



Four stage median-average filter for healing high density salt and pepper noise corrupted images

Bharat Garg¹ · K. V. Arya²

Received: 2 October 2019 / Revised: 1 August 2020 / Accepted: 6 August 2020 /

Published online: 26 August 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

This paper introduces a novel four stage filter algorithm to ameliorate images corrupted by very high density salt-and-pepper noise. The proposed algorithm exhibits two parallel trimmed median filters (TMF) at the initial stage followed by a masking logic that selects denoised pixel based on the priority. To reduce the blurring effect, higher priority is given to TMF with small window size. In the absence of noise-free pixels, the current extreme pixel is left unchanged at the first stage. Further, the denoising of unprocessed extreme pixels is done with TMF of large size window at the second stage. The remaining noisy pixels are improved by the running average filter at the third stage. Finally, the last stage handles the noisy pixels at the boundary and rare scenario. Since the proposed filter utilized non-extreme pixels to estimate denoised pixels value, it effectively eliminates salt and pepper noise along with better edge preservation. The performance analysis of the proposed filter is carried out with various benchmark images for varying noise density. The experimental results demonstrate on an average improvement of 2.09 dB (0.018) and 1.06 dB (0.0478) of PSNR (SSIM) respectively for wide (10% - 90%) and very-high (90% - 98%) noise density ranges over state-of-the-art filters.

Keywords Image processing · Salt-and-pepper noise · Median filter · Mean filter · Non-linear filter

1 Introduction

The salt and pepper noise is often introduced during image transmission when medium is noisy and/or during image capturing or storage when faults occur in camera sensors and memory, respectively. Under this faulty/noisy-environment condition, the pixels attain either

✉ Bharat Garg
bharat.garg@thapar.edu

K. V. Arya
kvarya@iiitm.ac.in

¹ ECED, Thapar Institute of Engineering and Technology, Patiala, 147001, India

² ABV-Indian Institute of Information Technology and Management Gwalior, Gwalior, 474015, India

lowest or highest gray value [14]. The situation becomes even worse at high noise density (ND) and images become imperceivable. A filter is an effective tool used for de-noising the image which could be a hardware or software that performs some operations on the input signal. Linear and non-linear filtering [20] approaches are the most prevalent filtering techniques.

Among the various approaches [20, 24], mean and median (or the combination of the both) filters are more popular in the family of linear and non-linear filtering techniques [14]. In standard mean filter, a small window (3×3) is constructed across the noisy pixel (NP) whose mean value replaces the candidate extreme pixel. The main problem with mean filters is the inability to preserve edges because no original pixel values are restored in this process. For this reason, exploration of median filters are used which perform excellently at lower ND but result in poor image quality at high ND condition due to the presence of more corrupted pixels the median mask.

Many other non-linear filters are presented to overcome the problems of mean and median filters [2, 7, 28]. In decision-based median filter (DBAMF) [28], a specific type of sorting (row, column and diagonal wise) is done on the 3×3 window such that the resultant matrix has the median as the central pixel of the window. This central pixel (median value) is then used to restore candidate extreme pixel. If the central pixel turns out to be a noisy, then NP is restored by the nearest noise-free pixel (NFP). The prime limitation of this filter is that, the median generally turns out to be noisy at high noise density. This results in repetition of the same pixel, causing a streaking/blurring effect. Other techniques such as weighted median approach [2] mostly divides information into two or more groups and based on the weighting factor, the group is picked up for processing. An improved tolerance based selective arithmetic mean filtering technique (ITSAMFT) [7] computes the ND and if found above a given threshold it restores noisy-pixel by the mean of current window else leave the pixel unaltered by considering it as noise-free. The ITSAMFT improves the denoising performance at high noise density with small loss in edge details. However, at higher noise densities, this filter fails catastrophically due to lack of original pixels in the created groups. Many trimmed median filters [1, 11] also exist that provide satisfying solutions only in specific noise density ranges. Further, several linear and non-linear filters are proposed which are good at a certain limit of noise density, but tend to lose edge details and image blurring [11] beyond that Deivalakshmi and Palanimsamy [8] have used wavelet thresholding to improve the performance of ITSAMFT.

To achieve a good image quality at high noise density, recently, a probabilistic decision based median (PDBM) filter is reported in [5]. The PDBM filter estimates the denoised pixel by either trimmed median [11] or patch median [28] based on the noise density. A new three-value weighted median with variable size window is proposed by Lu et al. [19] where window values are segregated into multiple groups on the basis of pixels correction with the maximum, minimum or middle values. The largest group is used to determine the weight factor which is then used for further processing. If window under inspection is completely noisy then the window size is increased till at least an un-corrupted pixel is encountered. On the other hand, a linear prediction based adaptive median filter first decides the pixel to be noisy based on a threshold and then denoises candidate pixel using median of adaptive window [25]. In [10], a different applied median filter (DAMF) is presented that considers value of neighbour pixels and adaptive window to remove noise. Recently, a new method based on pixel density (BPDM) has been introduced by Erkam and Gokrem [9] where they first determine whether current pixel is NP or NFP and then decide window size where most repetitive NFP replaces the candidate noisy pixel.

The above mentioned filtering approaches fail catastrophically when the noise density is very high ($\geq 90\%$). Therefore, to recover images with high image quality at such a higher noise density, a new four stage median-average (FoMA) filtering algorithm is proposed. The proposed algorithm qualitatively and quantitatively outperformed all the existing salt-and-pepper noise removal techniques at high noise density. The major contributions of the paper are as follows:

- 1) A critical analysis on the state-of-the-art salt and pepper noise removal techniques is presented.
- 2) A novel four stage median average (FoMA) filtering algorithm and its implementation is presented. The proposed filter at the initial stage parallelly computes the median of trimmed window of sizes 3×3 and 5×5 where final output is selected by the priority based logic. The second stage iterates the trimmed median operation to denoise corrupted pixels left at the first stage. Further, the adaptive running average is exploited to denoise the corrupted pixels left at the initial two stages. Finally, the rare corrupted pixels left at the boundaries are restored using previously processed pixels.
- 3) The performance of the proposed FoMA vis-a-vis existing filters from wide range (10% - 90%) to very high (90% - 98%) noise density ranges is demonstrated. Finally, the performance of the proposed filter is demonstrated on the basis of quality and time-complexity metrics evaluated using various benchmark images.

The remaining of the paper is structured as follows: Section 2 discusses the various algorithms presented in the literature for salt and pepper noise removal. The Section 3 first presents the proposed FoMA filtering algorithm with its block diagram followed by an illustration of denoising using proposed algorithm. The Section 4 presents simulation results and analysis on various benchmark images with varying noise density. Finally, Section 5 concludes the paper.

2 Related work

Various mean and median filtering algorithms have been presented to eliminate impulse noise which perform poorly under high noise density condition due to majority of corrupted pixels in the 3×3 mask window. This means that most of the pixels remain virtually unprocessed. To improve the performance, new filters like Un-symmetric Trimmed median filter (UTMF) [2] and Un-symmetric Trimmed Midpoint Filter (UTMP) are appended to the DBAMF. UTMF provides better performance on lower noise densities, whereas UTMP works well on higher noise densities. Both algorithms consider a 3×3 window across the noisy pixels and replace candidate pixel by the mean and mid-point of the window in case of UTMF and UTMP, respectively.

The modified decision based un-symmetric trimmed median (MDBUTMF) filter is one of the first filters to produce considerably good results on wide range of noise density [11] where the candidate extreme pixel is restored by the median of all non-corrupted pixels available in the window (3×3 size) centred on the pixel under inspection. In case of uncorrupted pixels, the restoration is done using mean filter. The only problem with the above mentioned filter is that, the mean values obtained are as good as any random value thus, causing streaking in the output image. Unbiased-based weighted mean filter (UWMF) [15] is based on the spatial bias, Minkowski distance and spatial distances in the x and y directions. The UWMF shows a better performance than the MDBUTMF [11]. Further a switching adaptive median and fixed weighted mean filter (SAMFWMF) is presented in

[21] which provides optimal edge preservation even in the presence of very high noise density. A fast switching based median mean filter (FSBMMF) [31] denoises each noisy pixel ($x_{i,j}$) based on (1).

$$y(x_{i,j}) = \begin{cases} med3(x_{i,j}), & \text{if } med3(x_{i,j}) \text{ is NFP} \\ med5(x_{i,j}), & \text{elseif } med5(x_{i,j}) \text{ is NFP} \\ mean3(x_{i,j}), & \text{elseif } x_{i,j} \text{ is non-boundary pixel} \\ PPP(x_{i,j}), & \text{else PPP of row/column} \end{cases} \quad (1)$$

Here, $med3$, $med5$, $mean3$, and PPP represent median operation with 3×3 , 5×5 , mean operation with 3×3 window and previously processed pixel (PPP) operation, respectively. It is useful in preventing streaking effect by using mean of previously processed pixels at places where the median results are not satisfactory.

