Preprocessing using Enhanced Median Filter for Defect Detection in 2D Fabric Images

S.Anitha [#1], V.Radha [#2]
Department of Computer Science
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore
[1] mail.anitha08@gmail.com
[2] radharesearch@gmail.com

Abstract—Quality control is one of the basic issues in textile industry. Analysis of texture content in digital images plays an important role in the automated visual inspection of fabric images to detect the defects. The preprocessing stage is an essential initial step in the patterned fabric defect detection system in retaining the important information. And the major problem in processing patterned fabric images is the presence of impulse noise which appears as bright dots or dust particles over the image. This paper proposes an enhanced switching median filter to identify the noisy pixel from the noise free pixel. And the noise is removed based on threshold estimation using genetic algorithm. The performance of the proposed method is evaluated using the metrics such as peak signal to noise ratio, edge preservation capacity and structure preservation capacity and time complexity.

Keyword-Defect Detection, Median Filter, Genetic Algorithm

I. INTRODUCTION

Quality inspection of textile products is an important problem for fabric manufacturers. Quality control is an indispensable component of modern manufacturing, and the textile industry is no different to any other industry in this respect. The investment in the automated fabric inspection system is economically attractive when reduction in personnel cost and associated benefits are considered. There are numerous works reported in the past two decades during which computer vision based inspection has become one of the most important application areas. The texture materials can be further divided into uniform, random and patterned textures.

The preprocessing stage is an essential initial step in the proposed patterned fabric defect detection system. This phase focuses on techniques that enhance the patterned fabric image. Denoising techniques is considered a challenging research area because the process of denoising is irreversible. Therefore, care should be taken while removing noise, as some techniques mistakenly might remove important regions of an image as noise.

During image acquisition, inhomogeneities occur due to variance in relative position of the light source, camera position and the textile position. These inhomogeneities make some part of the image appear darker and many have uneven contrast. Moreover, the presence of impulse noise also degrades and distorts the images. Impulse noise can be fixed-valued (salt and pepper) or random-valued noise. Both of these can be mistakenly identified as defect pattern and therefore has to be removed.

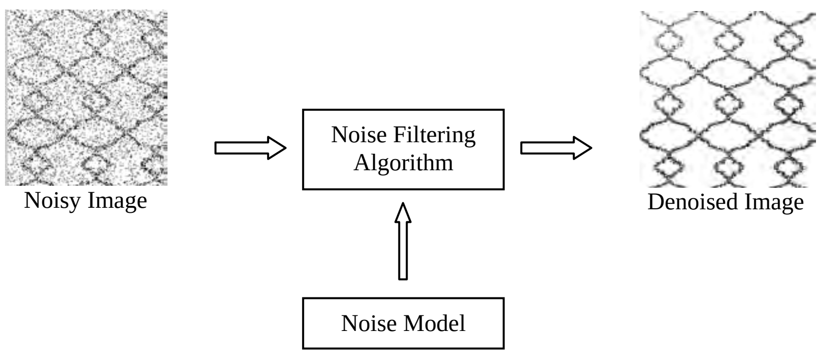


Fig 1. General Noise Removal Process

Among the various proposed methods, the median filter is one of the most commonly used non-linear filters. It has already been established that median filters are more efficient in removing salt and pepper noise and are computationally inexpensive algorithms. However, it also has the drawback of smearing detailed regions like edges of the original image.

Several methods have been proposed to solve this problem and they include adaptive filter [1][2], multistate median filter [3], weighted median filter [4] and switching median filters [5]. Vector directional filters uses directional image vectors during denoising [6].

The denoising task is considered as the problem of estimating the noise model in a fabric image using which the best method of restoration can be designed. In general, it consists of using a filtering algorithm for this purpose. The filtering algorithm has to be selected carefully with the aim of constructing a denoised image that is as close to the noise free image as shown in the above Fig 1.

II. PROPOSED PREPROCESSING METHOD

The proposed algorithm is an unified model termed as Enhanced Directional Switching Median Filter (EDSMF) and considers Salt & Pepper noise. This method is an improved version of the filter proposed by [7] (KWF method) who used a modified version of Switching Median Filter (SMF) based on the rank order arrangement to implement impulse noise removal. The KWF method, when used with patterned fabrics, sometimes produces excessive smoothening in highly textured area. To solve this problem, the KWF method is enhanced to use an improved directional adaptive criterion that can distinguish between texture and noisy regions and then use a switching filtering only to noisy regions.

The proposed methodology behind the EDSMF denoising algorithm is diagrammatically presented in Fig 2. The methodology includes three main steps.

