Decision Based Detail Preserving Algorithm for the Removal of Equal and Unequal Probability Salt and Pepper Noise in Images and Videos

Vasanth Kishorebabu[1], Kumar Karuppaiyan[4], Nagarajan Govindan[3], Ravi Natarajan[2], Sundarsingh Jebaseelan[3], and Godwin Immanuel[3]
[1]Department of Electronics and Communication Engineering, Vidya Jyothi Institute of Technology, India
[2]Department of Electrical and Electronics Engineering, Vidya Jyothi Institute of Technology, India
[3]Department of Electrical and Electronics Engineering, Sathyabama University, India
[4]Department of Electronics and Instrumentation Engineering, Sathyabama University, India

Abstract: *A novel vicinity based algorithm for the elimination of equal and unequal probability salt and pepper noise with a fixed 3x3 kernel is proposed. The proposed method uses a tree based switching mechanism for the replacement of corrupted pixel. The processed pixel is checked for 0 or 255; if found true then the pixel is considered as noisy else termed non noisy and left unaltered. If the pixel is noisy then it checks for the 4 neighbors of the processed pixel. If all the 4 neighbors are noisy then mean of the 4 neighbors are replaced. If any of the 4 neighbors are not noisy then the corrupted pixel is replaced by unsymmetrical trimmed mean. Under high noisy conditions if all the elements of the current processing window is noisy then global mean replaces the corrupted pixel. The proposed algorithm exhibits better performance both quantitatively and qualitatively over the standard and existing algorithms at very high noise densities. The performance of the existing non linear filters are outclassed by the proposed algorithm in terms of PSNR, IEF, MSE, and SSIM and also preserves fine details of an image even at high noise densities. The algorithm works well even for gray scale, color images and video.*

Keywords: *Unequal probability salt and pepper noise, unsymmetrical trimmed mean, edge preservation.*

Received July 6, 2014; accepted December 16, 2014

1. Introduction

A Images and videos are often corrupted by salt and pepper noise due to poor transmission medium or ambiguity in the acquisition unit or faulty memory location channels in hardware. The salt and pepper noise takes high and low values of the images. The pre-processing operation in image processing is noise filtering. The filters that are designed should remove noise without removing the information of an image. Non linear filters such as median filter were good in eliminating salt and pepper noise but do not remove noise at high noise densities [3]. An Adaptive median filter employed varying window size eliminated the above problem but induced blurring in images [7]. Over the years special filters like Centre Weighted median Filter (CWF) [8], recursive filters, Threshold Decomposition Filters (TDF) [16] were proposed. The above algorithm operated all the pixels in the image in spite of pixel not being noisy. Hence to elude the above flaw threshold based switched filters such as Pixel wise Mean Absolute Difference (MAD) filter, Signal Dependent Rank Ordered Mean (SD-ROM) filters were developed [1, 5]. At high noise conditions, the decision was not strong and hence the threshold based algorithms fails to preserve the fine detail of the image. Haraty and Ghadder [10] considered removal of salt and pepper noise is an important issue before performing recognition of Arab texts. The above problem was eliminated by Decision Based Algorithm (DBA) [12]. The algorithm restored the image at high noise densities by replacing the neighborhood pre-processed pixel of the current processing window. The DBA algorithm induces streaks due to the repeated replacement of pre-processed pixels. Cascaded algorithms [4] were proposed for the salt and pepper noise elimination. A Modified Decision Based unsymmetrical trimmed Median was proposed (MDBMF) [2]. This algorithm replaced with unsymmetrical median instead of conventional median. At higher noise densities the algorithm resulted in fading. The MDBMF algorithm was refined with Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) [6]. At higher noise densities the algorithm also exhibits fading. The Modified Decision Based Unsymmetrical Trimmed Midpoint Algorithm (MDBUTMPF) algorithm [14] replaced the corrupted pixel with Unsymmetrical trimmed midpoint. Syamala *et al.*, [13] introduced Adaptive B Spline Interpolation Algorithm (ABSIF) for the removal of high density salt and pepper noise. Veerakumar *et al*., [16] introduced Recursive B Spline Algorithm (RBSA) statistic for the removal of salt and pepper noise. The Decision Based Neighborhood Referred Unsymmetrical Trimmed Variants (DBNRUTVF) [15] replaced the noisy pixel with the mean of 4 neighbors or unsymmetrical trimmed median or midpoint based on the content of the current processing window. The algorithm presented here addressed in literatures works for equal probability salt and pepper noise. The Paper addresses the performance of the proposed algorithm for both equal and unequal noise probabilities. Section 2 deals with various noise models. Section 3 gives the proposed algorithm section 4 gives insight of the proposed algorithm section 5 shows the simulation results of proposed filter with existing filters. Section 6 gives the conclusion.

2. Noise Models

The image degradation is noise model is discussed as follows:

- *Noise Model* 1: Salt and pepper noise with equal noise probability: If [0 255] denote the dynamic range of y', i.e., 0 <= *Mij* <= 255 for all *(i,j)*, then they are denoted by Salt-and-pepper noise: the gray level of y at pixel location *(i j)* is illustrated in the Equation 1.

$$Yij = \begin{cases} 0 & \text{with probability } p; \\ Mij & \text{with probability } 1 - p - q; \\ 255 & \text{with probability } q; \end{cases} \quad (1)$$

Where $s=p+q$ denotes the salt-and-pepper noise level [3].

- *Noise Model 2*: Salt and pepper noise with unequal noise probability White pixels more than black pixels: For the Model 2, it is similar to equal probability noise model, except that each pixel might be corrupted by more number of "salt" (255) noise than "pepper" (0) noise with unequal probabilities. Let $P1$ and $P2$ be the probability of occurrence of salt (255) and pepper (0) respectively.

$$Yij = \begin{cases} P1 & \text{for } X=0; \\ 1\text{-}P & \text{for } X= Mij; \\ P2 & \text{for } X=255; \end{cases} \quad (2)$$

Where is the noise density $P=P1+P2$ and $P1 \neq P2$ where $P1 > P2$ [11].

