IJICIS, Vol.20, No.2, 53-66
DOI: 10.21608/ijicis.2020.43046.1030

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/

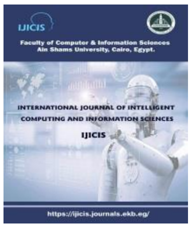

IMAGE DE-NOISING USING INTELLIGENT PARAMETER ADJUSTMENT

Iman Mostafa
Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Egypt
i.m.m.a.167@gmail.com

Ahmed A. Eltahawi*
Information System Department, Faculty of Computers & Informatics, Suez Canal University, Egypt
a.othman@ci.suez.edu.eg

Atef M. Ghuniem
Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Egypt
atmohagh@gmail.com

Received 2020-09-15; Revised 2020-10-26; Accepted 2020-10-31
Available online 2021-1-5

Abstract

Image de-noising is one of the main steps in the medical image analysis process. In medical imaging, noise usually occurs at the capture stage of medical machines such as the ultrasound machines. This noise may hide important information that affects the diagnosing process. Current medical image de-noising techniques still need modifications to enhance their de-noising capabilities, especially traditional parameter dependent techniques such as VisuShrink. This technique has a threshold that needs to be adjusted to efficiently de-noise the images. In this paper, an intelligent framework is proposed to assign a threshold to VisuShrink technique based on the current image features. These features extracted from the image using Scale Invariant-Feature Transform (SIFT) technique are used to train different machine learning (ML) techniques for predicting the appropriate threshold. The experimental results showed that the proposed framework managed to reduce the noise compared to VisuShrink technique with a fixed threshold.

* Corresponding author: Ahmed A. Eltahawi
Information System Department, Faculty of Computers & Informatics, Suez Canal University, Egypt
E-mail address: a.othman@ci.suez.edu.eg

1 Introduction

Image de-noising is the process of removing the unwanted data (noise) acquired during the capture process to restore the original image features. These features may hold important information necessary for accurate diagnosis, therefore, image de-noising became one of the most important topics nowadays.

Medical images used for diagnosing human diseases play a very important role in our healthy life. Most medical imaging machines such as Magnetic Resonance Imaging (MRI) and Ultrasound may add some noise to the images during the capture processes. This noise must be removed or at least reduced without affecting the original image details to be ready for accurate diagnoses. In literature, there are many studies tried to address the problem of medical image de-noising [1, 2]. VisuShrink technique is one of the popular de-noising techniques based on wavelet thresholding [3, 4, 5], however, it has two main drawbacks. First, It may lead to over-smoothing the image resulted in losing some important details taht are required for accurately diagnosing in the medical image field. Second, it applied a fixed equation to calculate the de-noising threshold for all images paying no attention to the image features which may work properly for some images but not for all images. Therefore, there is still a need for an intelligent technique that calculates the de-noising threshold for the image based on its characteristics.

In this research, we propose an intelligent framework to predict the de-noising threshold for the image based on its features. This threshold is then used by VisuShrink de-noising technique to enhance the image. Features are extracted from the images using a popular feature extraction technique (SIFT). These features are used as inputs to train different ML techniques such as Linear Regression (LR), Regression Tree (RT), and Fuzzy System (FS) with the optimum VisuShrink threshold as a target output. The trained model is then used to predict the threshold of new unseen images used by VisuShrink to enhance the images. The results of the proposed framework are compared with the results of using VisuShrink with a fixed threshold.

This paper is organized as follows. Section 2 presents an explanation for VisuShrink de-noising technique, SIFT, and some related work. Section 3 provides a detailed explanation of the proposed framework. Section 4 shows the conducted experiments and the results. Section 5 concludes the paper and suggest some future work.

2 Background & Related Work

In this section, a background review of the main topics discussed in this study will be covered. The VisuShrink de-noising technique and SIFT feature extraction techniques will be explained. Moreover, three de-noising categories namely spatial domain, transform domain, and intelligent based de-noising methods will be covered. In each category, different previous related de-noising techniques will be discussed.

2.1 VisuShrink

Dohono and Johnston [5], proposed a non-adaptive thresholding image de-noising technique called VisuShrink (VISU). It apply a universal threshold calculated using the following fixed equation

$$\lambda_{visu} = \sigma * \sqrt{2 * log(M)} \quad (1)$$

where σ is the standard deviation of the image, and M is the number of pixels in the image. The threshold calculation depends only on the standard deviation and the size of the image which are not enough to calculate an efficient threshold suitable for all images.