- Impulse noise detection using enhanced directional detector from the noisy image X.
- Create binary image B that identifies noisy pixels and noise free pixels.
- Perform modified adaptive directional switching median filter to remove noise.

The proposed EDSMF method enhances the traditional Vector Median Filter (VMF) by combining it with adaptive directional detector for impulse noise removal in noisy fabric image.

A. Traditional Vector Median Filter

All denoising methods depend on a filtering parameter ‘h’. This parameter measures the degree of filtering applied to the image. For most methods, the parameter ‘h’ depends on an estimation of the noise variance σ2. The result of a denoising method Dh can be defined as a decomposition of any image ‘v’ as given in Equ(1).

$$Dhv + n(Dh, v) \tag{1}$$

where Dhv is more smooth than v and n(Dh, v) is the noise predicted by the method. The denoising method smoothens v and also to make sure that the contents lost due to noise are recovered, it uses an variant of median filter, called vector median filter and combines it with directional noise detection and adaptive switching vector median filter.

Median filter, a non-linear filtering technique, uses a window that moves over a signal and at each point, the median value of the data within the window is taken as the output. The impulse response of the median filter is zero and thus makes its use attractive to suppress impulsive noise. Median filters are robust and are well-suited for data smoothing when the noise characteristics are not known and also has the capability to preserve edges.

In the vector median approach, the samples of the vector-valued input signal are processed as vectors as opposed to component-wise scalar processing. The vector median operation inherently utilizes the correlation between the signal components giving the filters some desirable properties.

A vector median filter is defined as the vector that corresponds to the minimum sum of distances to all other vector pixels. The selection of the pixel with minimum sum of distance may be readily visualized as finding the pixel nearest to the ‘centre’ of the pixels within the neighbourhood viewed as a cluster in the gray level space.

The VMF algorithm has three main steps. The first step, after dividing an image into fixed-equal sized windows, computes the Euclidean distance from every pixel to every other pixel in its neighbourhood in the current window chosen. Then it arranges the vector pixels of this window in ascending order on the basis of the sum of distances. The ordering used for sum of distances is associated with the vector pixels also. The vector pixel with the smallest sum of distances is the vector median pixel. The VMF is represented as follows.