A recursive and adaptive filter that varies current window size on the basis of noise density and recursively denoises the candidate pixel is presented in [22]. Further, better approximation techniques such as interpolation approaches are presented (e.g.: recursive cubic spline interpolation filter (RSIF)) is presented in [30]. This filter has one drawback that it requires at least two uncorrupted pixels for prediction. However, at high noise density, two uncorrupted pixels in a 3×3 window may not be available. Other problem with this filter is the high computational time. To address the problem of high computation time, a fast adaptive high performance filter (FAHPF) approach presented in [4] which exhibits three stages where the first, second and third phase employs filtering operations, namely, overlapping medians, running averages, and a 3×3 mean, respectively. Initially, number of frames (F_w of size $W \times W$) is considered for computing overlapped median. At first stage, for each frame F_w , a binary frame m_w of size $M \times N$ is generated where noisy and noise-free pixels attains 1 and 0 values, respectively. The overlapped median (OM) output is given by (2)

$$OM = F_1 \cdot * m_1 + F_3 \cdot * (\overline{m_1} \& m_3) + F_5 \cdot * (\overline{m_1} \& \overline{m_3} \& m_5) + \dots + F_s \cdot * (\overline{m_1} \& \overline{m_3} \dots \& \overline{m_{s-2}} \& m_s) \quad (2)$$

where $\overline{m_w}$ is the complement of m_w , $(\&)$ is the pixel wise AND operation, and $(\cdot *)$ indicates pixel-wise multiplication.

Recently, an adaptive switching weighted median filter is presented in [12]. This method uses a set of rules to determine the candidate pixel is noisy or noise-free rather than considering all extreme pixels (0 and 255) as corrupted ones. This is an effort to ensure that pixels which are originally 0 or 255 in the image do not get replaced. In this algorithm, median is calculated whenever the number of information pixels in the considered window is odd otherwise few important pixels of the window under consideration are repeated and their median is calculated. If the selected window exhibits only extreme pixels then its size is changed to 5×5 and similar steps are repeated. If the pixels still remain unprocessed in 5×5 window, the nearest uncorrupted pixel is used as its substitute. The only limitation of this filter is the use of nearest uncorrupted pixel at the last step which may lead to blurring of images under high noise density conditions.

Recently, Erkan et al. [10] have developed differently applied median filter where image is first converted into binary by assigning ‘1’ to the noisy pixels and ‘0’ to rest of the pixels. These binary values are used to segregate corrupted values from

uncorrupted ones. In case the window turns out to be completely corrupted, its size is increased till an uncorrupted pixel is encountered. The maximum size of the window is set at 7×7 . If an uncorrupted value exists in the window then mean of all uncorrupted values is used to substitute the pixel in question. The use of pre-processed pixels and symmetric padding makes it better than the other filters. However, the use of large window size, results in edge information loss which ultimately leads to poor image quality.

A modified cascaded filter presented in [16], restores noisy pixel either by trimmed median when it finds some noise-free pixels in window of adaptive size or by the mean value of all pixels when all pixels are corrupted. Further, an adaptive trimmed median filter that effectively restores the salt and pepper corrupted pixel even at very high noise density [13] where noisy pixels are restored using adaptive trimmed median window or by interpolation based technique for medium or very high noise density. Since noise distribution in an image is irregular, the two dimensional scattered data interpolation (2D-SDI) can be used to estimate value of noisy pixel. In [27], a 2D-SDI based natural neighbour Galerkin method in an adaptive sized window is used to compute the value of noisy pixel. It provides improved denoising over the recent decision based filters. An iterative switching algorithm that preserves the details of image is presented in [3]. This algorithm first identifies corrupted pixels using local statistic of the textons in different directions iteratively followed by restoring noisy pixels using fuzzy rules.

Recently, machine learning has been introduced in image denoising where support vector machine is often used as a classifier to detect impulse noise [18, 26]. More recently, deep neural networks (NN) are becoming a prominent research area for image denoising. Several deep learning algorithm for the removal of noise have been presented in the literature [17, 23, 29]. These algorithms, first train an NN to be an impulse noise detector, which is then used to detect noisy pixels in an image, and finally different filtering mechanisms (e.g., mean filtering, median filtering, and edge-preserving filtering) are employed to restore the noisy pixels detected [17, 23, 29]. In addition to the removal of impulse noise, the neural networks are used to suppress additive Gaussian noise [6, 33, 34].

To overcome the limitations of poor image quality at higher noise density, the next section presents the proposed novel four stage median-average filtering algorithm.

3 The proposed filter algorithm

This section puts forth a two-fold information structure, firstly, the block diagram representation of proposed filtering approach followed by its algorithm and secondly an illustration using an example is presented.

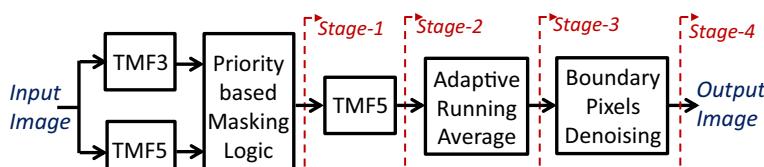


Fig. 1 Block diagram of the proposed FoMA filter

Algorithm 1 Proposed FoMA($nImg$, $OImg$).

1: **Input** $nImg$ ▷ Input Image
 2: **Output** $OImg$ ▷ Output Image
 3: **for** each noisy pixel $i P_{i,j}^n$ of $nImg$ **do** ▷ Start 1st stage
 4: Compute $w_{3 \times 3}^n$ and $w_{5 \times 5}^n$ centred at $i P_{i,j}^n$.
 5: $o1P_{i,j} \leftarrow \text{TMF3}(w_{3 \times 3}^n)$; ▷ Call procedure TMF3
 6: $o2P_{i,j} \leftarrow \text{TMF5}(w_{5 \times 5}^n)$; ▷ Call procedure TMF5
 7: **if** $o1P_{i,j} == P^{nf}$ **then**
 8: $OImg1(i, j) \leftarrow o1P_{i,j}$;
 9: **else**
 10: $OImg1(i, j) \leftarrow o2P_{i,j}$;
 11: **end if** ▷ End 1st stage
 12: **end for** ▷ Start 2nd stage
 13: **for** each noisy pixel $i P_{i,j}^n$ of $OImg1$ **do** ▷ Start 2nd stage
 14: Compute $w_{5 \times 5}^n$ centred at $i P_{i,j}^n$.
 15: $OImg2(i, j) \leftarrow \text{TMF5}(w_{5 \times 5}^n)$;
 16: **end for** ▷ End 2nd stage
 17: **for** each noisy pixel $i P_{i,j}^n$ of $OImg2$ **do** ▷ Start 3rd stage
 18: Compute $w_{3 \times 3}^n$ centred at $i P_{i,j}^n$.
 19: $OImg3(i, j) \leftarrow \text{ARA}(w_{3 \times 3}^n)$;
 20: **end for** ▷ End 3rd stage
 21: **for** each noisy pixel $i P_{i,j}^n$ of $OImg2$ **do** ▷ Start 4th stage
 22: Compute $w_{3 \times 3}^n$ centred at $i P_{i,j}^n$.
 23: $OImg4(i, j) \leftarrow \text{BPD}(w_{3 \times 3}^n)$;
 24: **end for** ▷ End 4th stage
 25: $OImg \leftarrow OImg4$;
 26: **Return** $OImg$; ▷ Return denoised output image
 27: **End** ▷ End of Algorithm
 28: **Procedure 1:** $\text{TMF3}(w_{3 \times 3}^n)$ ▷ TMF3 procedure
 29: Compute w_3^{nf} ▷ Compute noise free pixels in $w_{3 \times 3}^n$.
 30: **if** $w_3^{nf} != 0$ **then** ▷ Check w_3^{nf} have at least NFP.
 31: **Return:** median(w_3^{nf})
 32: **else**
 33: return: Central noisy pixel of $w_{3 \times 3}^n$ ▷ TMF3 procedure
 34: **end if**
 35: **Procedure 2:** $\text{TMF5}(w_{5 \times 5}^n)$ ▷ TMF5 procedure
 36: Compute w_5^{nf} from $w_{5 \times 5}^n$.
 37: **if** $w_5^{nf} != 0$ **then** ▷ Check number of noise-free pixels
 38: **Return:** median(w_5^{nf})
 39: **else**
 40: **Return:** Central noisy pixel of $w_{5 \times 5}^n$ ▷ ARA procedure
 41: **end if**
 42: **Procedure 3:** $\text{ARA}(w_{3 \times 3}^n)$ ▷ ARA procedure
 43: Compute w_3^{nf}
 44: **Return:** mean(w_3^{nf})
 45: **Procedure 4:** $\text{BPD}(w_{3 \times 3}^n)$ ▷ BPD procedure
 46: **if** $i P_{i,j}^n$ does not belongs to periphery **then**
 47: $oP_{i,j} = \text{mean}(oP_{i,j-1}, oP_{i-1,j-1}, oP_{i-1,j}, oP_{i-1,j+1})$
 48: **else if** $i P_{i,j}^n$ belongs to first/last row **then**
 49: $oP_{i,j} \leftarrow oP_{i,j-1}$;
 50: **else if** $i P_{i,j}^n$ belongs to first/last col **then**
 51: $oP_{i,j} \leftarrow oP_{i-1,j}$;
 52: **end if**
 53: **Return:** $oP_{i,j}$;

3.1 Block diagram of proposed four-stage median average filter

The block diagram representation of proposed FoMA filter is shown in Fig. 1. At the first stage, two trimmed median filters (TMF) with window sizes 3×3 and 5×5 namely TMF3 and TMF5, respectively are used to estimate denoised pixels. These TMFs compute the value of candidate noisy pixel using original (noise-free) pixels and leave this noisy pixel unchanged if current window have extreme pixels only. Further, the priority based masking logic selects denoised pixel either from TMF3 or TMF5 based on the priority. Higher priority is given to TMF3 to reduce the blurring effect. In the algorithm, w_3^{nf} and w_5^{nf} are the numbers of noise-free pixels of the noisy window $w_{3 \times 3}^n$ and $w_{5 \times 5}^n$, respectively. The output of the first stage is stored in $OImg1$ (similarly, the output of stage 2, stage 3 and stage 4 is stored in $OImg2$, $OImg3$, and $OImg$, respectively). The output of the first stage may exhibit noisy pixels if current window (5×5) exhibits extreme pixels only. Therefore, TMF5 is used again at the second stage to reduce noise at high noise density. The combined two stages (stage-1 and stage-2) provide adaptive median operation of window size up to 7×7 . Therefore, only small numbers of noisy pixels left unprocessed.