- *Noise Model 3*: Salt and pepper noise with unequal noise probability black pixels more than white pixels: For the Model 3, it is similar to Model 2, might be corrupted by more number of "Pepper" (0) noise than "salt" (255) noise with unequal probabilities. Let $P1$ and P2 be the probability of occurrence of salt (255) and pepper (0) respectively.

$$Yij = \begin{cases} P1 & \text{for } X=0; \\ 1\text{-}P & \text{for } X= Mij; \\ P2 & \text{for } X=255; \end{cases} \quad (3)$$

Where is the noise density $P=P1+P2$ and $P1 \neq P2$ where $P2 > P1$ [11].

3. The Proposed Algorithm (DBDPA)

The proposed algorithm for the removal of salt and pepper noise in images and videos is implemented as follows. If the image is a gray scale image, then the algorithm is applied directly on images. In case of color images, the image is split into corresponding red, blue and green planes. The algorithm is applied on individual planes and later concatenated. In case of videos, the videos are converted into frames and then the algorithm is applied. The proposed algorithm in this paper is addressed as Decision Based Detail Preserving Algorithm (DBDPA).

- *Step 1.* Find the mean of non noisy pixels in an image by checking each pixel of the image with 0 or 255 (Which are termed as noisy candidate from the noise model given in section 2).This mean is referred to Global trimmed Mean.
- *Step 2.* Choose 2-D window of size 3x3. The processed pixel in current window is assumed as Pxy.
- *Step 3.* Convert sorted 2D array into array. Arrange the 1D data in increasing order which is given by S.
- *Step 4.* Check the processed pixel Pxy for 0 or 255
- *Step 5.* If the processed pixel holds 0 or 255 it is considered to be a noisy pixel.
- *Step 6.* Now check the 4 neighbors of the processed pixel for 0 or 255. If all the neighbors are found to be noisy (i.e., holding 0 or 255), replace corrupted pixel with mean of the 4 neighbors.
- *Step 7.* If any of the 4 neighbors is not noisy the corrupted pixel is replaced with Unsymmetrical trimmed mean. For higher noise densities there may be a chance that all the pixels of the current processing window might be noisy. Hence it is not possible to calculate the unsymmetrical trimmed mean. So under above stated conditions noisy pixels are replaced with global mean instead of Unsymmetrical trimmed mean.
- *Step 8.* If the Processed pixel does not hold 0 or 255 it is considered as non noisy pixel and hence left Unaltered.
- *Step 9.* Move the window to the next pixel. The above steps from 2 to 8 and is repeated for the entire image.

4. Insight of the Proposed Algorithm

The processed pixel is checked for low (0) or high (255) values of the gray level values. This process is done on entire pixels in the image. The large matrix refers to image and values enclosed inside a rectangle is considered to be the current processing window. The element encircled refers to processed pixel. The above discussed methodology is illustrated as below:

- *Case 1*: Pixel is noisy, some of the four neighbors are noisy, and the processed pixel is 0 which is considered to be noisy. Now check the 4 neighbors of the processed pixel which is given as 0, 123, 164, and 255. Some of the four neighbors are also noisy. Arrange the data inside the current processing window in increasing order.

Unsorted Array: 155 255 255 0 255 123 255 164 255

Sorted Array: 0 123 155 164 255 255 255 255 255

The number of noisy candidate is less than three inside the current processing window. The processed pixel is termed as noisy and the noisy pixel is replaced by unsymmetrical trimmed mean, which is evaluated as follows find the mean of uncorrupted pixel (which is trimmed mean (123, 155, 164) resulting in 147 which replaces the corrupted pixel 0 as illustrated in Figure 1.

$$\begin{pmatrix} 155 & 255 & 255 & 0 & 255 \\ 0 & 255 & 123 & 255 & 255 \\ 255 & 164 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 155 & 255 & 255 & 0 & 255 \\ 0 & 147 & 123 & 255 & 255 \\ 255 & 164 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Corrupted image Segment. Restored Image Segment.

Figure 1. Illustration of case 1.

- *Case 2*: Pixel is noisy, some of the four neighbors are noisy (But all are noisy), all the elements inside the window are noisy. This case deals with the conditionthat the processed pixel is noisy (which is 255), now check for the 4 neighbors of the processed pixel (all are 255), Here all the neighbors of the processed pixel are noisy. Count the number of noisy pixels inside the current processing window (which is 9 for this case). We cannot apply unsymmetrical trimmed mean because there is no data uncorrupted to find the trimmed mean. Hence we find global mean of the image. Consider in the above corrupted image segment (in this case it is assumed to be 5x 5 but in real time it may take the form (512 x512) or (256 x 256)). Global mean is calculated by finding the mean of all the uncorrupted pixels of the given image segment (in case of original simulation this operation is carried out for the entire image) which are indicated in square box. The corrupted pixel is replaced by the global mean, which is given as (123+172+204)/3=166 (123, 172 204 are uncorrupted candidates of the image segment) as shown in Figure 2.

$$\begin{pmatrix} 255 & 255 & 255 & 0 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & \boxed{123} \\ 0 & \boxed{172} & 255 & \boxed{204} & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 255 & 255 & 255 & 0 & 255 \\ 255 & 166 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 123 \\ 0 & 172 & 255 & 204 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Corrupted image Segment. Restored Image Segment.

Figure 2. Illustration of case 2.

- *Case 3*: Pixel is noisy; all the four neighbors are noisy. In this case the processed pixel is noisy which is (0), now check for the 4 neighbors of the processed pixel (which are 0 255 255 255). It was found that all the 4 neighbors are noisy; Hence find the mean of the 4 neighbors which is illustrated as follows (255+255+255+0)/4= 191. The processed pixel is noisy, the 4 neighbors of the processed pixel is also noisy and hence replace the noisy pixel with mean of the 4 neighbors (which is 191) as shown in Figure 3.

$$\begin{pmatrix} 233 & 0 & 205 & 0 & 255 \\ 225 & 0 & 255 & 255 & 255 \\ 117 & 255 & 165 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 233 & 0 & 205 & 0 & 255 \\ 225 & 191 & 255 & 255 & 255 \\ 117 & 255 & 165 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Corrupted image Segment. Restored Image Segment.