$$XVMF = vector\ median(window) \tag{2}$$

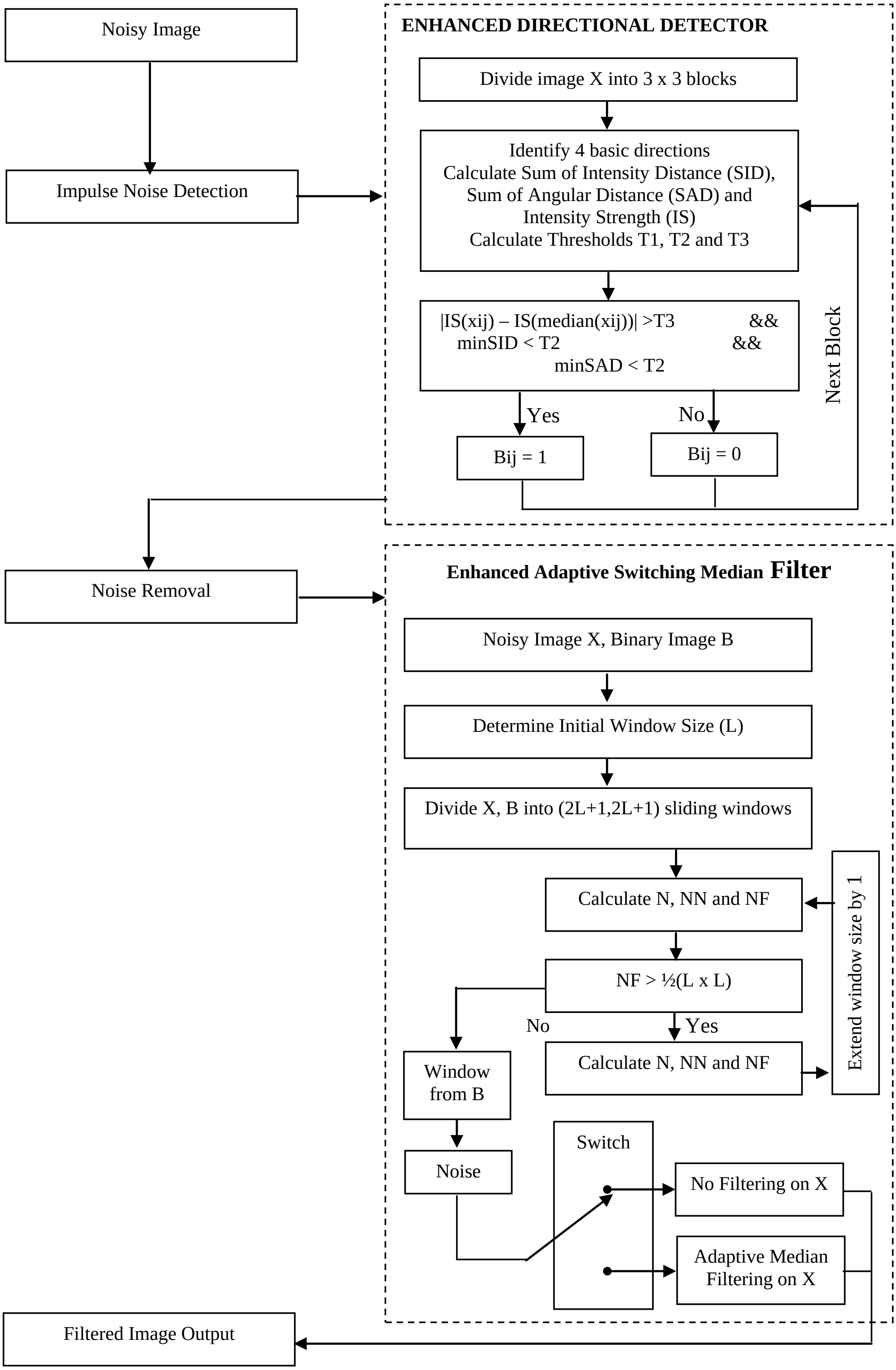


Fig 2. Enhanced Directional Switching Median Filter

If δi is the sum of the distances of the i^{th} vector pixel with all the other vectors in the kernel and is calculated using Equ (3).

$$\delta_i = \sum_{n=1}^{N} d(x_i, x_n) \tag{3}$$

where d(Xi,Xn) represents a distance measure between the i[th] and the nth neighbouring vector pixels with (1 ≤ i < N) and Xi and XN are vectors with N=9. The ordering may be illustrated using Equ (4) and this implies the same ordering to the corresponding vector pixels Equ (5). In these equations, the subscripts represent the ranks.

$$\delta(1) \le \delta(2) \le \ ... \ \le \delta(9) \tag{4}$$

$$x(1) \le x(2) \le \ ... \ \le x(9) \tag{5}$$

Since the vector pixel with the smallest sum of distances is the vector median pixel, it will correspond to rank 1 of the ordered pixels Equ (6).

$$\mathrm{XVMF} = \mathrm{X}(1) \tag{6}$$

The steps are consolidated in Fig 3. The VMF is highly effective in removing impulsive noise but also has a disadvantage. More than one pixel derives the minimum distance thereby more than one qualified pixel to replace the center pixel.

Step 1	:	Calculate Vector Median as sum of distances from every pixel to every other pixel in its neighbourhood in the filtering window.
Step 2	:	Select the pixel with minimum distance as vector median of that window.
Step 3	:	Replace noise pixel with vector median

Fig 3. Steps in Vector Median Filter

The enhanced version solves both these problems by using a procedure to differentiate edges / boundaries from other regions and applying VMF only to other regions. The problem is solved by minimizing the repeated calculations involved. This is performed by using a procedure that calculates the minimum distance in a fast manner and applying VMF only to those pixels that are affected by impulse noise. The problems of more than one pixel having the same minimum distance and distortion are solved by using a simple rule-based distance calculation. The enhanced version of VMF has the advantage of edge preservation and reduction of computation complexity.

Let X is the input noisy image. The first step of the algorithm is to identify noisy and noise free pixels in the image. Consider a sliding window of a noisy image X of size 3 x 3 with current pixel coordinates (i, j). In the proposed filter, four directions are considered, namely, TLBR (Top Left-Bottom Right), TCBC (Top Centre-Bottom Centre), TRBL (Top Right-Bottom Left), CRCL (Center Right - Centre Left) Fig 4.

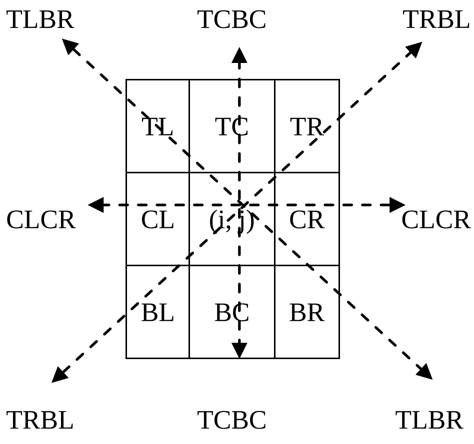


Fig 4. Directions Considered

In the next step, two measures namely, Sum of Intensity Distance (SID) and Sum of Angular Distance (SAD) are calculated for each direction. The SIDs for the four directions are calculated using Equ (7) to (10).