For low to high noise density ranges ($10\% - 90\%$), two stages are sufficient to provide denoised image of good quality. However, at very high noise density ($> 90\%$), certain noisy pixels the output of denoised image of stage-2 might be present. Therefore, adaptive running average (of window 3×3) is used to denoise these noisy pixels at third stage. In the ARA, previously processed pixels within 3×3 window are also considered while computing the average value. Finally, the noise at the boundary is healed at the last stage. The proposed filter gradually decreases noisy pixels with each stage and provides completely noise free image at the last stage. To reduce implementation complexity/computations, last stage or even last two stages (stage-3 and stage-4) can be eliminated with small loss in output image quality at very high noise density.

The mathematical expression for operation at Stage-1, Stage-2, Stage-3, and Stage-4 are given by (3), (4), (5), and (6), respectively.

$$OImg1 = \begin{cases} med3(nImg), & \text{if } med3(nImg) \text{ is NFP} \\ med5(nImg), & \text{Otherwise} \end{cases} \quad (3)$$

$$OImg2 = med5(OImg1) \quad (4)$$

$$OImg3 = avg5(OImg2) \quad (5)$$

$$OImg = \begin{cases} ARA(OImg3), & \text{if, pixel deosnt from boundary} \\ PPP(OImg3), & \text{else, PPP of that row/col} \end{cases} \quad (6)$$

The convention $med3$, $med5$, and ARA represent median operation with a window size of 3×3 , 5×5 and adaptive running average of noise-free pixels within 3×3 window, respectively.

3.2 Proposed FoMA algorithm

The pseudo-code of the proposed FoMA algorithm is shown in Algorithm 1. The algorithm exhibits four procedures/subroutines namely TMF3, TMF5, ARA and BPD which are called at different stages of filtering. The TMF3 and TMF5 parallelly computes the median of given

3×3 and 5×5 noisy window, respectively while eliminating all extreme pixels. At the first stage, the algorithm denoises each pixel using TMF3. If the denoised pixel is found noisy, then the denoising is done via TMF5 (see line number 5 to 11 of algorithm 1). These TMF3 and TMF5 return noisy pixels if all pixels in their respective window are extreme (see line number 28 to 41 of algorithm 1). Therefore, the stage-1 provides good results when noise density is low. However, the output from the stage-1 contains several noisy pixels when the noise density is medium or high. Therefore, at the second stage denoising is done using TMF5. It significantly eliminates extreme pixels when the noise density is up to medium level. However, at high noise density, the output image at stage-2 may contain few noisy pixels. These pixels are denoised using the procedure ARA (adaptive running average) defined in Algorithm 1 (see line number 42 to 44). Finally, at the last stage, noisy pixels available on the periphery are denoised using the procedure called BPD. This procedure denoises the extreme pixels on the first/last column/row using average of previously processed pixels (see line number 45 to 53 of algorithm 1). Since the proposed algorithm computes the denoised pixels using non-extreme pixels at first and second stage and at later stages evaluates using previously process pixels only, it provides significantly improved quality than the existing algorithms.

3.3 Illustration of proposed algorithm

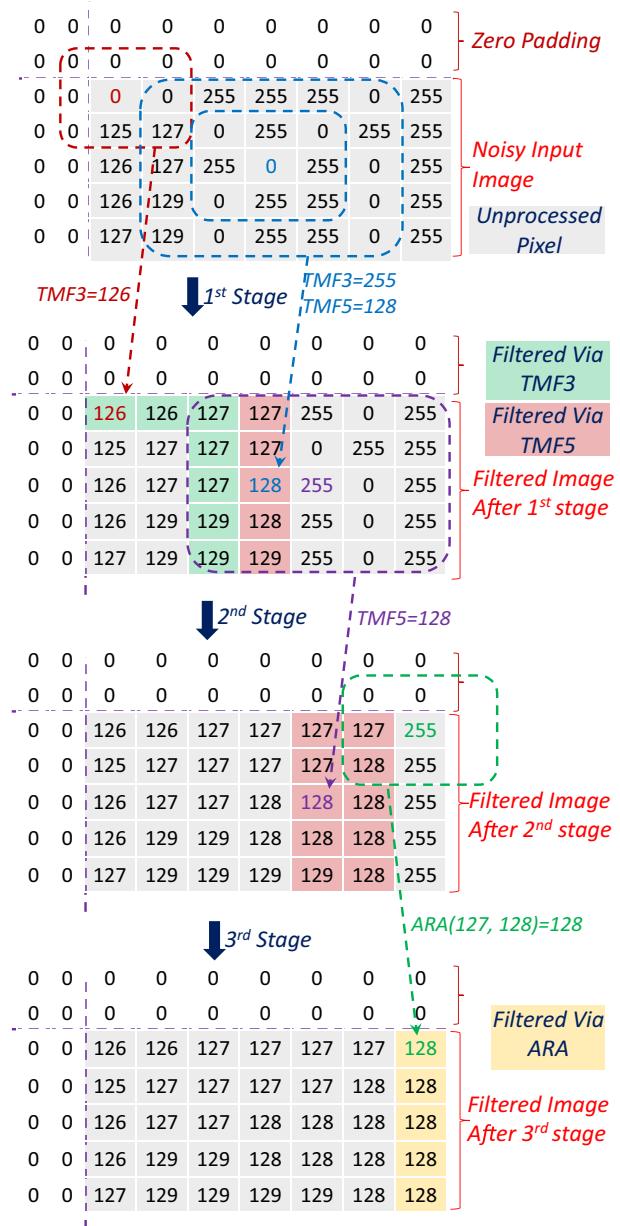
An example shown in Fig. 2 illustrates the processing steps of the proposed FoMA filter on an image segment. Initially, a noisy image-segment is created and zero padding is done at the periphery. However, the zero padding is done on all four sides of the image, padding on left and top side of the image is shown in Fig. 2 just for illustration. The resulting noisy image-segment is applied to the first stage of the proposed filter. The output image segment as shown in figure contains pixels of gray, green and red colors which reflect unprocessed, processed pixels using TMF3 and TMF5 respectively. It can be seen from the figure that there are several unprocessed noisy pixels (last three columns) in first stage output image-segment. This image-segment is then applied to the second stage where each extreme pixel is processed by TMF5 only. The output of second stage as shown in the figure reflects that image segment still contains few extreme pixels (last column). Therefore, this image segment is further processed at third stage where adaptive running average of with window size of 3×3 is considered. Finally, image-segment from the output of third stage is free from all noise. At very high noise density, there might be some noisy pixels at the output of the third stage which could be healed by the fourth stage of the proposed filter.

The analysis of simulation results for the proposed filter is demonstrated in next section.

4 Simulation results and analysis

The efficacy of the proposed FoMA filter is analysed over the existing techniques namely SMF [14], DBAMF [28], ITSAMFT [7], MDBUTM [11], FSBMMF [31], RSIf [30], PDBM [5], DAMF [10], BPDM [9] and FAHPF [4] using different benchmark images such as Cameraman, House, Jetplane, Lake, Mandrill, Peppers, Pirate, Walkingbridge and medical images (X-ray image). Further, colour benchmark images are also considered for the quality analysis towards wider applicability. All the filtering algorithms are implemented in MATLAB (Version 2017a) and simulations are performed on a computer with Intel i5 CPU and 8GB of RAM. Initially, the salt and pepper noise of varying noise density (wide range from 10% to 90%) is introduced in these benchmark images. The noise density (ND) represents

Fig. 2 Illustration of the proposed FoMA filter



the number of corrupted pixels to the total number of pixels and can be expressed as

$$ND = \frac{N_{NP}}{N_{NP} + N_{NFP}} \quad (7)$$

where, N_{NP} and N_{NFP} represent the number of noisy and noise-free pixels, respectively. The MATLAB inbuilt function (`imnoise(InputImage, 'salt & pepper', ND)`) is used to add noise in the Input image of ND amount.

The resulting noisy images are filtered via proposed and existing filtering techniques and finally, the quality metrics are extracted. Quantitative and qualitative analysis is done based on the extracted quality metrics and reconstructed images, respectively.