Although this technique is widely used, it has some weak points. First, It can not remove speckle noise as it deals only with additive noise. Second, the process of calculating the threshold value is fixed and very basic. It does not take into consideration the image characteristics, therefore, an inaccurate threshold may be resulted and affect the de-noising results. In some cases, it may lead to over-smoothing the image and, therefore, lose some finite details because of the inappropriate threshold calculation strategy. Hence, there is still a need for an intelligent method to calculate the threshold based on the current image features.

2.2 Scale Invariant Feature Transform (SIFT)

Lowe [6], proposed a feature extraction and recognition technique called SIFT. It is used to describe the internal features of the images called descriptors. These descriptors are calculated from the regions around a set of key points represent the most interesting points inside the images. One of its main applications is object recognition which used to detect the object at another image even if it is rotated or on a different scale. SIFT consists of four main steps as follows:

1. Scale space extrema detection: In this step, the key points are detected using different scales of the image.
2. Key point localization: The key points detected at the first step are re-examined and the points with low contrast or localized around edges are discarded.
3. Orientation assignment: The orientation of each key point is assigned to ensure that the points are invariant to orientation.
4. Key point descriptors: In this step, a descriptor vector of dimension 1×128 is calculated for each key point.

2.3 Spatial Domain Filtering

Image de-noising in the spatial domain is a very popular strategy that includes many techniques. These techniques may be classified into Linear and Non-linear techniques [7]:

- Linear Filters:

  Mean and Wiener filter techniques are considered the most popular linear filter techniques. The mean filter uses a moving square window called mask around the corrupted pixel. This pixel is then replaced with the average of its mask neighbors including itself. However, this technique may result in blurring the image edges and losing some fine details of the image. Also, it is ineffective in removing some types of noise such as impulsive noise. On the other hand, Wiener filter [8, 9], is a linear de-noising technique that uses the minimum mean square error to minimize the least-squares error between the de-noised version and actual outputs of the image. However, it requires prior information about the noise power spectra of the original image and the noise. Moreover, it works well when the image has a low variance noise and may result in blurring the edges of the image.

- Non-Linear Filters:

  Median filter [10, 11] is one of the popular non-linear filters denoising techniques invariant to monotonic variations. An odd number square window is created and centered around the image pixel. this pixel is replaced by the median of the pixels inside the window after sorting these values. It has the same principle of the sliding window of the mean filter, however, it acts better in preserving useful details in images. The main drawback of this filter is not performing well in smoothing non-impulsive noise (e.g: additive Gaussian noise) as linear filters.

  Spatial median [12] is also a non-linear filter where the median is calculated by calculating the spatial depth between a point X and a set of points $(x_1 \cdots x_N)$ which helps in finding whether the pixel is noised or not. The weighted median filter is another type of this filter where each pixel has a weight or average differs from the spatial median that it has no empty mask.

2.4 Transform Domain Filtering

De-noising techniques in transform domain filtering may be classified as adaptive data transform and non-adaptive data transform. Dentino [13] introduces the first adaptive data transform denoising technique. Independent Component Analysis (ICA) [14, 15] used in non-Gaussian data de-noising is a popular example of an adaptive data de-noising technique.

On the other hand, Wavelet de-noising is considered a popular example of the non-adaptive data transform de-noising techniques [15]. In the wavelet domain, de-noising occurs by removing some high frequency corresponds to small detail components. De-noising in this domain is classified into wavelet thresholding, wavelet coefficient model and un-decimated wavelet transform (UDWT) based methods.

2.5 Intelligent Based De-noising

There are many de-noising techniques based on intelligent methods. In this study, we used fuzzy rules, linear regression, and regression trees to train our framework. In this section, a review of previous intelligent de-noising techniques is presented.

Linear Regression (LR) is one of the most commonly used predictive techniques [16]. It is used to describe and explain the relationship between one dependent variable and one or more independent variables. It is used in finding the best-fitting straight line through the points called regression line. LR is represented by the following equation:

$$y = X\beta + \varepsilon \tag{2}$$

where y is the dependent variable (response value), X is the independent variables (input variables), intercept β, and the error term ε.

P.S. Hiremath [17] used a linear regression model to remove Gaussian representation of speckle noise on ultrasound images. Dinh Hoan Trinh [18] used support vector regression (SVR) to build a de-noising model for medical images. This model consists of training and de-noising phases. In the training phase, a training set of images is used to train the SVR model given the noise version of the images. In the de-noising phase, the type and the level of the noise in the image are first determined, then, the de-noising model chooses automatically adaptive SVR functions to calculate the value for each pixel in the image. Dinh Hoan Trinh [19] also presented another study in de-noising medical images using kernel ridge regression. He used this method for the reduction of the Gaussian noise of Computed Tomography (CT) image and Rician noise of Magnetic Resonance Imaging (MRI) image. Regression Tree (RT) is another ML technique used widely in removing different types of noise from medical images. It is a tree graph tool simple to follow and understand used in researches and decision analysis. The observations about the object are represented as branches and the conclusions represented as leaves. It is one of the tools that is used also in data mining and statistics for predictive modeling approaches.