$$\mathrm{SID_{ij}(TLBR)} = |\mathrm{L(x_{ij}) - L(TL_{ij})} + |\mathrm{L(x_{ij}) - L(BR_{ij})}| \tag{7}$$

$$SID_{ij}(TRBL) = |L(x_{ij}) - L(TR_{ij}) + |L(x_{ij}) - L(BL_{ij})| \tag{8}$$

$$\mathrm{SID_{ij}(TCBC)} = |\mathrm{L(x_{ij}) - L(TC_{ij})} + |\mathrm{L(x_{ij}) - L(BC_{ij})}| \tag{9}$$

$$\mathrm{SID_{ij}(CLCR)} = |\mathrm{L(x_{ij}) - L(CL_{ij})} + |\mathrm{L(x_{ij}) - L(CR_{ij})}| \tag{10}$$

where L(.) is the brightness value of pixel x. The SAD of two pixels x_1 and x_2 can be calculated using Equu (11).

$$\arccos\left(\frac{x_1 x_2}{\sqrt{x_1^2}\sqrt{x_2^2}}\right) \tag{11}$$

Using the above equation, the SAD of the four directions are calculated using Equ (12) to (14).

$$SAD_{ij}(TLBR) = AD(x_{ij}, TL_{ij}) + AD(x_{ij}, BR_{ij}) \tag{12}$$

$$SAD_{ij}(TRBL) = AD(x_{ij}, TR_{ij}) + AD(x_{ij}, BL_{ij}) \tag{13}$$

$$SAD_{ij}(TCBC) = AD(x_{ij}, TC_{ij}) + AD(x_{ij}, BC_{ij}) \tag{14}$$

$$SAD_{ij}(CLCR) = AD(x_{ij}, CL_{ij}) + AD(x_{ij}, CR_{ij}) \tag{15}$$

Identify the minimum SID and SAD from the calculated values Let this be denoted min_{SID} and min_{SAD}. If min_{SID} is less than a threshold T1 and min_{SAD} is less than another threshold T2, then the pixel x_{ij} is treated as a noisy pixel, else it is treated as a normal uncorrupted pixel.

B. Threshold Estimation

The two thresholds T1 and T2 are calculated using the Genetic Algorithm (GA) proposed by [8].GAs provide a learning method inspired by evolutionary biology. GAs are the most popular class of evolutionary algorithms that use mechanisms such as reproduction, mutation, crossover), natural selection, and survival of the fittest to simulate biological evolution [9].Genetic algorithms begin the search for solutions in a population of initial hypotheses that are traditionally generated at random. Each hypothesis, called an individual or a chromosome, represents a potential solution of the problem. Individuals are encoded as bit strings whose interpretation depends on applications. Typically, individuals are represented in binary as strings of 0's and 1's. The initial population then evolves in generations. In each generation, every individual of the current population is evaluated according to the fitness function F, which is a predefined numerical measure for the problem at hand. A new population is generated by stochastically selecting the current fittest individuals. Some of the selected individuals are modified to produce new offspring individuals by mutating and recombining parts of them.

Some of these selected individuals are passed to the next generation intact. The new population is then used in the next iteration of the algorithm. Random search strategies powered by the genetic operators (mutation and crossover) are designed to move the population away from local optima where many algorithms face hindrance. In the GA based selection method, there are several operations that need to be determined. They are chromosome encoding and fitness function.

In chromosome encoding, a binary encoding scheme is used where a binary bit string represents an individual. Each individual represents a feature subset. The individuals are encoded by L-bit binary vectors. The bit with value 1 in a vector represents the corresponding feature being selected, while the bit with value 0 means the opposite. The length of each chromosome is determined by the number of features N.

Thus, in the encoding scheme used, the chromosome is a bit string whose length is determined by the number of parameters in the image. Each parameter is associated with one bit in the string. If the i^{th} bit is 1, then the i^{th} parameter is selected, otherwise, that component is ignored. Each chromosome thus represents a different parameter subset. In order to solve the problem of threshold value selection, the chromosomes are represented as an element vector with two values {T1, T2}. The genetic algorithm is designed to optimize two objectives:

(i) Maximize classification accuracy of the feature subset
(ii) Minimize the number of features selected.