The quality metrics considered for this analysis are the mean absolute error (MAE), image enhancement factor (IEF), peak signal to noise ratio (PSNR) and structural similarity index (SSIM) [32]. Let, $oImg$, $nImg$ and $rImg$ are original, noisy and reconstructed images, respectively. The mathematical expression of MAE is given by

$$MAE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |oImg(i, j) - rImg(i, j)| \quad (8)$$

Where, the parameters M and N represent number of row and column, respectively. The mathematical expression of IEF is given by

$$IEF = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (oImg(i, j) - nImg(i, j))^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (oImg(i, j) - rImg(i, j))^2} \quad (9)$$

Similarly, the PSNR in dB is given by

$$PSNR(db) = 10 \log(I_{max}^2 / MSE) \quad (10)$$

where I_{max}^2 and MSE are the maximum value of the signal and mean square error respectively. Whereas, the value of SSIM is given by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (11)$$

where μ_x (σ_x) and μ_y (σ_y) represent mean (standard deviation) value in x and y directions, respectively. Further, C_1 and C_2 represent constants to limit the SSIM value to 1.

4.1 Analysis on the wide noise density range

The Lena images (512×512) with 10% to 90% noise density are filtered via proposed and existing filters. The quality metrics (PSNR and SSIM) are extracted and summarized in Table 1. The simulation results demonstrate that the proposed FoMA algorithm provides higher value of the PSNR and SSIM at each sample of noise density over the existing algorithms. On an average, the proposed filter provides 2.09 dB and 0.018 higher values of PSNR and SSIM, respectively over the best known algorithm [31].

Further, the PSNR and SSIM of images filtered via different algorithm for varying noise densities are compared through the plots shown in Fig. 3. From the Fig. 3b, it can be observed that SSIM value of SMF, PDBM and BPDM decreases rapidly with noise density even when the noise density is smaller than 50% whereas, DAMF and ITSAMFT could provide acceptable value of SSIM up to 70% noise density. Whereas, the quality metrics plots of the proposed filter show high value of SSIM and PSNR for each sample of noise density and even at very high noise density (90%) (Table 2).

Finally, for qualitative analysis, the filtered Lena images are shown in Fig. 4. These images show that proposed filter recovers Lena image of vary high perceptual quality even at very high density (90%) over others. However, the images filtered via DBAMF, FSBMMF, RSIf and FAHPF show good image quality but the proposed filter provides images with better edge preservation over all these methods.

Table 1 Quality metrics (PSNR, SSIM, IEF and MAE) and complexity (Execution-time) of proposed and existing salt and pepper removal techniques for Lena image with varying noise density

Metrics	Noise Density (%)	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSIf [30]	PBDM [5]	DAMF [10]	BPD M [9]	FAHPF [4]	Proposed
PSNR (dB)	10	40.04	40.03	37.24	34.90	41.19	41.54	25.80	34.90	23.78	41.15	42.82
	20	32.11	36.37	36.45	34.16	37.29	37.39	20.76	34.16	20.58	36.77	39.21
	30	25.87	33.55	35.08	33.37	34.32	34.73	18.04	33.38	18.69	33.70	36.86
	40	20.77	31.39	32.59	32.34	32.40	32.62	16.34	32.34	17.33	31.70	34.83
	50	16.73	29.51	29.75	30.99	30.92	30.99	15.25	30.98	16.28	29.73	33.25
	60	13.51	27.47	26.49	28.20	29.56	29.21	14.49	28.22	15.30	28.11	31.77
	70	10.85	25.36	22.98	24.41	28.22	27.46	13.56	24.38	14.17	26.59	30.19
	80	8.76	22.89	18.74	20.25	26.72	25.31	11.87	20.24	12.88	25.62	28.50
	90	7.00	19.33	12.99	15.98	24.26	22.35	9.08	15.98	10.68	24.11	26.23
	Avg	19.52	29.54	28.03	28.29	31.65	31.29	16.13	28.29	16.63	30.83	33.74
SSIM	10	0.9968	0.9967	0.9957	0.9831	0.9970	0.9968	0.8418	0.9831	0.7801	0.9970	0.9976
	20	0.9713	0.9920	0.9923	0.9811	0.9928	0.9924	0.6295	0.9811	0.6543	0.9928	0.9949
	30	0.8655	0.9838	0.9848	0.9778	0.9854	0.9863	0.4843	0.9778	0.5665	0.9852	0.9908
	40	0.6344	0.9736	0.9600	0.9731	0.9771	0.9780	0.4033	0.9731	0.5024	0.9760	0.9857
	50	0.4023	0.9593	0.9038	0.9601	0.9670	0.9679	0.3555	0.9601	0.4507	0.9633	0.9789
	60	0.2383	0.9344	0.8117	0.9108	0.9332	0.9502	0.3222	0.9108	0.3998	0.9459	0.9696
	70	0.1318	0.8926	0.7019	0.7716	0.9327	0.9221	0.2793	0.7716	0.3194	0.9227	0.9552
	80	0.0734	0.8056	0.5439	0.5466	0.8990	0.8690	0.2059	0.5466	0.2281	0.8952	0.9309
	90	0.0332	0.6101	0.2877	0.2765	0.8139	0.7487	0.1084	0.2765	0.1170	0.8365	0.8775
	Avg	0.4830	0.9053	0.7980	0.8201	0.9465	0.9346	0.4033	0.8201	0.4465	0.9461	0.9646
IEF	10	243.29	320.31	66.83	88.65	374.43	415.36	11.58	88.68	6.94	416.69	559.37
	20	169.90	247.81	261.20	149.76	292.17	305.33	6.63	149.67	6.43	391.21	476.77
	30	115.97	197.27	284.14	184.92	231.69	256.62	5.40	184.29	6.36	392.72	415.62
	40	88.14	155.11	207.84	196.55	201.20	207.51	4.91	196.54	6.19	226.46	346.48
	50	44.43	123.44	132.48	172.37	172.20	171.59	4.80	171.76	6.05	207.24	293.15

Table 1 (continued)

Metrics	Noise Density (%)	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSIf [30]	PBDM [5]	DAMF [10]	BPD M [9]	FAHPF [4]	Proposed
MAE	60	16.61	97.05	76.18	111.26	153.84	151.63	4.86	110.48	5.83	186.81	255.25
	70	6.45	68.47	39.70	57.36	137.89	113.67	4.58	57.21	5.31	130.39	210.86
	80	2.91	43.69	16.91	24.14	104.23	75.43	3.47	24.12	4.30	70.71	162.37
	90	1.58	22.20	5.04	10.06	66.43	43.35	2.06	10.06	2.94	55.72	106.51
	Avg	76.59	141.70	121.15	110.56	192.68	193.39	5.36	110.31	5.59	230.88	314.04
	10	0.46	0.44	0.80	2.61	0.41	0.40	1.51	2.61	3.85	0.57	0.35
	20	1.01	0.96	0.91	2.82	0.91	0.89	4.59	2.82	8.23	0.86	0.76
	30	1.68	1.58	1.40	3.06	1.48	1.44	8.12	3.06	12.41	1.27	1.19
	40	2.52	2.32	2.31	3.33	2.14	2.09	11.78	3.33	17.03	1.93	1.69
	50	4.03	3.23	3.91	3.73	2.87	2.83	15.04	3.73	21.60	2.57	2.27
	60	7.85	4.29	6.80	4.48	3.64	3.62	18.00	4.48	26.46	2.97	2.89
	70	17.52	5.89	11.98	6.24	4.49	4.83	22.23	6.25	31.98	3.85	3.67
	80	38.45	8.38	21.74	11.11	5.76	6.67	32.57	11.10	40.05	4.81	4.69
	90	74.90	13.96	45.41	23.47	8.27	10.30	58.97	23.47	53.88	7.60	6.35
	Avg	16.49	4.56	10.58	6.76	3.33	3.67	19.20	6.76	23.94	2.94	2.65
Computation Time (Sec)	10	0.052	0.029	0.630	0.349	0.041	1.472	0.024	0.441	0.672	0.345	0.139
	20	0.088	0.046	0.726	0.352	0.075	2.943	0.049	0.495	1.379	0.345	0.252
	30	0.138	0.072	0.740	0.355	0.109	4.576	0.084	0.531	2.033	0.346	0.379
	40	0.166	0.092	0.863	0.410	0.124	5.324	0.123	0.592	2.940	0.347	0.526
	50	0.212	0.133	0.942	0.412	0.158	7.154	0.192	0.696	4.404	0.347	0.635
	60	0.261	0.170	0.975	0.437	0.184	9.639	0.287	0.783	5.943	0.348	0.930
	70	0.318	0.232	1.027	0.450	0.202	12.142	0.414	0.904	7.119	0.352	1.013
	80	0.380	0.302	1.117	0.474	0.242	16.322	0.572	1.015	8.122	0.365	1.304
	90	0.447	0.385	1.154	0.495	0.297	20.628	0.810	1.182	9.528	0.366	1.401
	Avg	0.229	0.162	0.908	0.415	0.159	8.911	0.284	0.738	4.682	0.351	0.731