Figure 3. Illustration of Case 3.

- *Case 4*: Processed pixel is not noisy the processed pixel is 119 which are between 0 and 255. The processed pixel is termed as non noisy and processed pixel is unaltered as shown in Figure 4. The algorithm is represented in the form of flowchart as shown in Figure 5.

$$\begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 104 & 119 & 255 & 255 & 255 \\ 0 & 103 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 104 & 119 & 255 & 255 & 255 \\ 0 & 103 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Corrupted image segment. Restored image segment.

Figure 4. Illustration of case 5.

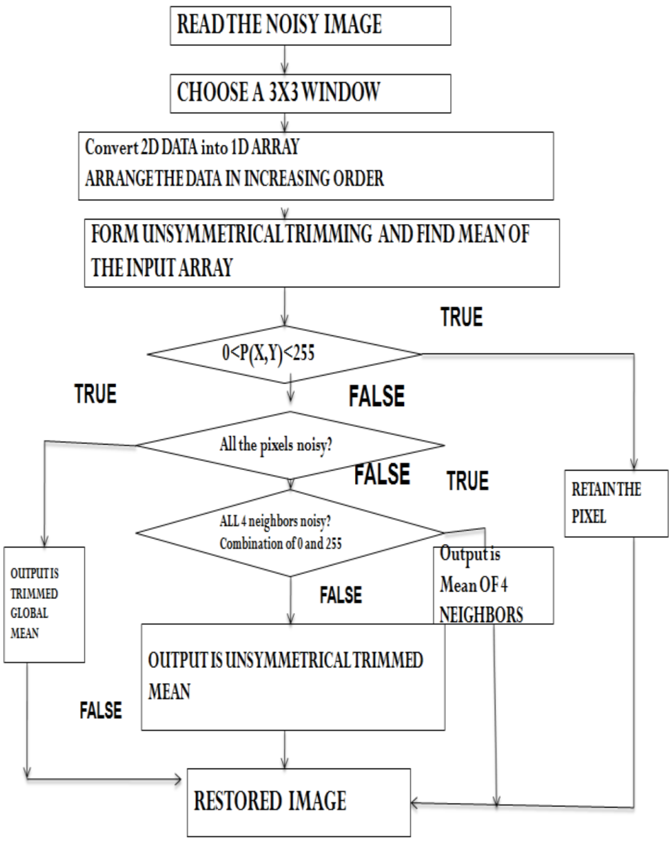


Figure 5. Illustration of the proposed algorithm as flowchart.

5. Simulation Results and Discussions

The Quantitative performance of the proposed algorithm is evaluated based on Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM). The Equation 4, 5, and 6 gives the PSNR, MSE and SSIM respectively.

$$\text{PSNR} = 20 \text{ Log } 10\left(\frac{255}{\text{MSE}}\right) \quad (4)$$

$$\text{MSE} = \sum_i \sum_j \left(\frac{(rij\text{-}xij)2}{M \times N}\right) \quad (5)$$

Where *r* refers to Original image, n gives the corrupted image *x* is denotes restored image, *M x N* is the size of Processed image.

$$\text{SSIM(x,y)} = \frac{(2\mu x\mu y + C1)(2\sigma xy + C2)}{(\mu x2 + \mu y2 + C1)(\sigma x2 + \sigma y2 + C2}) \quad (6)$$

Where μx is the average of x, μy is the average of y, σx Standard deviation of x, σy is the Standard deviation of y. $C1=(K_1L)2$, $C2=(K_2L)2$, two variables to stabilize the division with weak denominator; L the dynamic range of the pixel-values (for an 8 bit image it takes from 0 to 255), K1=0.01 and K2=0.03 by default All the simulation was done in Intel i3 2350M CPU at 2.30 GHz with 4 GB RAM capacity on MATLAB 2008b. The algorithms used in this paper are derived from the references cited inside the square bracket. The filters such as Standard Median Filter (SMF) [3], Adaptive Median Filter (AMF) [7], Alpha Trimmed Mean Filter (ATMF) [3], Threshold Decomposition Filter (TDF) [18], Progressive Switched Median Filter (PSMF) [17], DBA [11], Improved Decision Based Median Filter (IDBA) [9], Modified Decision Based Median Filter (MDBMF) [2], MDBUTMF [6] were used for the comparison. The first half of the simulation results deal with noise model 1 and the later deals with noise model 2 and 3 respectively. In this paper noise model 1 refers to equal probability salt and pepper noise and noise model 2, 3 refers to unequal probability salt and pepper noise. Tables 1, 2, 3, and 4 gives the quantitative comparison of various algorithms for PSNR, IEF, MSE and SSIM on Cameraman image corrupted by Salt and pepper noise respectively.

Table 1. Comparison of various algorithms for PSNR on cameraman image corrupted by Salt and pepper noise.

ND in %	SMF 3x3	AMF	αTMF α=4	DBA	MDBMF	MDBUTMF	DBDPF
10	26.6	30.7	21.24	30.31	35.24	35.82	34.61
20	23.6	29.4	20.03	28.70	31.96	31.6	31.86
30	20.6	27.1	18.32	26.53	29.23	29.67	29.57
40	17.3	25.2	16.21	25.18	27.12	27.71	28.21
50	14.2	21.8	14.4	23.47	25.26	25.92	26.5
60	11.7	18.0	12.82	22.09	23.51	24.25	25.49
70	9.46	14.0	11.42	20.65	21.43	22.33	23.89
80	7.55	10.3	10.2	19.07	19.10	20.27	22.15
90	6.19	7.56	9.15	15.79	16.36	17.22	19.76

Table 2.Comparison of various algorithms for IEF on cameraman image corrupted by salt and pepper noise.