For this purpose, the Mean square error (MSE) between the original image and the restored image is used as a fitness function. MSE is calculated using Equ (16).

$$MSE(Z,Y) = \frac{1}{3}\sum_{i=1}^{N}\sum_{k=1}^{3}\left(Z_i^k - Y_i^k\right)^2 \tag{16}$$

where Z_i^k *and* Y_i^k are the k^{th} component value of i^{th} pixel in the original image and the restored image respectively. Here, N is the total number of pixels in an image. During the process of genetic algorithm operation the value of fitness function gradually decreases while the number of generations grows. The genetic algorithm based threshold estimation procedure is given in Fig 5.

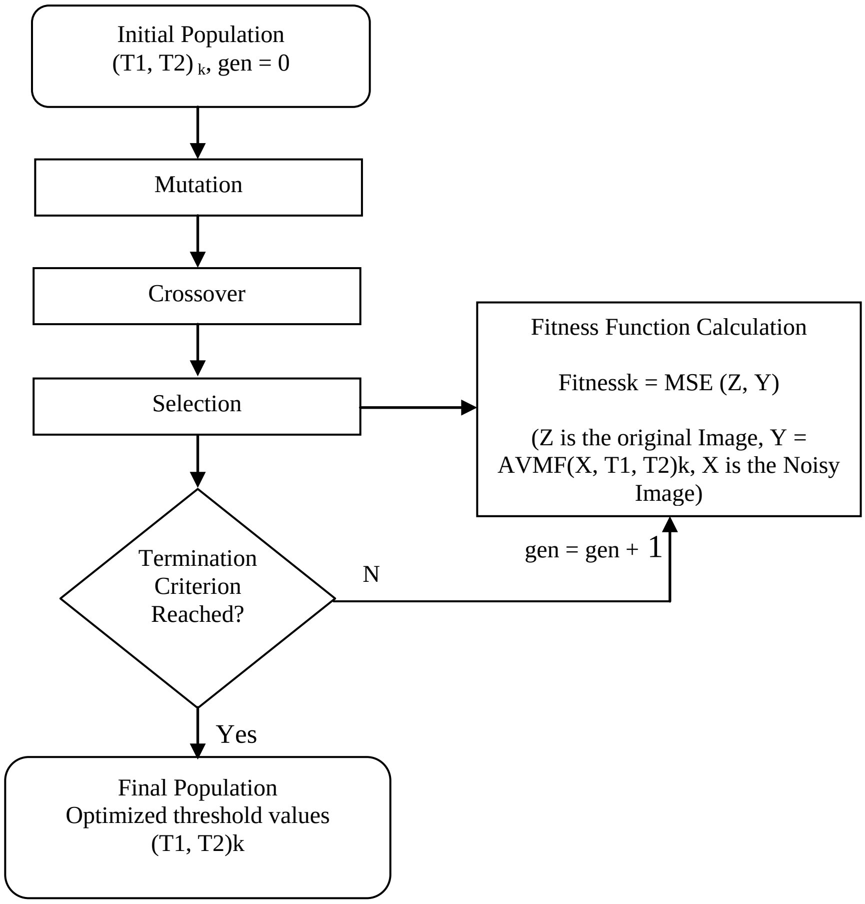


Fig 5. Optimized Threshold Value Estimation Procedure

The analysis of the dependencies shows that during the process of genetic algorithm operation and hence during the growth of the number of generations the value of fitness function gradually decreases, reaching approximately 20% relative to its initial value. And it is possible to achieve the result only after 15-20 generations. Performed experiments show that optimized parameters of the filter, averaged on several runs of genetic algorithm, differ insignificantly from each other for different test images.

C. Formation of Binary Image

The genetic-based threshold calculation algorithm has the drawback that when the impulse noise is uniformly distributed, both T1 and T2 are very high. It is a well-known fact that high threshold values cannot detect noisy pixels correctly and often misclassify them as noise free pixels. Hence, a small threshold value is desired. But with a low threshold value the number of noise free pixels preserved reduces. For this reason, there exists a trade-off in the selection of threshold values and optimal threshold should detect maximum noisy pixel while preserving all noise free pixels. For this purpose, in the proposed algorithm, a third threshold value, T3, is used. T3 is calculated by sorting the pixels in the windows in ascending order of intensity excluding the central pixel x_{ij}. The index position indicates the Intensity Strength (IS) of the pixel using which the second step of noise detection is performed Equ (17).

$$B_{ij} = \begin{cases} 1 & |IS(x_{ij}) - IS(median(x_{ij}))| > T3 \\ & \&\&\ \min_{SID} < T1\ \&\&\ \min_{SAD} < T2 \\ 0 & otherwise \end{cases} \quad (17)$$

Here, T3 is calculated using Equ (18).

$$T3 = 0.5(4L2 + 4L + 3) \cdot p(1 - p) \quad (18)$$

where (4L2 + 4L + w) is the number of the sorted sample data including the current pixel $x_{i,j}$ which multiples 3 times and p is the noise density. This formula works since theoretically, the noise density in the entire image is identical to the noise density in the sliding window. For example, if p=40% and w=1, then T3 is set as 3 in the

3×3 sliding window. Thus, the result of noise detection is a binary image with a value zero indicating noise free pixels and a value 1 indicating a noisy pixel and is denoted as image B.