Decision trees are classified into classification trees and regression trees. In classification trees, the target variables take a finite set of values. However, in regression trees, the target takes continuous values. RT consists of the beginning point of the trees (root node), leaf nodes, and the branches connecting the nodes.

Remya Ravi Nair [20] proposed a method for de-noising images using the decision tree-based algorithm. This algorithm consists of 3 stages for detecting the noise pixels. The first step is the Isolation module (IM) to determine whether the pixel is in a smooth region or not. The second step is the Fringe module (FM) to locate the edge pixels which may be determined as a noisy pixel in the IM.

The third step is the Similarity module (SM) to solve the presence of noisy signals in the noise-free areas. These modules are followed by edge preserving filter to remove the noise.

Fuzzy rules (FR) is one of the ML techniques used in different applications of prediction. The fuzzy logic system cooperates in the development of image processing and analysis. It attempts to imitate the human visual system in some way. It has been applied in reality in many applications like simplified control of robots, preventing unwanted temperature fluctuations in air-conditioning systems, cruise-control for auto-mobiles, medicine technology, cancer diagnosis, etc. For developing fuzzy logic systems (FLS), three main factors are required [21]: the selection of the fuzzy rule base, The definitions of the membership functions, the structure of the fuzzy system (number of rules and membership functions). Equations of FL scheme in image processing [22] of an image $I(x, y)$ is defined as:

$$F : I(x, y) \Rightarrow I_F(x, y) \tag{3}$$

$$s[I_F(x, y)] = I_{FN}(x, y) \tag{4}$$

$$D : I_F^{\prime}(x, y) \Rightarrow I^{\prime}(x, y) \tag{5}$$

Where F is set of mapping functions to map the image to the fuzzy domain, IF is the image in the fuzzy domain, IF N is the new fuzzy image after a set of fuzzy operators J applied to the fuzzy image, D is the de-fuzzification function to return the fuzzy image to the original domain and Ij(x, y) is the processed image.

In many studies, FL is used to construct different approaches of image de-noising. Mansi Pathak [23] proposed a survey of different studies of de-noising images using several fuzzy based techniques. Giovanni Palma [24] presented how to fuzzify a crisp image depending on the conversion of statistical noise present in an image into a de-noising imprecision. Monika Sharma [25] proposed a new fuzzy-based approach to detects and remove salt and pepper noise in the grayscale image in two steps. Fuzzy logic is first used to detect the noise, which is then replaced using 3×3 mask filter scanning the whole image and removing low-level noise. 3×2 mask is then used to remove noise from the corners and edge, followed by a 4×4 mask for high-level noise. R.Pushpavalli [26] proposed a hybrid de-noising method consists of a new switching median filter and a neuro-fuzzy network to remove different level of salt and pepper noise. The neuro-fuzzy network with parameters optimized by training is found quite effective in removing impulse noise while preserving image boundaries. Nguyen Minh Thanh [27] used a generalized fuzzy inference system (GFIS) which is a neuro-fuzzy system combining Mamdani model and Takagi-Sugeno(TS) fuzzy model. Amaninder Kaur Brar [28] combined the features of the neural network and fuzzy logic to remove various types of noise with preserving the edge sharpness and improving the image contrast.

3 The Proposed Framework

IIn this section, an intelligent framework for predicting the VisuShrink threshold is explained as shown in Figure 1. It consists of three phases namely, the pre-processing, the training and the testing phases. In the pre-processing phase, different types of noise are added to the input images, the optimum VisuShrink thresholds are calculated and the noisy images are randomly divided into training and testing sets. In the training phase, features are extracted from the training images and provided as inputs to any ML technique with the calculated optimum threshold as the target output. In the testing phase, features extracted from the testing images are used by the trained ML technique to predict the threshold. This threshold is then used to de-noise the testing images using VisuShrink and the results are recorded and saved.

