular Value Difference, Sliding Neighborhood, Deep Neural Network*

Date of Submission: 20-01-2019 Date of acceptance: 06-02-2019

I. INTRODUCTION

Several methods are proposed to denoise satellite and natural images and those images that split into sub components or channels for the purpose of denoising has become very popular. In our day to day life, digital images plays very notable role like they are exploited in monitoring the traffic in an intermittent manner, satellite television, identification of handwritten notes, signature validity and so on. While acquiring digital images, different kinds of noises are said to be introduced. Hence, denoising becomes more noteworthy than any other functions in image processing. Besides, preserving the image details and discarding the noise remains the main objective of image denoising methods.

An optimal design of butterworth filter called, Fractional Order Low pass Butterworth Filters (FOLBFs) using Gravitational Search Algorithm (GSA) was designed in [1] to accomplish best quality with fastest convergence rate. FOFBFs using GSA [1] further exploited the non-linear, non-uniform design to accurately approximate the magnitude response based on third order approximations. Besides by applying continuous-time filter or third order, the response was said to be fast and computationally proficient manner. However, with the application of Butterworth Filters, though the design feasibility was found to be efficient for carrying out pass-band and stop-band regions for signal processing, application of Butterworth Filters for denoising was not focused. In this work, Singular Difference Butterworth Filtering model is designed with the objective of preserving the edges along with minimizing the blurriness of the input image via singular difference so that efficient denoising can be achieved.

One of the most basic issues to be addressed in image processing is image denoising. Amongst several domains, filter provides the magnificent achievement in image denoising due to the sparseness and multifaceted composition. With the acceptance of Butterworth Filter for the last two decades, different methods and mechanisms have been investigated in filter domain. Electro Cardio Gram signals enhancement (ECG signals enhancement) was investigated in [2] to remove the noise like, electrode motion, baseline wander and so on.

This was performed by applying high pass butterworth filters. Besides, edge preserving filters were applied to eradicate the noise. Due to this, the signal-to-noise ratio (SNR) and root mean square error (RMSE) were found to be improved. Though noise were said to be improved, accuracy factor was not said to be concentrated. To lessen the noise and further to improve the rate of accuracy with minimum computational time and overhead, a deep neural network model with sliding neighborhood operation is applied to the edge preserved image.

The main contribution of the paper can be summarized as follows.

1. A novel high-performance Singular Difference Butterworth Filtering algorithm is proposed for filtering the natural and satellite images based on the singular difference, considering the pixel portion rather than the entire image to lessen the noise ratio and therefore improving the denoising accuracy.
2. Two edges – one for magnitude and one for direction are proposed for the proposed algorithm to preserve the edges with minimum blurriness.
3. The proposed Sliding Neighborhood Deep Neural Image Denoising algorithmlessens noise present in the edge preserved images by applying neighbourhood operation with deep insight, hence, decrease the computational time and overhead involved in image denoising.
4. A detailed analysis of the parameter settings (i.e. PSNR, average processing time and computational overhead) is also given to show the impact they may have on the performance of the proposed Singular Difference Butterworth Filtering and Sliding Neighborhood Deep Neural Image Denoising algorithm.

This paper is ordered as follows. Section 2 introduces the related works on image denoising methods applied for different images. Section 3 presents the system model and technical details of SDB-SDNN method. Section 4 describes the experimental configurations and performance evaluations with discussions presented in Section 5. Finally, Section 6 concludes the paper.

II. RELATED WORKS

In [3], the problems related to correcting baseline and noise reduction was investigated by utilizing asymmetric penalty function. With this, not only optimal solution was said to be achieved, but denoising was also said to be performed in an effective manner. However, high quality denoising was not achieved. To address this issue, Structural Similarity optimized Weiner Filter was applied in [4] to execute image denoising. Here, instead of reducing the mean square error, structural similarity between the true and estimated images were maximized. Due to this, regardless of the type of images, superior denoising performance was achieved. Despite improvement in the performance, the denoising speed was said to be compromised. To solve this issue, Sequential Generalization of K-means algorithm was combined with Principal Component Analysis [5]. With this, higher Peak Signal to Noise Ratio was said to be achieved with better denoising.

Redesigningan audio signal from multiple noisy signals necessitates to control the confront that diverge based on the conditions. A two-stage method was designed in [6] by combining best linear combination and applying a post filter to enhance the reconstruction. In [7], a coefficient updated version of Method of Optimal Directions (MOD) Dictionary Learning Algorithm (DLA) was designed with the objective of denoising the image with minimum computational time. In [8], an alternative training scheme with Denoising Auto-encoder (DA) was designed to perform image denoising and blind inpainting, therefore afford complex solutions. An overview of filtering methods for image denoising was presented in [9]. In [10],survey of imperative results related to mean and median absolute deviations was designed.

In [11], removal of noise and therefore classification of normal and abnormal images was designed by applying the median filter and Pixel Surge Model (PSM) and Probabilistic Principal Component Analysis (PPCA) by applying morphological and filter operations. Further, image denoising was performed by applying neural network-based classification. Yet another noise suppression method was designed in [12] by applying Total Variation technique.

With high reduction of reliability with respect to image noise poses severe threats in today’s imaging systems. Besides, in healthcare, the presence of noise poses several limitations in the reliability of medical image data. For example, in radiology using computed tomography (CT), the presence of noise deteriorates the anatomical structure analysis and hence limits the diagnostic applications.

In [13], Quantile Sparse Image was presented, therefore improving the optimization in terms of quantitative measures. Yet another sparse analysis priors was designed in [14] with the objective of providing a trade-off among computational load and performed through constrained dynamic method. However, the noise ratio was said to be compromised with increasing load. To conquer this issue, a local geometry based on directions of gradients and level lines was presented in [15]. Though noise was said to be reduced, the geometrical structure was said to be compromised. To solve this issue, dual projection method and iteratively reweighting method were designed in [16].