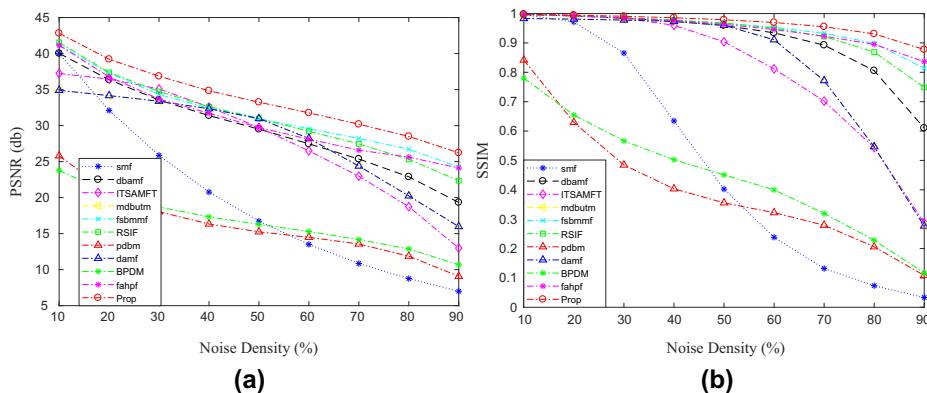


Fig. 3 Quality Metrics **a** PSNR and **b** SSIM Comparison of Lena Images Filtered via the Proposed and Existing Filters with Noise Density from 10% - 90%

In addition, to the PSNR and SSIM metrics, two more quality metrics namely mean absolute error (MAE) and image enhancement factor (IEF) are also computed at varying noise density (10% - 90%) using various benchmark images. The simulation results on an average show 26.48% improvement in the IEF metric than the best known existing filter. To provide more comparison between the proposed and existing algorithms, the execution time is computed as summarized in Table 1. From the simulation results, it can be observed that proposed filter on an average takes less execution time than the ISTAM, RSI, DAMF and BPDM filters. However, the time complexity of the SMF, DBAM, MDBUTM, FSBMM, PBDM, FAHPF is better, the performance (quality metrics) of these filters is very poor over the proposed filter.

4.2 Analysis on the medical images (X-ray)

The proposed filter is also evaluated on medical images where X-ray image (Thigh) is considered for the analysis. These images with 10% to 90% noise density are filtered via

Table 2 PSNR and Execution-time of proposed filter at different stages using Lena image with varying noise density

Noise Density (%)	PSNR (dB)				Execution Time (Sec)			
	Stage-1	Stage-2	Stage-3	Stage-4	Stage-1	Stage-2	Stage-3	Stage-4
10	34.05	37.83	40.68	42.82	0.097	0.111	0.125	0.139
20	31.18	34.64	37.25	39.86	0.176	0.201	0.226	0.252
30	29.31	32.56	35.01	36.86	0.265	0.303	0.341	0.379
40	27.70	30.77	33.09	34.83	0.368	0.420	0.474	0.526
50	26.44	29.37	31.58	33.25	0.445	0.508	0.572	0.635
60	25.26	28.07	30.18	31.77	0.651	0.744	0.837	0.930
70	24.01	26.67	28.68	30.19	0.709	0.810	0.912	1.013
80	22.66	25.18	27.08	28.50	0.913	1.043	1.173	1.304
90	20.86	23.18	24.92	26.23	0.981	1.121	1.261	1.401
Avg	26.83	29.81	32.05	33.74	0.512	0.585	0.658	0.731



Fig. 4 Lena Images (512×512) with Noise Density from 10% to 90% Filtered using: **a** SMF, **b** DBAMF, **c** ITSAMFT, **d** MDBUTM, **e** FSBMMF, **f** RSIf, **g** PDBM, **h** DAMF, **i** BPDM, **j** FAHPF and **k** The Proposed FoMA Filter

Table 3 PSNR and SSIM of proposed and existing noise removal techniques for Thigh image with varying noise density

Metrics	Noise Density (%)	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSIf [30]	PBDM [5]	DAMF [10]	BPDM [9]	FAHPF [4]	Proposed
PSNR (dB)	10	38.73	34.33	30.58	30.18	36.94	36.32	25.47	29.85	23.48	36.54	35.53
	20	31.17	32.30	32.58	28.04	33.63	33.48	19.75	27.72	20.13	33.34	33.49
	30	24.73	29.70	31.81	25.88	31.16	31.24	16.90	25.52	18.18	31.03	31.88
	40	19.42	27.63	30.21	23.69	29.42	29.53	15.33	23.39	16.74	29.14	30.60
	50	15.17	25.83	28.35	21.49	28.04	28.33	14.73	21.21	15.79	27.44	29.28
	60	12.30	23.93	26.62	19.41	27.06	26.40	14.04	19.14	14.80	26.01	28.28
	70	9.68	22.01	24.32	16.75	25.61	24.60	13.94	16.56	14.26	24.67	26.45
	80	7.63	19.98	21.78	14.09	24.62	22.63	13.79	13.94	13.61	23.72	25.20
	90	5.75	16.87	17.40	11.10	22.32	19.69	13.25	11.00	13.11	22.25	22.85
Avg		18.29	25.84	27.07	21.18	28.76	28.02	16.35	20.93	16.68	28.24	29.28
SSIM	10	0.9887	0.9771	0.9574	0.9084	0.9790	0.9774	0.8654	0.9084	0.6973	0.9800	0.9780
	20	0.9555	0.9619	0.9659	0.8795	0.9651	0.9669	0.6137	0.8795	0.5811	0.9666	0.9683
	30	0.8435	0.9570	0.9523	0.8419	0.9433	0.9506	0.4141	0.8419	0.5138	0.9451	0.9553
	40	0.6269	0.8998	0.9271	0.7790	0.9189	0.9300	0.2915	0.7790	0.4683	0.9189	0.9414
	50	0.3758	0.8628	0.8875	0.6992	0.8943	0.9084	0.2397	0.6992	0.4333	0.8903	0.9234
	60	0.1983	0.8066	0.8279	0.6002	0.8670	0.8734	0.2050	0.6002	0.4044	0.8551	0.9042
	70	0.0962	0.7235	0.7406	0.4466	0.8210	0.8172	0.1844	0.4466	0.3812	0.8054	0.8678
	80	0.0440	0.6131	0.6278	0.2701	0.7737	0.7431	0.1715	0.2701	0.3544	0.7583	0.8241
	90	0.0122	0.4201	0.4668	0.1243	0.6511	0.5668	0.1840	0.1243	0.3129	0.6650	0.7341
Avg		0.4601	0.8002	0.8170	0.6166	0.8681	0.8593	0.3522	0.6166	0.4607	0.8650	0.8996