ND in %	SMF 3x3	AMF	αTMF α=4	DBA	MDBMF	MDBUTMF	DBDPF
10	12.0	37.5	4.15	32.6	105.7	118.15	92.11
20	16.2	53.7	6.35	44.7	98.36	89.51	94.08
30	10.6	47.9	6.31	43	78.25	86.63	84.76
40	6.83	42	5.19	40.7	64.47	73.59	82.57
50	4.18	24.4	4.27	35.8	52.5	61.34	70.62
60	2.74	11.8	3.56	29.6	41.86	49.74	65.72
70	1.93	5.48	3	25.5	30.12	37.44	53.25
80	1.44	2.69	2.61	19.8	20.18	26.53	40.92
90	1.18	1.6	2.31	12	12.03	14.78	26.51

Table 3. Comparison of various algorithms for MSE on cameraman image corrupted by Salt and pepper noise.

ND in %	SMF 3x3	AMF	αTMF α=4	DBA	MDBMF	MDBUTMF	DBDPF
10	165	55.1	487	61	19	17	22
20	250	74.1	644	89	41	44.93	42
30	571	126	956	141	77	70.01	71
40	1758	192	1554	200	126	110.07	98
50	2424	420	2356	282	193	166.21	124
60	4412	1015	3391	410	289	244.02	183
70	7247	2573	4682	554	467	380.13	264
80	12370	6015	6202	822	799	610.03	396
90	15299	11403	7904	1516	1503	1233	686

Table 4. Comparison of various algorithms for SSIM on cameraman image corrupted by Salt and pepper noise.

ND in %	SMF 3x3	AMF	αTMF α=4	DBA	MDBMF	MDBUTMF	DBDPF
10	0.931	0.981	0.869	0.970	0.986	0.992	0.992
20	0.881	0.973	0.627	0.962	0.970	0.982	0.983
30	0.718	0.958	0.369	0.950	0.951	0.971	0.974
40	0.445	0.928	0.206	0.930	0.925	0.957	0.961
50	0.216	0.835	0.124	0.903	0.895	0.938	0.946
60	0.093	0.607	0.078	0.866	0.852	0.910	0.928
70	0.41	0.300	0.050	0.814	0.795	0.870	0.90
80	0.18	0.110	0.033	0.735	0.713	0.800	0.855
90	0.009	0.041	0.021	0.592	0.579	0.676	0.778

Figures 6, 7, and 8 illustrates the graphical performance of various algorithms over DBDPA algorithm on cameraman image for PSNR, IEF and MSE respectively.

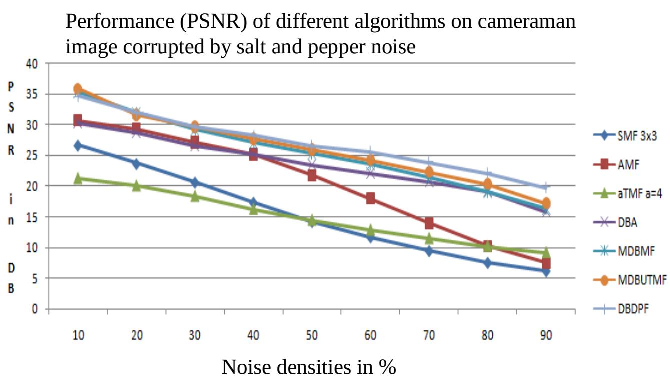


Figure 6. Performance of the PSNR for DBDPF algorithm over existing algorithms.

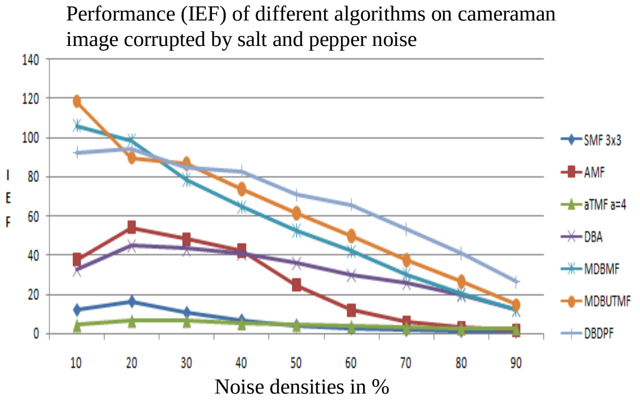


Figure 7. Performance of the IEF for DBDPF algorithm over existing algorithms.

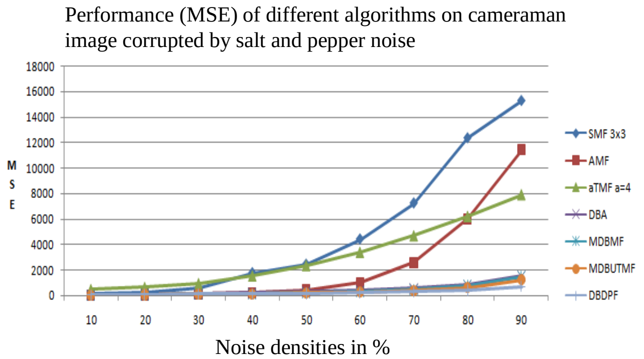


Figure 8. Performance of the MSE for DBDPF algorithm over existing algorithms.

Figure 9 gives the qualitative performance of the various algorithms on cameraman image corrupted by salt and pepper noise (Noise model 1) from 70% to 90%. Figure 10 gives the qualitative results of various algorithms corrupted by salt and pepper noise at 90% (noise model 1). Information preservation performance is justified through the construction of a synthetic image with 21 visually distinguishable gray levels. Figure 11 illustrates the performance of various algorithms on synthetic image corrupted by 90% salt and pepper noise (Noise model 1). After applying the algorithms on the corrupted image the resulted image is subjected to canny edge detection to show case the edge map of the restored image. Figure 12 gives the edge map of the restored synthetic image corrupted by 90% salt and pepper noise (noise model 1). Figure 13 gives the qualitative performance of the proposed algorithm on veg shop color image corrupted by 90% salt and pepper noise. Figure 14 shows qualitative performance of proposed algorithm corrupted by 50% of Salt and pepper noise on rhino.avi video. It was found from the tables that the DBNRUTMF fairs less at low noise densities when compared to MDBUTMF. For medium and high noise densities the DBNRUTMF shows very good by exhibiting a higher PSNR, IEF and a good SSIM values with lower MSE. This indicates that the proposed algorithm shows better results in noise removal. A high value of PSNR, IEF and SSIM indicates that the proposed algorithm shows good noise suppression characteristics, improved quality of an image after noise removal and excellent information preserving capabilities respectively. The quantitative performance of the proposed algorithm was found good at higher noise densities. The qualitative performance of the proposed algorithm was also found to exhibit good visual result in various images. The unequal probability salt and pepper noise is added to the images manually and tested on various images.