During data acquisition, the digital images are captured with the help of sensors. At time, the digital images captured through sensors are frequently found to be affected by noise. Besides, these types of noise is also said to be generated during transmission or image compression. Minimizing the noise and improving the image quality are considered as the main processes for all digital image processing tasks.

In [17], patch-based denoising methods were introduced and were estimated with both qualitative and quantitative factors, therefore improving the computational time and quality. Yet another non-local means filter was investigated in [18] for image denoising with natural images. However, with the improper selection of threshold, image denoising was also said to be compromised. To address this issue, a threshold-free denoising procedure was designed in [19] by localizing the low and high frequency domains. A survey of image denoising algorithms was presented in [20].

Distinguished from prior works, we establish a method of image denoising system to strengthen the image denoising accuracy with minimum time and overhead based on the deep neural network model in this paper.

III. METHODOLOGY

During the process of image acquisition, certain amount of noises are said to be combined with the pure image. Due to this, the edges are said to be compromised, further creating blurriness in the image and hence image denoising is said to occur. In this work, to preserve edge and minimize blurriness of the input image along with the image denoising, a Singular Difference Butterworth and Sliding Deep Neural Network (SDB-SDNN) method is investigated. Figure 1 given below shows the block diagram of SDB-SDNN method.

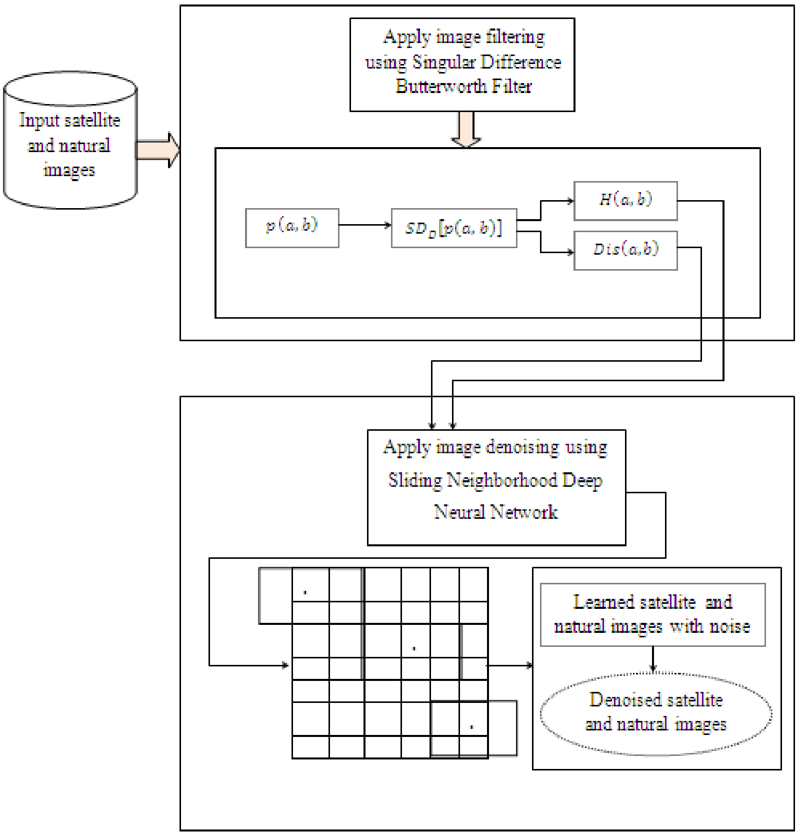
Input satellite and natural images
Apply image filtering using Singular Difference Butterworth Filter
$p(a,b)$
$SD_D[p(a,b)]$
$H(a,b)$
$Dis(a,b)$
Apply image denoising using Sliding Neighborhood Deep Neural Network
Learned satellite and natural images with noise
Denoised satellite and natural images

Figure 1 Block diagram of SDB-SDNN

As shown in the figure, the SDB-SDNN method incorporates two steps. The first step is to preserve the edges with minimum blurriness using Singular Difference Butterworth Filter model. The second step is image denoising using Sliding Neighborhood Deep Neural Network model. The main aim here is to inspect their internal structures and identify their strengths and weaknesses to enable the proposal of image denoising with input satellite and natural images that integrates the strengths of these two models.

3.1 Singular Difference Butterworth Filter model

In this section, Singular Difference Butterworth Filter is applied to the input satellite and natural images with the objective of retaining or preserving the edges and reducing the blurriness. Usually, while applying filter, edges have to be retained or preserved. Here, the edges represent the boundaries between objects or images. In this work, the estimates of Edge Singular Difference Butterworth Magnitude and Edge Singular Difference Butterworth Direction from Singular Difference Butterworth Filter results obtained by appropriate scaling are measured. Figure 2 shows the flow diagram of Singular Difference Butterworth Filter model.

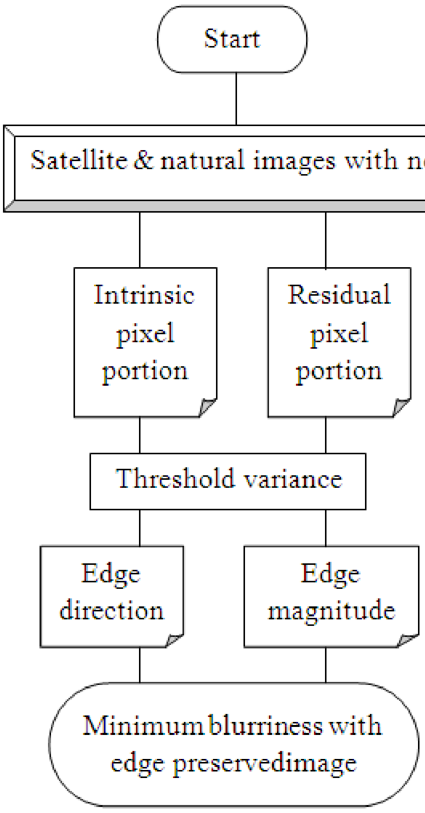

Figure 2 flow diagram of Singular Difference Butterworth Filter

As shown in the figure, the design of Singular Difference Butterworth Filter model includes four different steps. The steps of the proposed Singular Difference Butterworth Filter model are given below.

a. Generate the Singular Difference model with different intrinsic portions of actual and real images with noise variance.

b. Use the Butterworth Filter to segment the new noisy image into its low-pass filters and high-pass filters, and then calculate the size and estimate the variance of the noise.

c. Obtain Singular Difference by applying the noise variance and image size for corresponding diagonal portions of the image.

d. Finally, obtain the edge direction and magnitude with the objective of retaining the edges and reduce the blurriness.