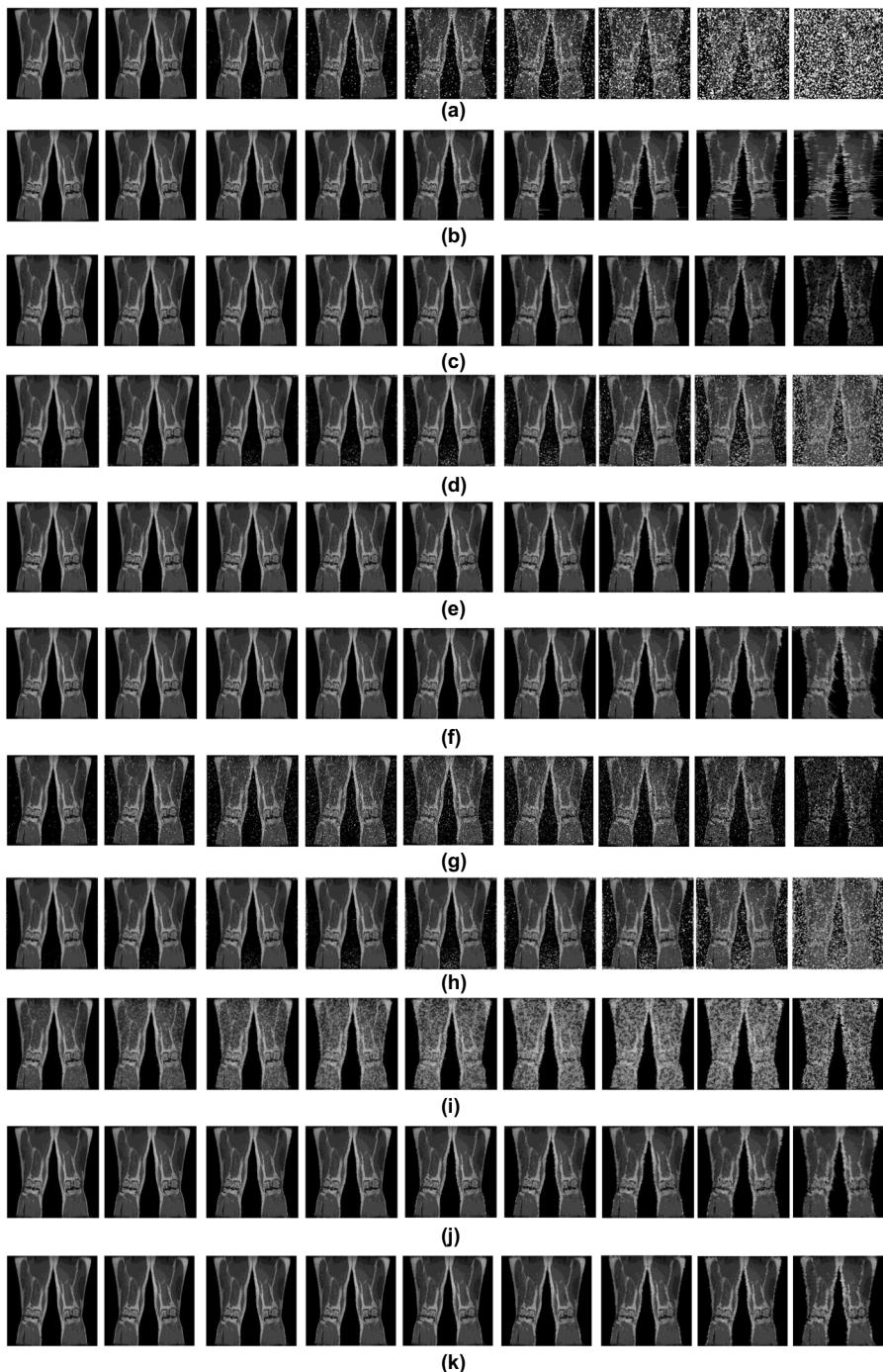


Fig. 5 Thigh Images (512×512) with Noise Density from 10% to 90% Filtered using: **a** SMF, **b** DBAMF, **c** ITSAMFT, **d** MDBUTM, **e** FSBMMF, **f** RSIf, **g** PDBM, **h** DAMF, **i** BPDM, **j** FAHPF and **k** The Proposed FoMA Filter

Table 4 PSNR and SSIM of proposed and existing filtering techniques on various benchmark images at 90% noise density

Metrics	Benchmark	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSlf [30]	PBDM [5]	DAMF [10]	BPDM [9]	UWMF [15]	SAMFWMF [21]	FAHPF [4]	Proposed
PSNR (dB)	Cameraman	6.59	19.17	13.05	14.56	22.93	21.32	9.01	14.56	10.57	15.57	16.01	23.00	25.08
	House	6.71	21.52	12.30	15.02	24.83	23.67	8.29	15.04	10.17	16.07	16.52	25.68	29.73
	Jeiplane	6.38	20.16	10.30	13.92	22.11	21.57	6.51	13.96	9.19	14.90	15.32	22.03	24.67
	Lake	6.51	17.99	12.24	14.17	20.80	20.13	8.53	14.16	10.15	15.17	15.59	21.22	23.11
	Mandrill	7.09	18.96	12.81	16.45	21.17	20.71	9.01	16.45	10.74	17.61	18.10	20.90	22.07
	Pepper	6.77	19.57	13.30	15.26	23.32	22.12	9.29	15.27	10.68	16.33	16.79	23.73	25.90
	Pirate	6.90	19.68	13.59	15.65	22.97	21.86	9.73	15.64	10.85	16.74	17.21	22.95	24.56
	Walkbridge	6.72	18.56	12.97	14.76	19.99	19.84	9.32	14.76	10.51	15.80	16.24	20.16	21.36
Avg		6.71	19.45	12.57	14.98	22.26	21.40	8.71	14.98	10.36	16.02	16.47	22.46	24.56
SSIM	Cameraman	0.0321	0.7454	0.3083	0.2349	0.8242	0.8007	0.1330	0.2349	0.1242	0.2514	0.2584	0.8513	0.8986
	House	0.0305	0.7868	0.2594	0.2289	0.8560	0.8428	0.0873	0.2289	0.0957	0.2449	0.2518	0.8908	0.9374
	Jeiplane	0.0369	0.7465	0.2138	0.2395	0.8025	0.7915	0.0859	0.2395	0.1078	0.2563	.2635	0.8377	0.8901
	Lake	0.0444	0.6249	0.3290	0.2881	0.7477	0.7138	0.1565	0.2881	0.1496	0.3082	0.3169	0.7807	0.8370
	Mandrill	0.0400	0.4997	0.2837	0.3664	0.6203	0.5907	0.1271	0.3664	0.1276	0.3921	0.4031	0.6508	0.7282
	Pepper	0.0311	0.6528	0.3137	0.2608	0.8064	0.7691	0.1207	0.2608	0.1262	0.2790	0.2868	0.8327	0.8669
	Pirate	0.0315	0.5752	0.3198	0.2959	0.7347	0.6813	0.1259	0.2959	0.1231	0.3166	0.3255	0.7638	0.8176
	Walkbridge	0.0414	0.5467	0.3378	0.3354	0.6053	0.5888	0.1541	0.3354	0.1518	0.3588	0.3689	0.6464	0.7263
Avg		0.0360	0.6473	0.2957	0.2812	0.7497	0.7223	0.1238	0.2812	0.1258	0.3009	0.3094	0.7818	0.8378

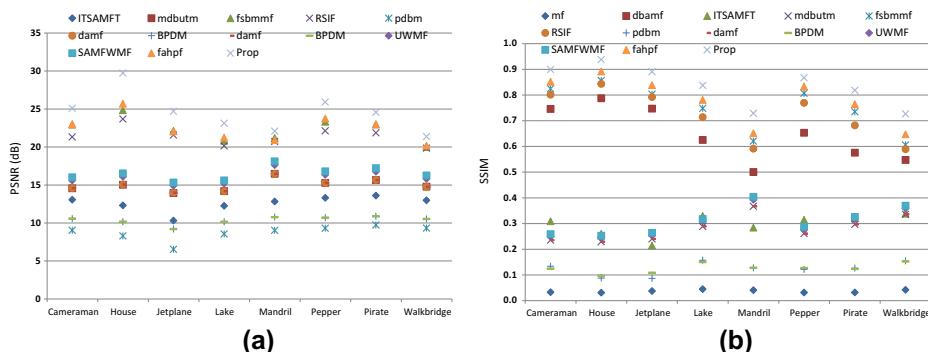


Fig. 6 The scatter plot to compare the performance of the proposed filtering algorithm using various benchmark images at 90% noise density in terms of PSNR and SSIM

different algorithms and the extracted quality metrics are summarized in Table 3. These quantitative metrics demonstrate that proposed FoMA algorithm provides higher value of the PSNR and SSIM at each sample of noise density over the existing algorithms. The proposed filter on an average improves the value of PSNR and SSIM by 0.52 dB and 0.0315, respectively over the best known algorithm [31]. Finally, for qualitative analysis, the filtered X-ray images are shown Fig. 5. The existing filters at very high noise density (90%) fail to recover good medical image while the proposed filter shows X-ray images of very high quality. Therefore, the proposed filter can be effectively utilized for the removal of noise in medical images.

4.3 Analysis on Gray scaled images with different features

The quality analysis of the proposed filter is also done on various benchmark images having different features (or image information). Initially, salt and pepper noise with 90% noise density is introduced and resulting noisy images are filtered via proposed and existing filters. The extracted quality metrics are summarized in Table 4. These results show that the PSNR and SSIM of images filtered via proposed filter exhibit higher value on all benchmarks over the images filtered via existing algorithms. On an average, the proposed filter provides 2.1 dB and 0.056 higher values of PSNR and SSIM, respectively than the best known algorithm [4]. However, the images with large edges (Mandrill and Walkbridge) exhibit poor quality metrics over the smooth images (House and Cameraman) for each sample of noise density using all the filters.

The comparative quality metrics analysis of the proposed filter than the existing is illustrated by the plots shown in Fig. 6 where proposed filter shows higher PSNR and SSIM values in comparison to the existing filters for all benchmark images. Among the various benchmarks, House and Walkbridge images show maximum and minimum value of quality metrics respectively. Finally, for the qualitative analysis, the recovered images are illustrated in Fig. 8. The figure shows that the proposed filter produces images with better perceptual vision in comparison to the images extracted from the existing filters.

4.4 Analysis on very high noise density (90% - 98%)

For quality analysis at very high noise density, Lena images with 90% to 98% noise density are generated. These images are filtered via proposed and existing salt and pepper

Table 5 PSNR and SSIM of proposed and existing salt and pepper removal techniques for Lena image with varying noise density