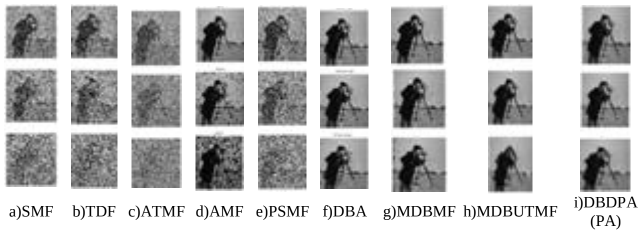


Figure 9. Qualitative Performance of the different algorithms on cameraman image corrupted by 70%, 80%, 90% (shown in row 1 to 3) salt and pepper noise. The results of different algorithm illustrated in column.

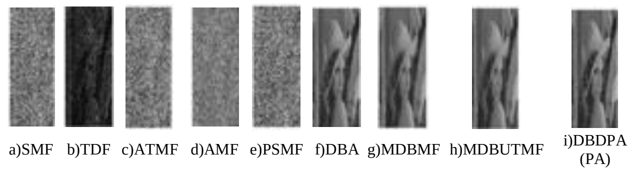


Figure 10. Qualitative Performance of different algorithms on Lena image corrupted by 90% salt and pepper noise. The results of different algorithm illustrated in column.

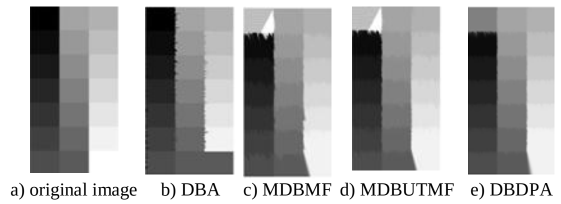


Figure 11. Qualitative Performance of different algorithms on synthetic image corrupted by 90% salt and pepper noise. The images displayed in column wise are different algorithms.

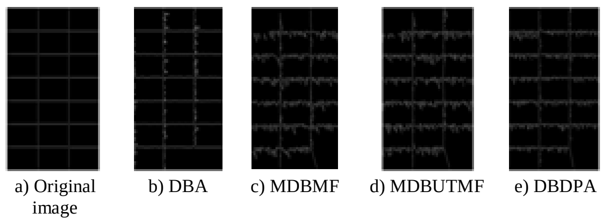


Figure 12. Edge Map of different algorithms on synthetic image corrupted by 90% salt and pepper noise. The images displayed in column wise are different algorithms

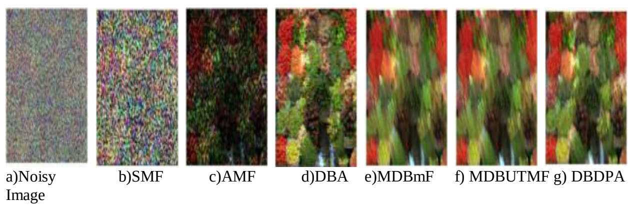
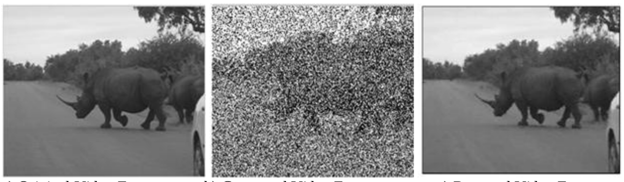


Figure 13. Qualitative Performance of different algorithms on veg shop color image corrupted by 90% salt and pepper noise. The results of different algorithm illustrated in column wise.



a) Original Video Frame. b) Corrupted Video Frame. c) Restored Video Farme.

Figure 14. Qualitative performance of the proposed algorithm corrupted by 50% of salt and pepper noise on rhino.avi video.

Tables 5, 6, and 7 gives the comparison of various algorithms for PSNR, IEF and SSIM on Cameraman image corrupted by noise model 2.

Table 5. Comparison of various algorithms for PSNR on cameraman image corrupted by noise model 2.

ND in %	SMF 3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	26.84	30.91	19.46	21.55	28.15	31.13	31.08	40.91	39.5
20	26.35	31.25	19.38	21.17	27.89	30.75	30.72	36.23	35.7
30	25.26	30.56	19.25	20.42	27.09	29.95	29.92	34.41	33.96
40	24.48	2 9.93	19.12	20.13	26.62	29.45	29.40	33.99	33.5
50	21.02	28.36	17.96	18.08	24.35	28.23	28.32	32.97	32.41
60	19.67	27.21	17.38	17.32	23.16	27.20	27.39	31.43	31.15
70	19.29	26.96	17.23	17.01	22.68	27.04	27.22	30.82	30.63
80	18.26	26.14	16.64	16.27	21.86	26.33	26.75	30.04	30.07
90	17.54	25.73	16.28	15.88	20.91	26.09	26.64	29.78	29.83

Table 6. Comparison of various algorithms for IEF on cameraman image corrupted by noise model 2.

ND in %	SMF 3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	7.82	19.97	3.24	2.31	10.58	21.01	20.76	199.82	145.56
20	15.56	48.14	6.91	4.72	22.20	42.85	42.60	151.60	135.78
30	20.36	68.99	10.38	6.67	30.99	59.88	59.48	167.17	150.69
40	17.59	61.64	9.80	6.45	28.80	55.18	54.63	157.12	140.73
50	11.63	62.94	8.61	5.90	25.00	61.15	62.41	181.89	160.13
60	9.85	56.03	8.03	5.73	22.02	55.90	58.40	148.11	138.82
70	9.88	57.83	8.32	5.84	21.60	58.91	61.44	140.74	134.73
80	8.84	54.27	7.82	5.59	20.28	56.64	62.43	133.28	134.24
90	8.16	53.86	7.62	5.57	17.74	58.46	66.40	136.67	138.35

Table 7. Comparison of various algorithms for SSIM on cameraman image corrupted by noise model 2.