With the above four steps, let us consider the real image regarded as a mixture of the noise and pure.This is expressed as given below.

$$q(a,b) = p(a,b) + s(a,b) \qquad (1)$$

From the above equation (1), '$q(a,b)$' represents the real image with noises, '$p(a,b)$' represents the actual image and '$s(a,b)$' denotes the noise present in the image. Then, the actual image and noisy images are further formulated as given below.

$$p(a,b) = \sum_{i=1}^{n} p_i(a,b) + r_p(a,b) \qquad (2)$$

$$s(a,b) = \sum_{i=1}^{n} s_i(a,b) + r_s(a,b) \qquad (3)$$

From the above equation (2) and (3) '$p_i(a,b)$' and '$s_i(a,b)$' represents the intrinsic pixel portions of the actual image and real image with noises respectively. Next, '$r_p(a,b)$' and '$r_s(a,b)$' represents the residual pixel portions of the actual image and real image with noises respectively. Then by substituting the above two equations (2) and (3), the equation (1) is written as given below.

$$q(a,b) = \sum_{i=1}^{n} p_i(a,b) + r_p(a,b) + \sum_{i=1}^{n} s_i(a,b) + r_s(a,b) \qquad (4)$$

Then, with the real image with noise, using the Butterworth Filter, a satellite or natural image is said to be segmented into its low-pass and high-pass filters. The low-pass filterspart represents the actual pixel portions of an image, while the three high-pass filter parts represent the detailed information of an image. They are the horizontal 'H', vertical 'V', and diagonal parts 'D' respectively. In this work, the diagonal parts are considered for further processing because the boundaries between objects are of higher interest. Hence, new Butterworth coefficients are evaluated by thresholding the original Butterworth coefficients. In the proposed work, the thresholds are determined by utilizing the variance 'σ' of the noise. It is formulated as given below.

$$\sigma = \frac{Avg\left(H_B^n\right)}{0.674} \qquad (5)$$

From the above equation (5), the variance 'σ' of the noise denotes the ratio of average of the Butterworth coefficients [1] with respect to the diagonal portion of the image (i.e. denoting the high pass butterworth filter). The numerator denotes the mean absolute deviation [3].

$$SD_D[p(a,b)] = sV_D(\sigma) f_n\left[Size\left(p(a,b)\right)\right] \qquad (6)$$

From the above equation (6), the singular difference 'SD' with respect to the actual image '$p(a,b)$' is obtained by applying the noise variance '$sV_D(\sigma)$' for the corresponding diagonal portion of the image 'D' and the size of the actual image '$Size\left(p(a,b)\right)$'. Followed by the identification of singular difference, the edge magnitude and edge direction are mathematically evaluated as given below.

$$EdgeDirection = Hp(a,b) = \frac{1}{1+\left(\sqrt{2}-1\right)\left[\frac{Disp\ (a,b)}{Dis\ _0}\right]^{2D}} \qquad (7)$$

From the above equation (7), 'D' represent the order and in the proposed work, it represents the diagonal portion of the image with 'Dis_0' denoting the specified distance from the origin with respect to the actual image '$p(a,b)$'.

$$EdgeMagnitude = Disp(a,b) = \sqrt{a^2 + b^2} \qquad (8)$$

The pseudo code representation of Singular Difference Butterworth Filter is given below.

```
Input: actual image '$p(a,b)$', noisy image'$s(a,b)$', intrinsic pixel portions of the actual image '$p_i(a,b)$', intrinsic pixel portions of the real image with noises '$s_i(a,b)$', residual pixel portions of the actual image '$r_p(a,b)$', residual pixel portions of the real image with noises '$r_s(a,b)$'
Output: Edge Preserved Image '$ep_i = ep_1, ep_2, \ldots, ep_n$'
Begin
    For each actual and noisy image real image with noises
        Evaluate real image with noises using (1)
        Evaluate actual image and real image with noises using (2) and (3)
        Measure variance '$\sigma$' using (5)
        Measure singular difference using (6)
        Evaluate edge direction using (7)
        Evaluate edge magnitude using (8)
    End for
End
```

Algorithm 1 Singular Difference Butterworth

As given in the above Singular Difference Butterworth Filtering algorithm, for each actual and noisy real image with noises, the purpose of this algorithm is to not only retain the edges but also minimize the blurriness. After training the model using satellite and natural images (that includes large set noisy and denoised images), magnificent results that are comparable to the results of Singular Difference Butterworth Filtering algorithmis said to be arrived at in much less computational time.

3.2 Sliding Neighborhood Deep Neural Network model

Once the edges of the input images have been retained with minimum blurriness, the next step in the design is to remove the noise present in the images. In this work, as the objective remains in concentrating on a learning based model, the selection of image denoising model depends on the complexity of the function to be learned. As this work deals with satellite and natural images, the model selected have to accommodate all the essential image features, competent of learning and extracting region of interest with intense sharpness from the image. Hence, Sliding Neighborhood Deep Neural Network model has been selected that has the capability of modeling high-level abstractions in image features.

In Sliding Neighborhood Deep Neural Network model, image denoising is executed with single pixel operation rather than processing the entire image at a single sequence. The Sliding Neighborhood Deep Neural Network operation is an operation that is conducted with a pixel at a time with deeper insights according to the edge preserved images as input. Here, the value of the output pixel is measured by applying the corresponding input pixel's edge-preserved neighborhood value.