D. Enhanced Directional Switching Median Filter

The next stage of the algorithm considers the binary image from the previous step and performs median filtering only for the noisy pixels. This process is called as the switching median filter. In this paper, an directional switching median filter is used.

The VMF algorithm proposed has the drawback of excessive smoothening during noise removal process. In order to solve this problem and to improve the visual quality of the image the switching median filter was modified to use an adaptive switching concept that switches to noise removal only when noisy regions are detected. The steps involved are given below.

Step 1 : Determine initial window size, L, using Equ (19) where ND is the noise density.

$$L = \begin{cases} ND \le 40 & 1 \\ 41 > ND < 60 & 2 \\ ND \ge 61 & 3 \end{cases} \tag{19}$$

Step 2 : This step starts by dividing both the noisy image X and its corresponding binary image B into (2L+1 x 2L+1) sliding windows

Step 3 : Calculate the total number of pixels (NP) number of noisy pixels (NNP) and number of noise free pixels (NFP) in the current filtering window of X using B.

Step 4 : Repeat Step 4a until NFP > ½(L x L)

Step 4a : Extend window size by 1 on all the four sides.

Step 5 : Replace noisy pixel with the median of noiseless pixels.

III.EXPERIMENTAL RESULTS OF PREPROCESSING

To analyse the performance of the preprocessing algorithm and its efficiency to enhance an input image, the test images in Fig 6 were used.

E. Performance metrics

To evaluate the proposed methods, four performance metrics were used. They analyse the algorithm's effectiveness in terms of image quality (Peak Signal to Noise Ratio), edge preservation capacity (Figure of Merit), structure preservation capacity (Mean Structural Similarity Index) and time complexity (Speed of enhancement). The method to calculate these metrics are presented below.

1) *Peak Signal to Noise Ratio (PSNR):* The PSNR is most commonly used as a measure of quality of reconstruction images. The higher the PSNR is the better the quality of the compressed or reconstructed image. To compute the PSNR, the block first calculates the mean-squared error using the following equation:

$$\mathrm{MSE} = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \tag{20}$$

In the above equation, *M* and *N* are the number of rows and columns in the input images, respectively. Then the block computes the PSNR for gray scale images using the following equation:

$$\mathrm{PSNR} = 10 \log 10 \left[\frac{R^2}{MSE} \right] \tag{21}$$

2) Edge Preservation Capacity:

The edge preservation capacity of the proposed algorithm is estimated using Pratt's Figure of Merit (FoM) [10] and is estimated as follows

$$FoM = \frac{1}{\max\{\hat{N}, N_{ideal}\}} \sum_{i=1}^{\hat{N}} \frac{1}{1 + d_i^2 \alpha} \tag{22}$$

where $\hat{N}$ and N_{ideal} are the number of detected and ideal edge pixels, respectively, d_i is the Euclidean distance between the i^{th} detected edge pixel and the nearest ideal edge pixel, and α is a constant typically set to 1/9. FoM ranges between 0 and 1, with unity for ideal edge detection.

3) Structure Preservation Capacity:

The structure preservation capacity of the preprocessing algorithm proposed is analyzed using Mean Structural Similarity Index (MSSI). The MSSI is a quality measure that is used to evaluate the overall image quality between the original (X) and the enhanced image (Y). The following equation is used to calculate this measure.

$$MSSI(x,y)=\frac{1}{M}\sum_{j=1}^{M}\frac{(2\mu_x\mu_y+c_1)(2\,\mathrm{cov}_{xy}+c_2)}{(\mu_x{}^2+\mu_y{}^2+c_1)(\sigma_x{}^2+\sigma_y{}^2+c_2)} \tag{23}$$

where M is the number of local windows, μ_x is the average of x, μ_y is the average of y, σx^2 is the variance of x, σy^2 is the variance of y, and cov_{xy} the covariance of x and y. c1 and c2 are two small values included to stabilize the division with weak denominator[11]. In this research, 8 x 8 window size is used during experimentation.