ND in%	SMF 3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	0.87	0.96	0.73	0.85	0.95	0.94	0.94	0.99	0.993
20	0.86	0.96	0.72	0.76	0.95	0.94	0.94	0.98	0.987
30	0.83	0.96	0.70	0.61	0.94	0.94	0.94	0.98	0.979
40	0.82	0.95	0.69	0.59	0.93	0.94	0.93	0.97	0.977
50	0.73	0.94	0.62	0.43	0.91	0.93	0.93	0.97	0.972
60	0.68	0.93	0.59	0.37	0.88	0.92	0.92	0.96	0.963
70	0.64	0.92	0.55	0.32	0.86	0.91	0.91	0.95	0.953
80	0.59	0.90	0.50	0.26	0.82	0.90	0.90	0.94	0.944
90	0.52	0.89	0.44	0.23	0.78	0.89	0.90	0.93	0.937

Table 8, 9, and 10 gives the comparison of various algorithms for PSNR, IEF and SSIM on Cameraman image corrupted by noise model 3.

Table 8. Comparison of various algorithms for PSNR on Cameraman image corrupted by noise model 3.

ND in %	SMF 3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	26.83	30.69	19.47	21.54	28.05	30.93	30.89	40.13	39.22
20	25.82	30.75	19.34	21.01	27.32	30.19	30.21	36.28	35.72
30	25.32	30.78	19.27	20.62	27.22	30.04	30.01	34.80	34.37
40	21.16	24.30	18.13	19.40	22.52	28.74	28.96	33.33	33.24
50	20.26	23.38	17.74	18.76	21.78	24.42	24.63	28.02	31.59
60	19.14	22.65	17.36	18.01	20.48	23.76	24.02	27.1	29.9
70	17.77	21.02	16.57	17.09	19.38	22.31	22.63	25.84	29.07
80	15.65	19.09	14.93	15.78	17.63	21.97	22.38	22.9	28.64
90	13.72	16.44	13.37	14.52	15.42	21.34	21.70	21.48	27.29

Table 9. Comparison of various algorithms for IEF on Cameraman image corrupted by noise model 3.

ND in %	SMF3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	7.83	19.06	3.25	2.31	10.36	20.14	19.95	167.36	135.7
20	12.61	39.25	6.00	4.17	17.84	34.52	34.66	140.31	123.2
30	17.45	61.41	8.93	5.91	27.07	51.84	51.45	155.09	140.5
40	8.47	17.44	6.74	5.64	11.58	48.50	50.97	139.01	136.6
50	8.39	17.22	7.04	5.94	11.91	21.91	22.95	57.63	114.22
60	7.73	17.34	7.10	5.95	10.53	22.38	23.80	49.89	92.11
70	6.58	13.92	6.42	5.62	9.54	18.72	20.15	43.24	88.85
80	4.70	10.40	4.69	4.85	7.44	20.20	22.21	50.01	93.79
90	3.44	6.43	3.43	4.13	5.09	19.89	21.60	42.43	78.33

Table 10. Comparison of various algorithms for SSIM on Cameraman image corrupted by noise model

ND in %	SMF 3x3	AMF	TDF	αTMF	PSMF	DBA	IDBA	MDBUTMF	DBDPF
10	0.87	0.95	0.73	0.85	0.95	0.94	0.94	0.99	0.993
20	0.85	0.96	0.71	0.75	0.94	0.94	0.94	0.98	0.985
30	0.83	0.96	0.70	0.64	0.94	0.94	0.94	0.98	0.982
40	0.79	0.93	0.67	0.58	0.91	0.93	0.93	0.97	0.975
50	0.74	0.93	0.63	0.49	0.88	0.92	0.92	0.962	0.967
60	0.70	0.91	0.59	0.41	0.85	0.91	0.91	0.952	0.955
70	0.60	0.88	0.51	0.34	0.79	0.90	0.90	0.939	0.945
80	0.50	0.82	0.41	0.28	0.72	0.88	0.89	0.932	0.937
90	0.40	0.73	0.34	0.23	0.63	0.86	0.87	0.903	0.921

Figures 15, 16, and 17 gives the graphical comparison of various algorithms over proposed algorithm for PSNR, IEF and SSIM on Cameraman image corrupted by noise model 2.

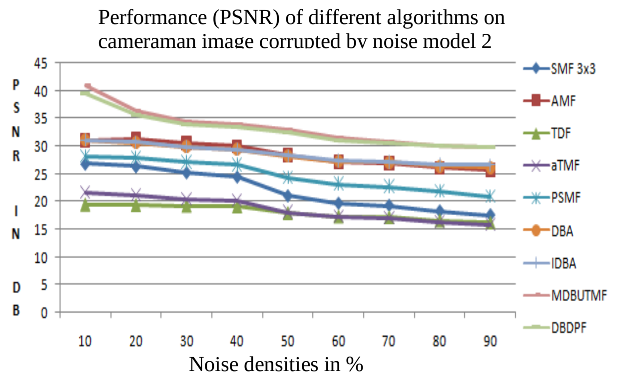


Figure 15. Performance of the PSNR for DBDPF algorithm over existing algorithms for noise model 2.