The edge-preserved pixel's neighborhood represents certain set of pixels, based on their locations corresponding to that pixel, also referred to as the midpoint pixel. Besides, the edge-preserved pixel's neighborhood corresponds to rectangular section by traversing from one portion to the next in an image matrix with the neighborhood section traversing the same order. The midpoint pixel represents the real pixel in the edge preserved image (with lesser blurriness) being processed by the operation.

If the neighborhood possesses odd number of rows and columns, the midpoint pixel is said to be the center of the neighborhood. On the other hand, if one of the dimensions has even representation, the midpoint pixel is left of midpoint or just above midpoint. Finally, an activation function is applied to the midpoint value to perform image denoising with minimum error. Figure 3 shows the block diagram of Sliding Neighborhood Deep Neural Network model.

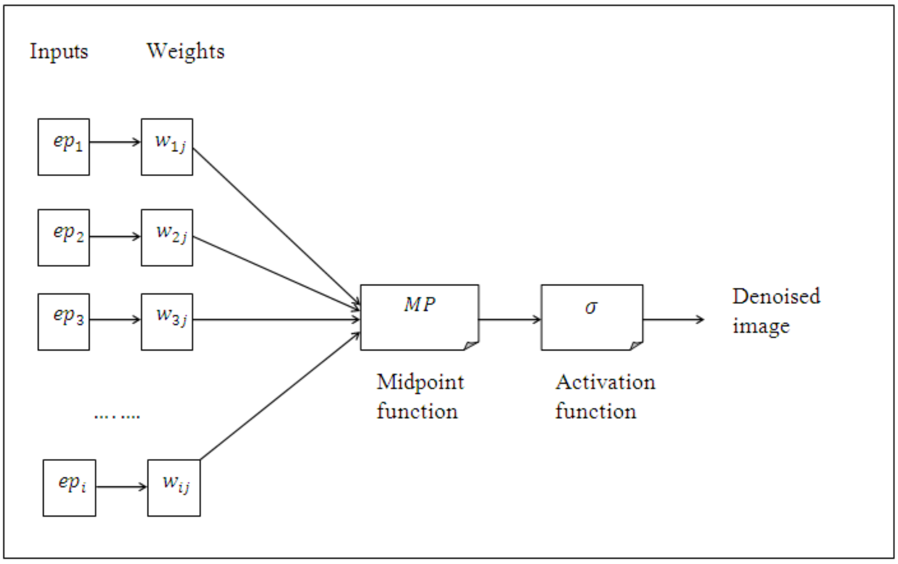

Figure 3 Block diagram of Sliding Neighborhood Deep Neural Network model

As shown in the figure, consider a neural network with 'i' input, 'j' output units with any number of hidden units. When the input pattern 'ep_i' from a training set is presented to the network, it generates an output 'q_i' different in general from the target 't_i'. In the corresponding hidden layer, the value from each input unit is multiplied with the weight 'w_{ij}' and the weighted average of input patterns '$WAVG(ep_i)$' are derived. It is formulated as given below.

$$WAVG(ep_i) = \frac{(w_{1j} + w_{2j} + \ldots + w_{ij})}{n} \qquad (9)$$

Besides, the error between 'q_i' and 't_i' has to be minimized for effective denoising of satellite and natural images. It is formulated as given below.

$$E = \frac{1}{2}\sum_{i=1}^{n}(q_i - t_i)^2 \qquad (10)$$

From the above equation (10), the error 'E' is used to measure the change in hidden to output layer weights and the change in input to hidden layer weights, with the objective of minimizing the global error rate. This process is reiterated foreach pixel in the input image. With the error 'E' evaluated as given in the above equation (10), to summarize how an edge-preserved sliding neighborhood operation with deep neural network is performed, let us consider an edge preserved image with minimum blurriness obtained from the above sections as input. Here, a single pixel is chosen rather than entire image. Then, the pixel's neighborhood value is obtained. Followed by which the midpoint pixel is mathematically evaluated as given below.

$$MP = FLOOR\left(([i,j] + 1)/2\right) \qquad (11)$$

From the above equation (11), for any '(i,j)' neighborhood, the midpoint pixel 'MP' return the greatest integer value '$FLOOR$' with 'i' rows and 'j' columns respectively. Followed by the evaluation of the midpoint, an activation function is applied to the values of the pixels in the neighborhood. This is mathematically evaluated as given below.

$$\sigma(p) = \frac{1}{1 + e^{-p}} \qquad (12)$$

From the above equation (12), the activation function '$\sigma()$' for image 'p' with 'a' and 'b' representing the rows and columns is measured according to the error function 'e' over output neurons for each image. The pseudo code representation of Sliding Neighborhood Deep Neural Image Denoising is given below.

```
Input: Edge Preserved Image '$ep_i = ep_1, ep_2, \dots, ep_n$',
Output: Denoised image
Begin
    For each Edge Preserved Image '$ep_i$'
        Repeat
            Measure the weighted average of input patterns using equation (9)
            Measure the error function using equation (10)
            Measure midpoint pixel using equation (11)
            Evaluate the activation function using equation (12)
        Until (each pixel in the input  image is processed)
    End for
End
```

Algorithm 2 Sliding Neighborhood Deep Neural Image Denoising

As given in the above Sliding Neighborhood Deep Neural Image Denoising algorithm, each Edge Preserved Image 'ep_i' given as input, the objective here remains in minimizing the noise present in the image by applying the sliding neighborhood operation. Initially, the weight is derived by measuring the average of input patterns. Followed by which, error function is obtained. Then, the pixel's neighborhood value is obtained by measuring the midpoint pixel. Finally, the activation function is applied to the midpoint pixel value and this process is repeated for all pixels in the input image with minimum error rate.