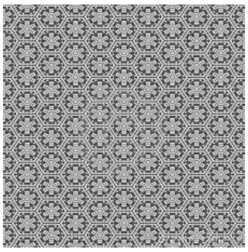
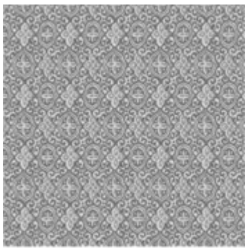
	
(a) TPF1	(b) TPF2
	
(c) TPF3	(d) TPF4

Fig 6. Test Images

4) Time Complexity

Denoising time is the execution time taken by the filters to perform the operation of impulse noise removal on the noisy image and obtain the reconstructed image. The time is measured in seconds.

For evaluation, impulse noise was introduced artificially and the corrupted images were obtained by varying noise density from 0.1 - 0.9% in steps of 0.1. Table I shows the performance of the existing and proposed algorithm with respect to image quality measured in terms of Peak Signal to Noise Ratio. Tables II and III present the FoM and MSSI values obtained for the selected test images. The time taken by the KWF and EDSMF methods to perform enhancement operations is given in Table IV.

TABLE I
Peak Signal to Noise Ratio (dB)

Noise Density	TPF1		TPF2		TPF3		TPF4	
	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF
0.1	39.17	41.85	38.13	40.29	38.01	39.21	38.28	40.96
0.2	37.95	40.37	37.00	38.97	36.79	38.46	37.06	39.48
0.3	36.87	38.18	35.71	37.29	35.47	37.02	35.98	37.48
0.4	34.87	36.65	33.71	35.26	33.14	35.11	33.68	35.75
0.5	33.56	35.71	32.40	34.12	32.11	33.55	32.67	34.82
0.6	31.37	33.89	30.21	33.00	29.92	32.73	30.48	32.41
0.7	30.12	31.77	28.99	31.14	28.54	30.97	29.23	31.33
0.8	27.46	30.55	27.09	29.66	26.97	29.08	27.22	29.94
0.9	26.54	28.77	25.38	28.00	25.27	27.89	25.65	28.26

TABLE II
Edge Preservation Capacity (FoM)

Noise Density	TPF1		TPF2		TPF3		TPF4	
	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF
0.1	0.8110	0.8153	0.8034	0.8040	0.7982	0.8018	0.8039	0.8127
0.2	0.7832	0.7925	0.7740	0.7763	0.7779	0.7867	0.7816	0.7828
0.3	0.7398	0.7468	0.7301	0.7341	0.7240	0.7301	0.7356	0.7403
0.4	0.7203	0.7320	0.7088	0.7197	0.6968	0.7025	0.7166	0.7238
0.5	0.7027	0.7325	0.6978	0.6933	0.6874	0.6944	0.7018	0.7084
0.6	0.6433	0.7007	0.6272	0.6342	0.6197	0.6224	0.6359	0.6465
0.7	0.6317	0.6522	0.6235	0.6329	0.6127	0.6199	0.6243	0.6382
0.8	0.6202	0.6619	0.6144	0.6195	0.6079	0.6130	0.6183	0.6303
0.9	0.6148	0.6380	0.6046	0.6133	0.6065	0.5976	0.6123	0.6289

From the results, it is clear that proposed EDSMF method has improved the performance of the impulse noise filtering process in terms of PSNR, FoM, MSSI and execution speed. It can further be seen that increase in impulse noise, as expected, decreases the quality of denoised image. Even with the highest noise density (0.9%), the PSNR of the proposed system was on average 40.58dB, which was 38.4dB for the KWF method. With the lowest noise density, on average the PSNR obtained while using the proposed method was 28.23dB as opposed to 25.71db obtained by the existing method. Overall, on average, the proposed method showed 6.02% quality gain when compared with base KWF method.

TABLE III
Structure Preservation Capacity (MSSI)

Noise Density	TPF1		TPF2		TPF3		TPF4	
	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF
0.1	0.8310	0.8508	0.8318	0.8353	0.8320	0.8361	0.8303	0.8385
0.2	0.8037	0.8693	0.8371	0.8393	0.8344	0.8354	0.8305	0.8364
0.3	0.7921	0.8125	0.8359	0.8361	0.8314	0.8319	0.8362	0.8377
0.4	0.7787	0.7941	0.8340	0.8349	0.8323	0.8326	0.8318	0.8327
0.5	0.7757	0.7867	0.8322	0.8381	0.8343	0.8388	0.8356	0.8391
0.6	0.7236	0.7557	0.8338	0.8375	0.8317	0.8363	0.8308	0.8315
0.7	0.7006	0.7228	0.8313	0.8343	0.8320	0.8385	0.8369	0.8317
0.8	0.6856	0.7051	0.8331	0.8350	0.8321	0.8323	0.8339	0.8368
0.9	0.6409	0.6883	0.8310	0.8351	0.8370	0.8305	0.8323	0.8374