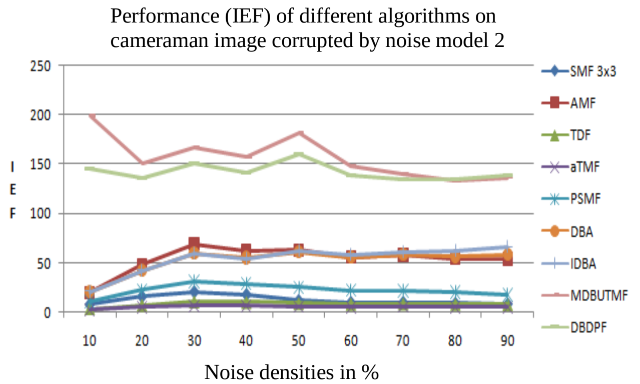


Figure 16. Performance of the IEF for DBDPF algorithm over existing algorithms for noise model 2.

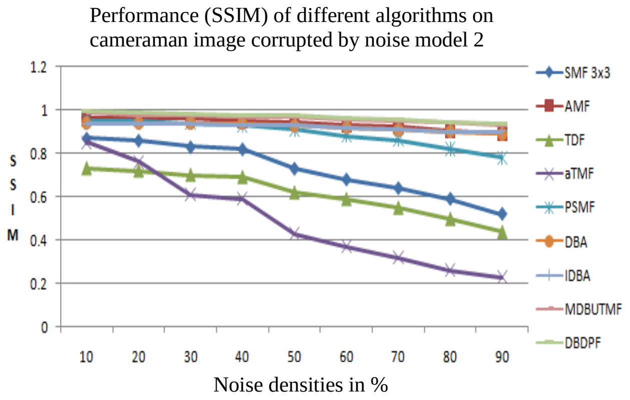


Figure 17. Performance of the SSIM for DBDPF algorithm over existing algorithms for noise model 2.

Figures 18, 19, and 20 gives the graphical comparison of various algorithms over proposed algorithm for PSNR, IEF and SSIM on Cameraman image corrupted by noise model 3. It was observed from figures and tables that the proposed algorithm is on par with the existing algorithm in terms of PSNR and IEF. The edge preservation characteristic of the proposed algorithm was found to be good when compared to other algorithms. The proposed algorithm was found to eliminate outliers in noise model 2 but the performance of the algorithm is slightly inferior when subjected to noise model 3. This is mainly due to the replacement of corrupted pixel by unsymmetrical trimmed mean value.

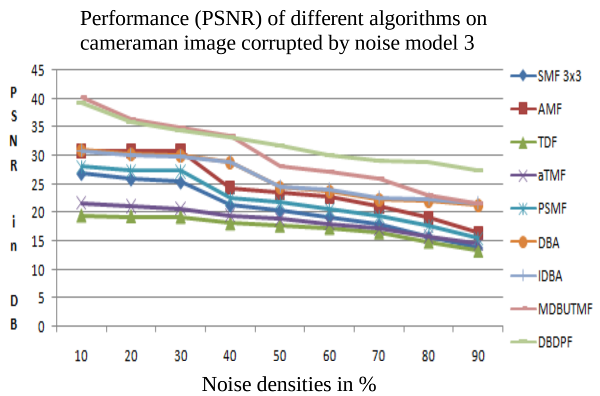


Figure 18. Performance of the PSNR for DBDPF algorithm over existing algorithms for noise model 3.

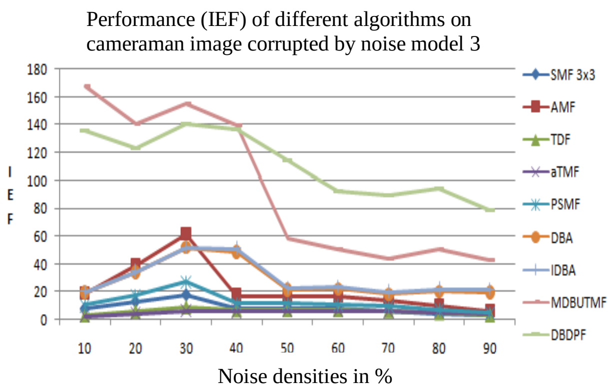


Figure 19. Performance of the IEF for DBDPF algorithm over existing algorithms for noise model 3.

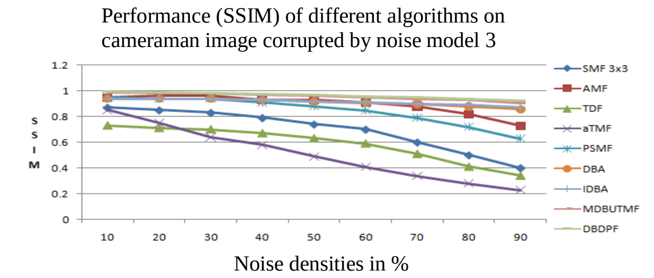


Figure 20. Performance of the SSIM for DBDPF algorithm over existing algorithms for noise model 3.

Figures 21 and 22 illustrates the quantitative performance of proposed algorithm for noise model 2 and 3 on cameraman image respectively. The main reason for the excellent results obtained by the proposed algorithm is that it replaces the corrupted pixel with mean of non noisy pixels inside the current processing window or mean of four neighbors.

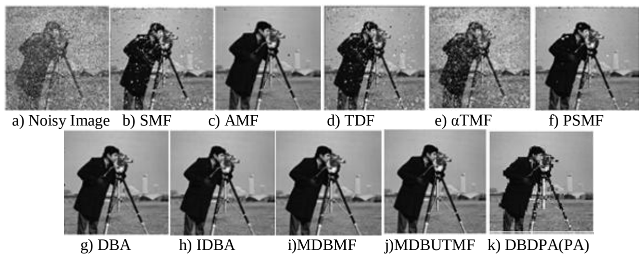


Figure 21. Quantitative Performance of different algorithms on Cameraman image corrupted by noise model 2.

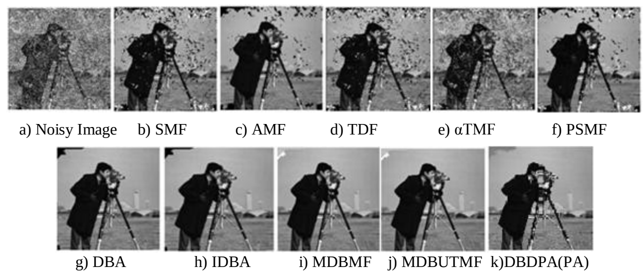


Figure 22. Quantitative Performance of different algorithms on Cameraman image corrupted by noise model 3.