IV. EXPERIMENTAL SETTINGS

In this paper, the satellite [22] and natural images [21] database is exploited to verify average processing time and image denoising accuracy of the experimental method. The average processing time and image denoising accuracy of the test method was compared with (Fractional Order Low pass Butterworth Filters (FOLBFs) using Gravitational Search Algorithm (GSA)) [1] and Electro Cardio Gram signals enhancement (ECG signals enhancement) [2]. The evaluation is established by using the following parameters:

First, to measure the image denoising accuracy, the noise level has to be measured. Therefore, the noise factor is analyzed by measuring the PSNR. PSNR measures the quality of reconstruction of denoised signal. The signal here refers to the original data and the noise is the error introduced by denoising. It is mathematically evaluated as given below.

$$PSNR = 10\ log10\left(\frac{L^2}{MSE}\right) \qquad (13)$$

From the above equation (13), 'L' represents the peak signal level, whereas 'MSE' represents the Mean Squared Error [23], which is the pixel value of the image. The value of MSE is evaluated as given below.

$$MSE = \left[\frac{1}{k*k}\right] \sum_{i=1}^{n} \sum_{j=1}^{n} [p(i-j) - s(i-j)]^2 \qquad (14)$$

Where '$p(i,j)$' and '$s(i,j)$' are the pixel intensity values of the original image and the noisy image respectively and 'k' symbolizes the size of input satellite and natural images. Next, the average processing time is mathematically evaluated as given below.

$$APT = Time\ (PP) + Time\ (ID) \quad (15)$$

From the above equation (15), the average processing time 'APT', refers to the time consumed during preprocessing 'PP' and the time consumed during image denoising 'ID'. It s measured in milliseconds (ms). Here, the time consumed during preprocessing includes thetime consumed to extract the image with respect to magnitude ad direction. Besides, the time consumed during image denoising involves the time consumed for evaluating the midpoint function and activation function respectively. Finally, the computational overhead is mathematically evaluated as given below.

$$CO = k * Storage\ (PP) + Storage\ (ID) \quad (16)$$

From the above equation (16), the computational overhead 'CO' is measured according to the samples provided as input 'k' and the storage consumed to carry out the preprocessing task '$Storage\ (PP)$' and image denoising task '$Storage\ (ID)$' respectively. It is measured in terms of kilobytes (KB). Lower communication overhead improves the efficiency of the method.

V. DISCUSSION

In this section, the performance of the SDB-SDNN method is evaluated and compared it with the (Fractional Order Low pass Butterworth Filters (FOLBFs) using Gravitational Search Algorithm (GSA)) [1] and Electro Cardio Gram signals enhancement (ECG signals enhancement) [2].

5.1 Peak Signal-to Noise Ratio

In this section, we would like to evaluate the peak signal-to noise ratio of SDB-SDNN method and compare it with the FOLBFs using GSA [1] and ECG signals enhancement [2] based on the number of samples provided as input for image denoising in the comparison experiments. The sample calculations and tabulation column is represented as given below.

Sample calculations

- **Proposed SDB-SDNN**: With satellite image provided as input for conducting experiments, the value of MSE being '280', the PSNR is measured. Accordingly, for '10' images given as input for conducting experiments, the PSNR is found to be '$95.9dB$'.

$$PSNR\ (using\ satellite\ image) = 10\log(10) * \frac{255}{280} = 9.59dB$$

In a similar manner, with natural image given as input, the value of MSE being '265', the PSNR is measured. Then, for '10' images given as input, the PSNR is found to be '$98.3dB$'.

$$PSNR\ (using\ natural\ image) = 10\log(10) * \frac{255}{265} = 9.83dB$$

- **FOLBFs using GSA**: With satellite image provided as input for conducting experiments, the value of MSE being '312', the PSNR is measured for FOLBFs using GSA [1]. Accordingly, for '10' images given as input for conducting experiments, the PSNR is found to be '$91.2dB$'.

$$PSNR\ (using\ satellite\ image) = 10\log(10) * \frac{255}{312} = 9.12dB$$

In a similar manner, with natural image given as input, the value of MSE being '320', the PSNR is measured for FOLBFs using GSA [1]. Then, for '10' images given as input, the PSNR is found to be '$90.1dB$'.

$$PSNR\ (using\ natural\ image) = 10\log(10) * \frac{255}{320} = 9.01dB$$

- **ECG signals enhancement**: With satellite image provided as input for conducting experiments, the value of MSE being '345', the PSNR is measured for ECG signals enhancement. Accordingly, for '10' images given as input for conducting experiments, the PSNR is found to be '$86.6dB$'.

$$PSNR\ (using\ satellite\ image) = 10\log(10) * \frac{255}{345} = 8.68dB$$

In a similar manner, with natural image given as input, the value of MSE being '360', the PSNR is measured for ECG signals enhancement. Then, for '10' images given as input, the PSNR is found to be '$85.0dB$'.

$$PSNR\ (using\ natural\ image) = 10\log(10) * \frac{255}{360} = 8.50dB$$

The tabular representation for measuring peak signal to noise ratio (PSNR) with satellite and natural images provided as input is given below.

Table 1 Comparison among the designed SDB-SDNN in terms of PSNR

No. of images	**PSNR (dB) [using satellite images]**			**PSNR (dB) [using natural images]**		
	SDB-SDNN	FOLBFs using GSA	ECG signals enhancement	SDB-SDNN	FOLBFs using GSA	ECG signals enhancement
10	95.9	91.2	86.8	98.3	90.1	85.08
20	91.5	87.5	84.1	93.5	88.5	82.4
30	93.5	82.5	80.2	95.8	82.3	80.1
40	95.8	80.4	78.4	97.2	85.8	83.5
50	82.4	75.2	72.1	85.6	82.4	80.2
60	85.6	78.9	73.5	88.9	83.5	80.5
70	88.9	90.3	82.5	90.2	85.8	82.3
80	90.3	82.4	80.2	92.4	88.2	83.1
90	92.4	85.8	81.4	94.5	90.1	88.5
100	88.5	86.1	83.2	90.1	87.6	84.2