TABLE IV
Speed of Processing (Seconds)

Noise Density	TPF1		TPF2		TPF3		TPF4	
	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF	KWF	EDSMF
0.1	1.01	0.99	1.18	1.14	1.23	1.10	1.13	1.09
0.2	1.01	0.99	1.18	1.14	1.23	1.10	1.13	1.09
0.3	1.01	0.99	1.20	1.14	1.24	1.10	1.14	1.09
0.4	1.09	1.05	1.21	1.14	1.24	1.11	1.14	1.09
0.5	1.09	1.05	1.21	1.14	1.24	1.11	1.14	1.11
0.6	1.12	1.05	1.21	1.16	1.24	1.12	1.15	1.11
0.7	1.12	1.06	1.22	1.16	1.26	1.12	1.15	1.11
0.8	1.13	1.06	1.22	1.16	1.26	1.12	1.15	1.13
0.9	1.14	1.06	1.22	1.16	1.26	1.12	1.16	1.13

Pratt's FoM is used to evaluate the edge preserving performance of the KWF and proposed methods with different noise density values. Again it could be seen that the EDSMF method has improved edge preservation which is evident by the nearing to unity values achieved. On average, the proposed method, showed an average efficiency gain of 1.24% when compared to the existing method with respect to FoM.

The MSSI values obtained reveal that the structure preserving capacity of the proposed method has improved when compared with the existing KWF method. This is ascertained by the near to unity MSSI values obtained by the EDSMF method.

From the tabulated results pertaining to speed, it could be seen that variation in noise density does not affect time taken to denoise. The speed ranged between the 1.01-1.26 seconds for KWF method while it was between 0.99 and 1.16 seconds for the proposed EDSMF method.

Thus, the runtime analysis reveals that the EDSMF method produces fast denoising results. On average, the KWF method took less than 1.17seconds while it was 1.10 seconds for the proposed method, which showed a speed efficiency gain of 5.87%.

IV. CONCLUSION

Preprocessing in fabric image involves identification and removal of noise. It is an important step that plays an important role in an automated defect detection system. It includes the variation of directions on each sub window and the optimal threshold value is calculated using genetic algorithm. The proposed method is evaluated based on the various performance metrics such as PSNR, FoM, MSSI and Time. Based on the experimental results it is evident that the proposed method is well suited for noise removal than the existing KWF method. Thus the proposed method performs better in noise removal of fabric images.

REFERENCES

[1] Manikandan, S., Uma Maheswari, O. and Ebenezer, D. Adaptive length Recursive weighted median filter with improved performance in impulsive noisy environment, WSEAS transaction on Electronics, 2004, Vol..1, Issue 3, Pp. 443-448.

[2] Kalavathy, S. and Suresh, R.M., A Switching Weighted Adaptive Median Filter for Impulse Noise Removal, International Journal of Computer Applications, 2011, Vol. 28, No.9, Pp.8-13.

[3] Chen, T. and Wu, H.R. , Space variant median filters for the restoration of impulse noise corrupted images, IEEE Transactions on Circuits Syst. II, Analog Digit. Signal Process.,2001, Vol. 48, No. 8, Pp. 784–789.

[4] Lin, H.C., Wang, L.L. and Yang, S.,Extracting periodicity of a regular texture based on autocorrelation functions. Pattern Recognition Letters,1997, Vol. 18, Pp. 433-443.

[5] Ping, W., Junli, L., Dongming, L. and Gang, C, A Fast and reliable switching median filter for highly corrupted images by impulse noise, IEEE International Symposium on Circuits and Systems, 2007, Pp. 3427-3430.

[6] Lukac, R., Adaptive color image filtering based on centerweighted vector directional filters, Multidimensional Systems and Signal Processing, 2004, Vol. 15, Pp. 169-196.

[7] Kang, C.C. and Wang, W.J.,Modified switching median filter with one more noise detector for impulse noise removal, International Journal of Electronic Communications,2009, Vol. 63, Pp. 998-1004.

[8] Goldberg, D. , Genetic Algorithms in Search, Optimization, and Machine Learning,1989, Addison-Wesley.

[9] Holland, J, Adaptation in Nature and Artificial Systems,1992, MIT Press.

[10] Yongjian Yu and Scott T. Acton, Speckle Reducing Anisotropic Diffusion, IEEE Transactions on Image Processing 2002, Vol. 11, no. 11.

[11] Kang, C.C. and Wang, W.J., Modified switching median filter with one more noise detector for impulse noise removal, International Journal of Electronic Communications, 2009,Vol. 63, Pp. 998-1004.