Metrics	Noise Density (%)	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSIf [30]	PBDM [5]	DAMF [10]	BPDM [9]	FAHPF [4]	Proposed
PSNR (dB)	90	6.93	19.48	12.99	15.94	24.06	22.11	9.08	15.93	10.64	23.98	26.08
	91	6.80	18.79	12.25	15.49	23.78	21.77	8.72	15.48	10.31	23.84	25.72
	92	6.60	18.17	11.54	15.05	23.33	21.28	8.38	15.05	10.01	23.55	25.43
	93	6.46	17.74	10.96	14.68	22.78	20.83	8.09	14.67	9.79	23.12	25.13
	94	6.32	17.20	10.22	14.33	22.11	20.37	7.75	14.33	9.37	22.74	24.76
	95	6.18	16.44	9.43	13.91	21.64	19.59	7.39	13.91	8.93	22.31	24.04
	96	6.04	15.79	8.81	13.58	20.95	19.09	7.09	13.58	8.53	21.79	23.43
	97	5.90	15.03	7.98	13.19	19.52	18.05	6.71	13.19	7.92	20.41	22.09
	98	5.72	14.14	7.16	12.74	18.45	17.30	6.33	12.73	7.25	19.52	19.99
Avg		6.33	16.97	10.15	14.32	21.85	20.04	7.73	14.32	9.19	22.36	24.07
SSIM	90	0.0311	0.6208	0.2857	0.2759	0.8128	0.7438	0.1094	0.2759	0.1144	0.8344	0.8756
	91	0.0292	0.5688	0.2499	0.2493	0.8009	0.7207	0.0945	0.2493	0.1022	0.8253	0.8639
	92	0.0263	0.5501	0.2255	0.2274	0.7807	0.6928	0.0876	0.2274	0.0958	0.8113	0.8543
	93	0.0202	0.4946	0.2025	0.1984	0.7614	0.6729	0.0812	0.1984	0.0895	0.7959	0.8448
	94	0.0203	0.4485	0.1730	0.1780	0.7360	0.6393	0.0711	0.1780	0.0812	0.7757	0.8266
	95	0.0200	0.3973	0.1373	0.1575	0.7052	0.5935	0.0552	0.1575	0.0637	0.7507	0.8069
	96	0.0160	0.3418	0.1126	0.1344	0.6697	0.5568	0.0474	0.1344	0.0563	0.7216	0.7828
	97	0.0165	0.2635	0.0831	0.1128	0.6106	0.4918	0.0349	0.1128	0.0437	0.6652	0.7298
	98	0.0098	0.1933	0.0548	0.0866	0.5486	0.4268	0.0245	0.0866	0.0326	0.6060	0.6793
Avg		0.0210	0.4287	0.1694	0.1800	0.7140	0.6154	0.0673	0.1800	0.0755	0.7540	0.8071

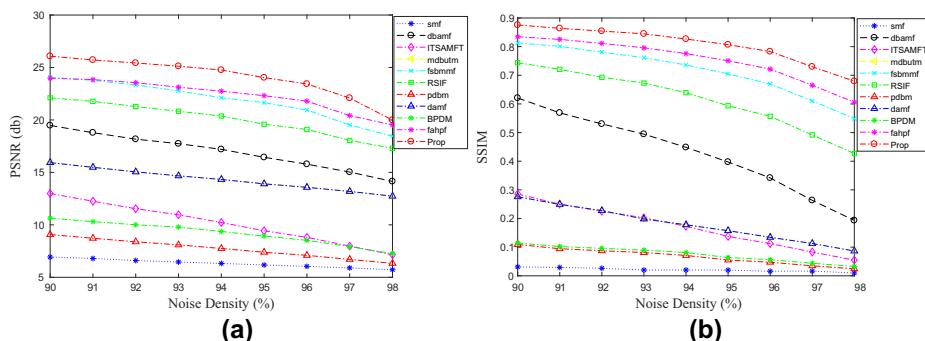


Fig. 7 Quality metrics **a** PSNR **b** SSIM comparison of the proposed and existing filters on Lena image with noise density from 90% to 98%

removal techniques. The extracted quality metrics are summarized in Table 5. The value of the PSNR and SSIM of the proposed filter shows higher quality of filtered images over the images recovered from the existing filters. Only three existing filters namely FSBMMF, RSIF and FAHPF would able to provide images with average PSNR value greater than 20 dB. Whereas, proposed filter an average provides 1.71 dB and 0.0531 higher values of PSNR and SSIM, respectively over the best known algorithm (Figs. 7 and 8).

For comparative quality analysis, PSNR and SSIM for varying ND are plotted as shown in Fig. 7. Finally, for the qualitative analysis, the filtered images are shown in Fig. 9. These images show that proposed filter provides good images even at 97% noise density. However, the visual quality of the images at 98% noise density is not so good but it provides image information of acceptable level.

4.5 Analysis on the colour images at high ND

The efficacy of the proposed algorithm is also evaluated on the color images at high ND (90%). Lena, Mandrill and Pepper images are considered for analysis. Initially, salt and pepper noise with 90% ND is introduced in these images and then filter via proposed and existing filters. The quality metrics of these filtered images are shown in Table 6. The simulation results show 1.06 dB and 0.0478 higher values of PSNR and SSIM, respectively by the proposed filter over the existing.

For qualitative analysis, these output filtered images are also illustrated in Fig. 10. The restored images via proposed filter exhibit good edge details without any artefact or other degradation. It is clearly visible from the simulation results of variety of gray scale and color images that proposed four stage median average filter has outperformed all the recent state-of-the-art filtering methods for impulse noise.

5 Conclusions

A novel four stage median average filter that eliminates high density impulse noise is proposed here. The proposed FoMA estimates the value of noisy pixel using original uncorrupted pixels only at first two stages using TMF3 and TMF5. Therefore, it provided images with very good edge details but effectively filtered high noise density up to the 80%. Under very high noise density environment, the proposed filter computes value of candidate noisy



Fig. 8 Various benchmark images (512×512) with 90% noise density filtered using: **a** SMF, **b** DBAMF, **c** ITSAMFT, **d** MDBUTM, **e** FSBMMF, **f** RSIf, **g** PDBM, **h** DAMF, **i** BPDM, **j** FAHPF and **k** the proposed FoMA filter

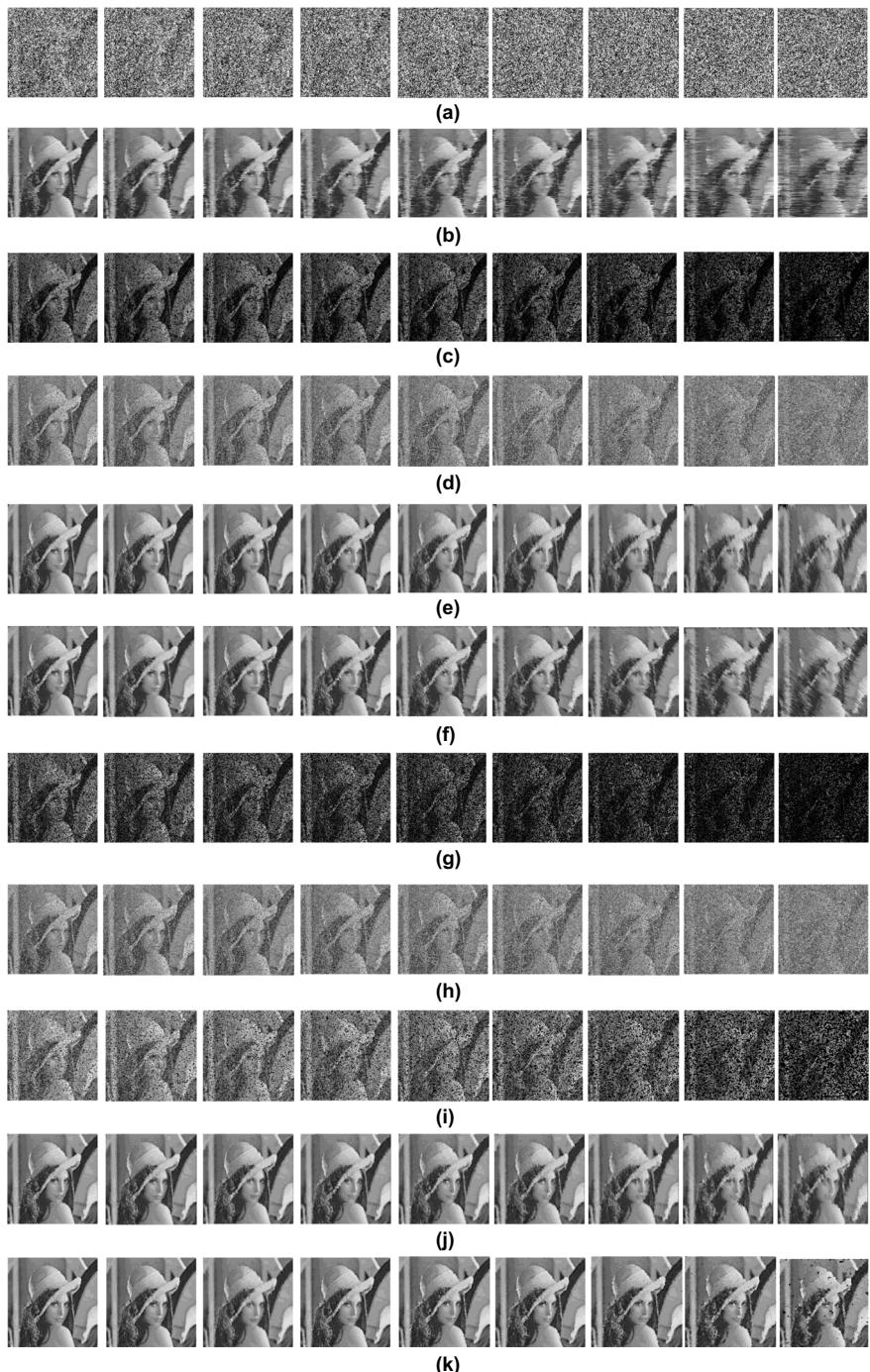


Fig. 9 Lena Images (512×512) with noise density from 90% to 98% Filtered using: **a** SMF, **b** DBAMF, **c** ITSAMFT, **d** MDBUTM, **e** FSBMMF, **f** RSIf, **g** PDBM, **h** DAMF, **i** BPDM, **j** FAHPF and **k** the proposed FoMA filter