Table 1 given above shows the PSNR of all the three methods using satellite and natural images. This experiment is set to verify the PSNR efficiency with respect to number of samples provided as input for image denoising. In this experiment environment, we combine two different functions, i.e. edge magnitude and edge direction to achieve the high efficiency PSNR rate. The PSNR is compared with FOLBFs using GSA [1] and ECG signals enhancement [2] by capturing the same amount of samples. As illustrated in the above table, the experimental results of PSNR rate for different number of samples with same size (e.g. 10, 20, 30,….,100) in the preprocessing phase or filtering is considered. From the experiment, it is inferred that the PSNR rate is inversely proportional to the number of sample images provided as input and to handle the same number of samples for image denoising, SDB-SDNN method generates minimum error than [1] and [2], both using satellite and natural images. Hence, as given in the table, the PSNR rate as is inversely proportional, the occurrence of noise ratio depends only on the type of image and image size and not on the number of sample images considered for experimentation. In other words, the PSNR rate for image denoising in SDB-SDNN method is higher than those in [1] and [2]. This is because of the application of Singular Difference Butterworth Filter in the SDB-SDNN method. The reasons for that are twofold. First, the former two methods performs the preprocessing for noise removal using continued fraction expansion methods with the aid of Butterworth filer mechanism whereas in SDB-SDNN method, with the objective of retaining the edges and reduce the blurriness, both edge magnitude and edge direction were considered, therefore reducing the noise ratio. Second, for performing filtering, the existing [1] and [2] methods applied single edge along with guided filter to remove the noise, which is said to be compromised in case of the large samples, whereas in SDB-SDNN method, edges are said to be preserved with minimum blurriness due to the application of Singular Difference Butterworth Filter. This in turn improves the PSNR rate using SDB-SDNN method by 8% compared to [1] and 13% compared to [2] with satellite images as input and 7% compared to [1] and 12% compared to [2] with natural images as input.

5.2 Average processing time

Followed by PSNR rate, second the average processing time using the three different methods is analyzed based on the sample images provided as input. The sample calculations and tabular representation of average processing time is given below.

Sample calculations

- **Proposed SDB-SDNN**: With the preprocessing and image denoising time being ‘$0.031ms$’ and ‘$0.128ms$’ for single satellite image, the average processing time using satellite image is given below.

$$APT\ (using\ satellite\ image) = 10 * [0.031ms + 0.128ms] = 1.59ms$$

In a similar manner, the preprocessing and image denoising time being ‘$0.036ms$’ and ‘$0.133ms$’ for single natural image, the average processing time using natural image is given below.

$$APT\ (using\ natural\ image) = 10 * [0.036ms + 0.133ms] = 1.69ms$$

- **FOLBFs using GSA**: With the preprocessing and image denoising time being ‘$0.038ms$’ and ‘$0.141ms$’ for single satellite image, the average processing time using satellite image is given below.

$$APT = 10 * [0.038ms + 0.141ms] = 1.79ms$$

In a similar manner, the preprocessing and image denoising time being '$0.042ms$' and '$0.165ms$' for single natural image, the average processing time using natural image is given below.

$$APT = 10 * [0.042ms + 0.165ms] = 2.07ms$$

- **ECG signals enhancement**: With the preprocessing and image denoising time being '$0.048ms$' and '$0.163ms$' for single satellite image, the average processing time using satellite image is given below.

$$APT = 10 * [0.048ms + 0.163ms] = 2.11ms$$

In a similar manner, the preprocessing and image denoising time being '$0.052ms$' and '$0.175ms$' for single natural image, the average processing time using natural image is given below.

$$APT = 10 * [0.052ms + 0.175ms] = 2.27ms$$

Table 2Comparison among the designed SDB-SDNN in terms of Average Processing Time

No. of images	**Average Processing Time (ms) [using satellite images]**			**Average Processing Time (ms) [using natural images]**		
	SDB-SDNN	FOLBFs using GSA	ECG signals enhancement	SDB-SDNN	FOLBFs using GSA	ECG signals enhancement
10	1.59	1.79	2.11	1.69	2.07	2.27
20	2.85	2.95	3.15	3.15	3.35	4.55
30	4.53	5.55	6.85	5.85	6.16	6.25
40	6.85	7.35	7.85	7.35	7.85	8.55
50	9.55	10.25	11.35	10.25	11.35	12.45
60	9.65	11.15	13.55	10.85	12.45	14.15
70	10.86	12.35	14.52	11.45	14.55	16.32
80	11.35	13.45	15.66	14.35	15.35	17.85
90	12.45	14.55	16.15	15.55	17.35	19.25
100	15.35	16.55	18.35	15.85	19.42	22.35

The targeting results of average processing time using SDB-SDNN method is compared with two state-of-the-art methods [1] and [2] in table 2 is presented for comparison based on the varied number of satellite and natural sample images as input. The SDB-SDNN method differs from the [1] and [2] in that we have applied Sliding Neighborhood Deep Neural Image Denoising algorithm for image denoising. With simulation being conducted for 10 sample images provided as input using satellite and natural images, Singular Difference Butterworth Filtering was first applied. With this filter, the singular difference was exploited instead of the entire image for extracting the edge magnitude and direction, therefore lessens the computational time. Further by considering the noise variance and size of the image for further processing, diagonal portions of the images were further considered because of which, the boundaries were said to be extracted in an efficient manner. Finally, by applying the Sliding Neighborhood Deep Neural Network for image denoising with the edge magnitude and edge direction portions reduces the noise factor. With the neighborhood factor considered for image denoising, the average processing time for image denoising is said to be reduced. In this way, the average processing time is found to be reduced by 11% compared to [1] and 22% compared to [2] using satellite images. In a similar manner, the average processing time is found to be reduced by 89% compared to [1] and 22% compared to [2] using natural images.

5.3 Computational overhead

Finally, the computational overhead using three different methods is analyzed. In SDB-SDNNmethod, computational overhead measures the storage consumed to carry out the preprocessing and image denoising. The computational overhead is evaluated in kilobytes (MB). The sample calculations along with the graphical representation are given below.