6. Conclusions

This paper deals with a novel algorithm that eliminates equal and unequal probability salt and pepper noise in images and videos is proposed. The algorithm shows good results in the elimination of high density salt and pepper noise in grayscale, color images and videos. The quantitative and qualitative results of the algorithm were found good. The algorithm also found to exhibit good results in unequal probability salt and pepper noise. Hence an algorithm for the removal of three salt and pepper noise models is proposed.

References

[1] Abreu E., Lightstone M., Arakawa K., and Mitra S., "An Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images,"

IEEE Transactions on Image Processing, vol. 5, no. 6, pp. 1012-1025, 1996.

[2] Aiswarya K., Jayaraj V., and Ebenezer D., "A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noise in Images and Videos," *in Proceedings of in International Conference on Computer Modeling and Simulation*, Sanya, pp. 409-413, 2010.

[3] Astola J. and Kuosmaneen P., *Fundamentals of Non linear Digital Filtering*, CRC Press, 1997.

[4] Balasubramanian S., Kalishwaran S., Muthuraj R., Ebenezer D., and Jayaraj V., "An Efficient Non Linear Cascade Filtering Algorithm for Removal of High Density Salt and Pepper Noise in Image and Video Sequence," *in Proceedings of in International Conference on Control Automation, communication and Energy Conservation*, Perundurai, pp. 1-6, 2009.

[5] Crnojevi V., Senk V., and Trpovski Z., "Advanced Impulse Detection Based on Pixel-Wise MAD," *IEEE Signal Processing Letters*, vol. 11, no. 7, pp. 589-592, 2004.

[6] Esakkirajan S., Veerakumar T., Subramanyam A., and PremChand C., "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetrical Trimmed Median Filter," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287-290, 2011.

[7] Hwang H and Hadded. R-A., "Adaptive Median Filter: New Algorithms and Results," *IEEE Transaction on Image processing*, vol. 4, no. 4, pp. 499-502, 1995.

[8] KO S. and Lee Y., "Center Weighted Median Filters and their Application to Image Enhancement," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 9, pp. 984-993, 1991.

[9] Nair M., Revathy K., Tatavarti R., "An Improved Decision Based Algorithm for Impulse Noise Removal," *in Proceedings of in Congress on Image and Signal Processing*, Sanya, pp. 426-431, 2008.

[10] Haraty R. and Ghadder C., "Arab Text Recognition," *The International Arab journal of Information Technology*, vol. 1, no. 2, pp. 156-163, 2004.

[11] Ramkumar R., Rajesh V., and Vasanth K., "Performance of the Decision Based Algorithm for the Removal of Unequal Probability Salt and Pepper Noise in Images," *in Proceedings of International Conference in Circuit Power and Computing Technologies*, Nagercoil, pp.1360-1365, 2014.

[12] Srinivasan K. and Ebenezer D., "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 189-192, 2007.

[13] Syamala-Jayasree P., Raj P., Kumar P., Siddavatam R., and Ghrera S., "A Fast Novel Algorithm for Salt and Pepper Image Noise Cancellation Using Cardinal B-splines," *Signal Image and Video Processing*, vol. 7, no. 6, pp. 1145-1157, 2013.

[14] Vasanth K. and kumar V., "A Decision Based Unsymmetrical Trimmed Mid Point Algorithm for the Removal of High Density Salt and Pepper Noise, *Journal of Applied Theoretical and Information Technology*, vol. 42, no. 2, pp. 245-251,2012.

[15] Vasanth k. and Kumar V., "A Decision Based Neighborhood Referred Unsymmetrical Trimmed Variants Filter for the Removal of High Density Salt and Pepper Noise in Images and Videos," *Signal Image and Video Processing*, vol. 9, no. 8, pp. 1833-1841, 2015.

[16] Veerakumar T., Esakkirajan S., and Vennila I., "Recursive Cubic Spline Interpolation Filter Approach for the Removal of High Density Salt-and-Pepper Noise," *Signal Image and Video Processing*, vol. 8, no. 1, pp. 159-168, 2014.

[17] Wang Z. and Zhang D., "Progressive Switching Median Filter for Removal of Impulse N from Highly Corrupted Images," *IEEE Transactions on Circuits Systems-II*, vol. 46, no. 1, pp. 78-80, 1999.

[18] Hongke Z., "Generalized Threshold Composition," *Journal of Electronics*, vol. 14, no. 1, pp. 63-67, 1997.

Vasanth Kishore studied ECE from Madurai Kamaraj University. In 2003 he obtained his B.E degree. He later got M.E degree in Applied Electronics and later went on to complete PhD from Sathyabama University. His current research interests include non linear filtering.

Kumar Karuppaiyan is an ECE in Adiyaman Engineering College of engineering and did his Masters in Instrumentation Engg from Madras institute of technology. He has more than 17 years experience in teaching. His area of interest is Bio medical instrumentation.

Nagarajan Govindan did his E.E.E from Manonmania Sundaranar University. He later obtained his applied electronics master degree from Anna University. He also did his second Masters in CSE from Sathyabama University. He has more than 10 years of teaching experience. His areas of interest include ontology for image retrieval.

Ravi Natarajan did his B.E in E.E.E from Madras University. He later went on to complete Masters in Power Systems from Anna University. He completed completed his PhD from Sathyabama University. He has more than 10 Years of teaching experience. His areas of interest include optimization problems in Engineering.

Sundersingh Jebaseelan Did his BE in E.E.E from Bharathiyar University. He completed his Masters in Power systems from Annamalai University. He completed his PhD from Sathyabama University. He has 10 years of teaching experience. His research area includes Facts controllers.

Godwin Immanuel was born in Nagercoil,Tamilnadu. He completed his Electrical and Electronics Engineering in Manonmania sundarnar university, Tirunelveli. He later completed his Masters in Power systems in Annamalai University and completed his PhD in Sathyabama University. His research interest includes optimization techniques in engineering.