Table 6 PSNR and SSIM of proposed and existing noise removal techniques for various colour images at 90% noise density

Metrics	Benchmark	SMF [14]	DBAMF [28]	ITSAMFT [7]	MDBUTM [11]	FSBMMF [31]	RSIf [30]	PBDM [5]	DAMF [10]	BPDM [9]	FAHPF [4]	Proposed
PSNR	Lena	6.67	19.65	12.56	14.86	24.24	22.32	8.61	14.86	10.27	23.96	26.08
	Mandrill	6.69	16.35	12.44	14.25	18.40	17.97	8.94	14.24	10.37	17.99	18.43
	Pepper	6.42	18.21	13.25	13.58	21.71	20.79	9.29	13.56	10.59	22.32	23.83
SSIM	Avg	6.59	18.07	12.75	14.23	21.45	20.36	8.95	14.22	10.41	21.42	22.78
	Lena	0.0309	0.6043	0.2766	0.2491	0.8030	0.7358	0.1041	0.2488	0.1088	0.8082	0.8556
	Mandrill	0.0447	0.4293	0.2809	0.3079	0.5455	0.5112	0.1437	0.3076	0.1362	0.5492	0.6043
SSIM	Pepper	0.0335	0.6211	0.3356	0.2188	0.7672	0.7323	0.1485	0.2177	0.1408	0.7873	0.8281
	Avg	0.0364	0.5516	0.2977	0.2586	0.7052	0.6598	0.1321	0.2580	0.1286	0.7149	0.7627

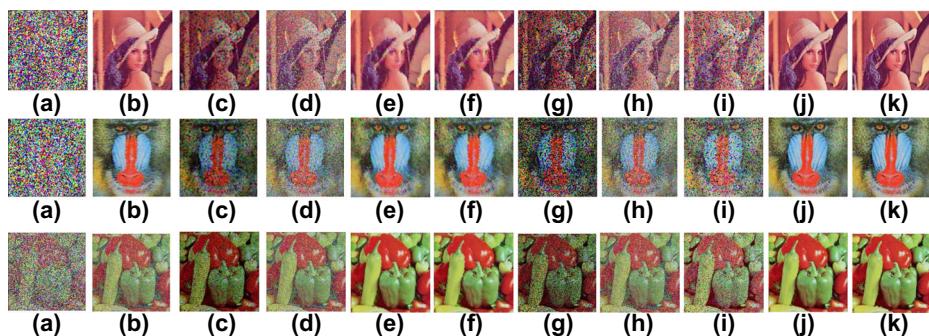


Fig. 10 Various coloured benchmark images (512×512) with 90% noise density filtered using: **a** SMF, **b** DBAMF, **c** ITSAMFT, **d** MDBUTM, **e** FSBMMF, **f** RSIf, **g** PDBM, **h** DAMF, **i** BPDM, **j** FAHPF and **k** the proposed FoMA filter

pixel using adaptive running average filter where only non-extreme pixels are considered. It significantly improved the quality of the recovered images. Finally, at the last stage, the noisy pixels available on the periphery are healed by fourth stage. The quality analysis of the proposed filter on various benchmark images showed that proposed filter provided higher value of PSNR and SSIM for all sample of high noise density images. Finally, the simulation results on an average show 26.48% improvement in IEF metric than the best-known existing filter.

References

1. Ahmed F, Das S (2013) Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean. *IEEE Trans Fuzzy Syst* 22(5):1352–1358
2. Aiswarya K, Jayaraj V, Ebenezer D (2010) A new and efficient algorithm for the removal of high density salt and pepper noise in images and videos. In: 2010 Second international conference on computer modeling and simulation, vol 4. IEEE, pp 409–413
3. Azhar M, Dawood H, Dawood H, Choudhary GI, Bashir AK, Chaudhary SH (2019) Detail-preserving switching algorithm for the removal of random-valued impulse noise. *J Ambient Intell Humaniz Comput* 10(10):3925–3945
4. Balasubramanian G, Chilambuchelvan A, Vijayan S, Gowrisankar G (2016) An extremely fast adaptive high-performance filter to remove salt and noise using overlapping medians in images. *The Imaging Science Journal* 64(5):241–252
5. Balasubramanian G, Chilambuchelvan A, Vijayan S, Gowrisankar G (2016) Probabilistic decision based filter to remove impulse noise using patch-based trimmed median. *AEU-International Journal of Electronics and Communications* 70(4):471–481
6. Burger HC, Schuler CJ, Harmeling S (2012) Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 2392–2399
7. Deivalakshmi S, Palanisamy P (2010) Improved tolerance based selective arithmetic mean filter for detection and removal of impulse noise. In: 2010 5th International conference on industrial and information systems. IEEE, pp 309–313
8. Deivalakshmi S, Palanisamy P (2016) Removal of high density salt and pepper noise through improved tolerance based selective arithmetic mean filtering with wavelet thresholding. *AEU-International Journal of Electronics and Communications* 70(6):757–776
9. Erkan U, Gökkem L. (2018) A new method based on pixel density in salt and pepper noise removal. *Turkish Journal of Electrical Engineering & Computer Sciences* 26(1):162–171
10. Erkan U, Gökkem L, Enginoğlu S (2018) Different applied median filter in salt and pepper noise. *Computers & Electrical Engineering* 70:789–798

11. Esakkirajan S, Veerakumar T, Subramanyam AN, PremChand C (2011) Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. *IEEE Signal Processing Letters* 18(5):287–290
12. Faragallah OS, Ibrahim HM (2016) Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise. *AEU-International Journal of Electronics and Communications* 70(8):1034–1040
13. Garg B (2020) Restoration of highly salt-and-pepper-noise-corrupted images using novel adaptive trimmed median filter. *Signal, Image and Video Processing*, <https://doi.org/10.1007/s11760-020-01695-3>
14. Gonzalez RC, Woods RE (2018) Digital image processing, 4th edn. Pearson, London
15. Kandemir C, Kalyoncu C, Toygar Ö (2015) A weighted mean filter with spatial-bias elimination for impulse noise removal. *Digital Signal Processing* 46:164–174
16. Karthik B, Kumar TK, Vijayaraghavan S, Sriram M (2020) Removal of high density salt and pepper noise in color image through modified cascaded filter. *J Ambient Intell Humaniz Comput*, 1–8
17. Liang S-F, Lu S-M, Chang J-Y, Lin C-T (2008) A novel two-stage impulse noise removal technique based on neural networks and fuzzy decision. *IEEE Trans Fuzzy Syst* 16(4):863–873
18. Lin T-C. (2012) Decision-based filter based on SVM and evidence theory for image noise removal. *Neural Comput and Applic* 21(4):695–703
19. Lu C-T, Chen Y-Y, Wang L-L, Chang C-F (2016) Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window. *Pattern Recogn Lett* 80:188–199
20. Mafi M, Martin H, Cabrerizo M, Andrian J, Barreto A, Adjouadi M (2018) A comprehensive survey on impulse and gaussian denoising filters for digital images. *Signal Processing*
21. Mafi M, Rajaei H, Cabrerizo M, Adjouadi M (2018) A robust edge detectionapproach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering. *IEEE Trans Image Process* 27(11):5475–5490
22. Meher SK, Singhawat B (2014) An improved recursive and adaptive medianfilter for high density impulse noise. *AEU-International Journal of Electronics and Communications* 68(12):1173–1179
23. Nair MS, Shankar V (2013) Predictive-based adaptive switching median filter for impulse noise removal using neural network-based noise detector. *SIViP* 7(6):1041–1070
24. Pitas I, Venetsanopoulos AN (2013) Nonlinear digital filters: principles and applications. Springer Science & Business Media, vol 84
25. Roy A, Laskar RH (2017) Non-causal linear prediction based adaptive filterfor removal of high density impulse noise from color images. *AEU-International Journal of Electronics and Communications* 72:114–124
26. Roy A, Singha J, Devi SS, Laskar RH (2016) Impulse noise removal using SVM classification based fuzzy filter from gray scale images. *Signal Process* 128:262–273
27. Sanaee P, Moallem P, Razzazi F (2019) An interpolation filter based on natural neighbor galerkin method for salt and pepper noise restoration with adaptive size local filtering window, *Signal, Image and Video Processing* 13(5):895–903
28. Srinivasan K, Ebenezer D (2007) A new fast and efficient decision-based algorithm for removal of high-density impulse noises. *IEEE Signal Processing Letters* 14(3):189–192
29. Turkmen I (2016) The ann based detector to remove random-valued impulse noise in images. *J Vis Commun Image Represent* 34:28–36
30. Veerakumar T, Esakkirajan S, Vennila I (2014) Recursive cubic spline interpolation filter approach for the removal of high density salt-and-peppernoise. *Signal Image and Video Processing* 8(1):159–168
31. Vijaykumar V, Mari GS, Ebenezer D (2014) Fast switching based median–mean filter for high density salt and pepper noise removal. *AEU-International Journal of Electronics and Communications* 68(12):1145–1155
32. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
33. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans Image Process* 26(7):3142–3155
34. Zhang K, Zuo W, Gu S, Zhang L (2017) Learning deep CNN denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3929–3938