Sample calculations

- **Proposed SDB-SDNN**: With '10' samples considered as experimentation using satellite images, the storage required for preprocessing was found to be '$12KB$' and the storage required for image denoising was found to be '$15KB$' using SDB-SDNN. Then the computational overhead is as given below.

$$CO = 10 * [12KB + 15KB] = 270KB$$

- **FOLBFs using GSA**: With '10' samples considered as experimentation using satellite images, the storage required for preprocessing was found to be '$16KB$' and the storage required for image denoising was found to be '$20KB$'. Then the computational overhead is as given below.

$$CO = 10 * [16KB + 20KB] = 360KB$$

- **ECG signals enhancement**: finally, with '10' samples considered as experimentation using satellite images, the storage required for preprocessing was found to be '$21KB$' and the storage required for image denoising was found to be '$25KB$' using ECG signals enhancement. Then the computational overhead is as given below.

$$CO = 10 * [21KB + 25KB] = 460KB$$

Computational overhead incurred during filtering is one of the challenges to be addressed for image denoising systems. With the increase in the number of sample images, minimization of computational overhead cannot be attained. But, optimization can be achieved. The comparison of computational overhead for SDB-SDNN method is measured and compared with [1] and [2] and is plotted in figure 4. The results reported in the figure confirm that with the increase in the number of satellite images given as input, the computational overhead also gets increased.

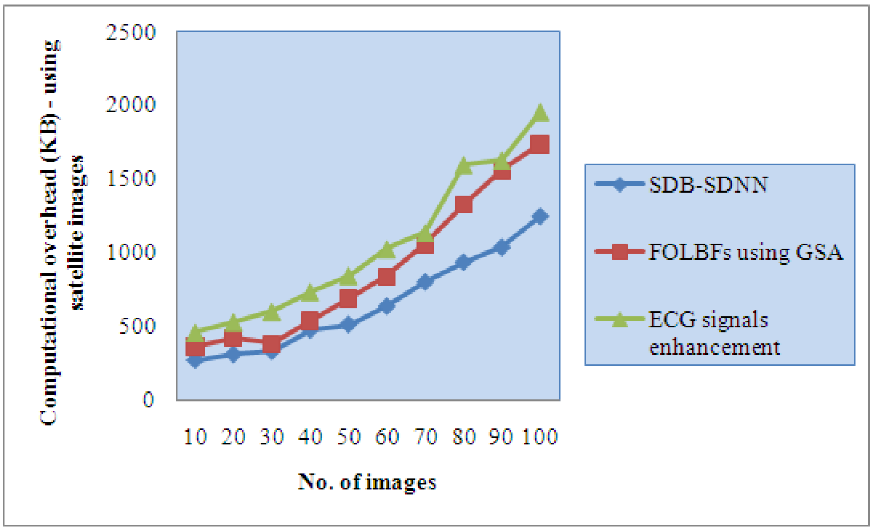

Figure 4 Computational overhead comparisons for SDB-SDNN, FOLBHs using GSA and ECG signals enhancement

Figure 4 given above shows the comparison performance of computational overhead for 150 different satellite images. As a result, 150 different satellite images are observed in x axis and computational overhead is observed in the y axis. With increase in the number of samples using the satellite images with different sizes, the computational overhead for image denoising also increases. Due to this, the computational overhead increases with the increase in the number of input images. As a simulation, with '10' satellite samples considered for experimentation, the storage required was found to be '$270KB$' using SDB-SDNN method, '$460KB$' when applied with FOLBHs using GSA [1] and '$460KB$' using ECG signals enhancement. However, performance analysis on an average found SDB-SDNN method comparatively better than [1] and [2]. This is because of the sliding neighborhood operation using the Sliding Neighborhood Deep Neural Image Denoising algorithm. By applying Sliding Neighborhood Deep Neural Image Denoising algorithm, weight is measured by the average of input patterns that has already being filtered by applying the Singular Difference Butterworth. Here, the performance of filtering is performed via edge magnitude and edge direction that helps in preserving the edge and minimizing the blurriness rate. Though the edges were preserved using [1] and [2], but, the correlation factor were not considered during noise removal. Hence, minimum amount of noise is said to be found in the edges, therefore compromising the image denoising factor. Furthermore, by applying singular difference, single pixel operationis considered rather than, the entire image at a single sequence. As the result, the complexity involved in computation is said to be reduced by applying the SDB-SDNN method. As a result, the computational overhead for image denoising is found to be comparatively lesser using SDB-SDNN method by 24% compared to [1] and 38% compared to [2].

VI. CONCLUSION

In this paper, Singular Difference Butterworth and Sliding Deep Neural Network (SDB-SDNN) method is proposed. The method consists of two parts, image filtering and image denoising. To start with, initially, image filtering is performed by applying the Singular Difference Butterworth Filter model. Here, singular difference is evolved intrinsic portions of the actual image and real images with noise variance. Followed by which, Butterworth Filter is applied to the singular difference resultant values for the purpose of segmenting the noisy image into its low-pass filters and high-pass filters. Here, the high-pass filter diagonal portions of the images were exploited for extracting edge magnitude and edge direction, with which the edge preserved image with minimum blurriness was said to be produced. Next, image denoising was performed with the edge preserved minimum blurriness image by applying the Sliding Neighborhood Deep Neural Image Denoising algorithm. With the application of this algorithm, midpoint pixel value was evaluated by applying pixel's neighborhood value, therefore ensuring minimum error or peak signal to noise ratio. The SDB-SDNN are implemented to validate the proposed method. Extensive experiments conducted demonstrate the effectiveness and robustness of the proposed method in terms of PSNR, average processing time and computational overhead efficiency.

REFERENCES

[1]. Shibendu Mahataa, Suman Kumar Sahab, Rajib Kara, Durbadal Mandala, " Optimal design of fractional order low pass Butterworth filter with accurate magnitude response", Digital Signal Processing, Elsevier, May 2017 (Fractional Order Low pass Butterworth Filters (FOLBFs) using Gravitational Search Algorithm (GSA))

[2]. Ming Liu, HuaQing Hao, Peng Xiong, Feng Lin, Zeng Guang Hou, Xiuling Liu, "Constructing a Guided Filter by Exploiting the Butterworth Filter for ECG Signal Enhancement", Journal of Medical and Biological Engineering, Springer, Dec 2017

[3]. Xiaoran Ning, IvanW. Selesnick, Laurent Duval, "Chromatogram baseline estimation and denoising using sparsity (BEADS)", Chemometrics and Intelligent Laboratory Systems, Elsevier, Sep 2014

[4]. Mahmud Hasan and Mahmoud R. El-Sakka, "Improved BM3D image denoising using SSIM-optimized Wiener filter", EURASIP Journal on Image and Video Processing, Springer, May 2018

[5]. Wenjing Zhao , Yue Chi, Yatong Zhou, Cheng Zhang, "Image Denoising Algorithm Combined with SGK Dictionary Learning and Principal Component Analysis Noise Estimation", Mathematical Problems in Engineering, Hindawi, Mar 2018

[6]. İlker Bayram, "A Multichannel Audio Denoising Formulation Based on Spectral Sparsity", IEEE/ACM Transactions on Audio, Speech and Language Processing, VOL. 23, NO. 12, DECEMBER 2015

[7]. Ender M. Eksioglu, "Online dictionary learning algorithm with periodic updates and its application to image denoising", Expert Systems with Applications, Elsevier, Jun 2013

[8]. Junyuan Xie, Linli Xu, Enhong Chen, "Image Denoising and Inpainting with Deep NeuralNetworks", Proceedings of the 25th International Conference on Neural Information Processing Systems, ACM, Dec 2012

[9]. Gourav Chalotra, Tejpal Sharma, Upinderpal Singh, "Various Filtering Methods for Image Noise Reduction using Denoising Algorithm", International Conference on Innovations in Computing (ICIC 2017), pp:29-35

[10]. T. Pham Gia, "The Mean and Median Absolute Deviations", Mathematical and Computer Modeling, Elsevier, Jun 2001

[11]. L. Mredhula, M.A. Dorairangaswamy, "An effective image denoising using PPCA andclassification of CT images using artificial neuralnetworks", International Journal of Medical Engineering and Informatics, Inderscience, Vol. 9, No. 1, 2017

[12]. Ivan W Selesnick, Ankit Parekh, Ilker Bayram, "Convex 1-D Total Variation Denoising with Non-convex Regularization", IEEE Signal Processing Letters, VOL 22, NO 2, FEB 2015

[13]. Franziska Schirrmacher, Thomas Kohler, Jurgen Endres, Tobias Lindenberger, Lennart Husvogt, James G. Fujimoto, Joachim Hornegger, Arnd D, Philip Hoelter, and Andreas K. Maier, "Temporal and Volumetric Denoising via Quantile Sparse Image Prior", Medical Image Analysis, Springer, July 2018

[14]. Javier Portillay, Antonio Tristan-Vegaz, Ivan W. Selesnick, "Efficient and Robust Image Restoration using Multiple-Feature L2-relaxed Sparse Analysis Priors", IEEE Transactions on Image Processing (Volume: 24 , Issue: 12 , Dec. 2015)

[15]. Gabriela Ghimpe teanu, Thomas Batard, Marcelo Bertalmío, Stacey Levine, "A Decomposition Framework for Image Denoising Algorithms", IEEE Transactions on Image Processing, VOL. 25, NO. 1, JANUARY 2016

[16]. Md. Robiu Islam, Chen Xu, Yu Han and Rana Aamir Raza Ashfaq, "A Novel Weighted Variational Model for Image Denoising", International Journal of Pattern Recognition and Artificial Intelligence Vol. 31, No. 12, Jul 2017

[17]. Monagi H. Alkinani and Mahmoud R. El-Sakka, "Patch-based models and algorithms forimage denoising: a comparative reviewbetween patch-based images denoisingmethods for additive noise reduction", EURASIP Journal on Imageand Video Processing, Springer, May 2017

[18]. Yan Jin, Wenyu Jiang, Jianlong Shao, Jin Lu, "An Improved Image Denoising Model Based onNonlocal Means Filter", Mathematical Problems in Engineering, July 2018

[19]. Manuel Schimmack, Paolo Mercorelli, "A Wavelet Packet Tree Denoising Algorithm for images of Atomic-Force Microscopy", Asian Journal of Control, Vol. 20, No. 2, pp. 1–12, March 2018

[20]. Aleksandra Pizurica, "Image denoising algorithms: from wavelet shrinkage to nonlocal collaborative filtering", Wiley Encyclopedia of Electrical and Electronics Engineering., John Wiley & Sons, May 2017

[21]. Gasper Tkacik, Patrick Garrigan, Charles Ratliff, Grega Milcinski, Jennifer M. Klein, Lucia H. Seyfarth, Peter Sterling, David H. Brainard, Vijay Balasubramanian, "Natural Images from the Birthplace of the Human Eye", PLoS ONE, June 2011 | Volume 6 | Issue 6 |

[22]. Shilpa Suresh, Shyam Lal , Chen Chen, and Turgay Celik, "Multispectral Satellite Image Denoising via Adaptive Cuckoo Search-Based Wiener Filter", IEEE Transactions on Geoscience and Remote Sensing (Volume: 56 , Issue: 8 , Aug. 2018) – Wiener filter based on the Adaptive Cuckoo Search

[23]. RigDas., and Themrichon Tuithung., "A Novel Steganography Method for Image Based on Huffman Encoding", IEEE Transaction on Image Processing, June 2012

D.Purushothaman "Satellie and Natural Image Denoising Using Singular Difference Butterworth and Sliding Deep Neural Network." International Journal Of Engineering Science Invention (Ijesi), Vol. 08, No. 01, 2019, Pp 87-98