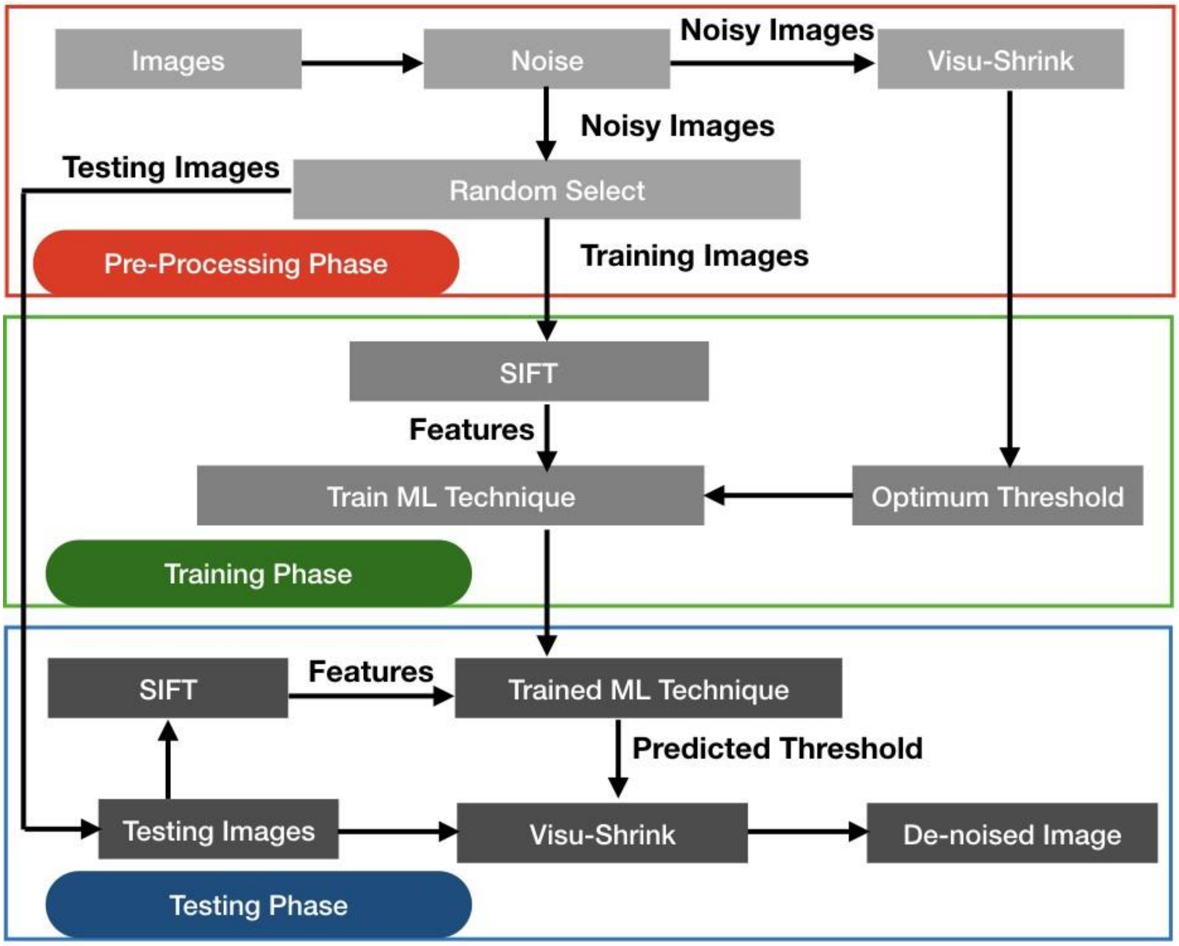

Figure 1: VisuShrink Threshold Prediction Framework

3.1 Pre-processing phase

In this phase, noises are added to the input images, optimum thresholds are calculated and the images are randomly divided into training and testing images (Algorithm 1). This phase proceeds as follows:

- Load the available input medical images. Images could be of any type such as ultrasound and MRI (Algorithm 1, LAI)
- Add different types of noise. In this step, three different types of noise are added to each image separately. These noises are salt and pepper, speckle, and Gaussian. The resulted noisy images are saved for further operations (Algorithm 1, ANT)
- Calculate the optimum threshold. In this step, the optimum VisuShrink threshold is calculated for each noisy image using trial and error. VisuShrink algorithm is applied on each noisy image using different values of thresholds and the PSNR (Equation) between the resulted image and the original image is calculated (Algorithm 1, AVA and CPM). The threshold obtained the maximum PSNR is considered the optimum threshold for this image with current noise. Three different optimum threshold matrices are generated as one for each noise type T_{NS_1} , T_{NS_2} ,and T_{NS_3} (Algorithm 1, SOM)
- Create training and testing sets. in this step, the training set of the noisy image of each type of noise is selected randomly. The remaining images for each type are used as testing images. Three different training sets for each type of noise are created and saved as T_{R_1} , T_{R_2} and T_{R_2} (Algorithm 1, CTM)

Algorithm 1 Pre-Processing Algorithm

```
LAI – Load available input images I_1, I_2, ..., I_n
for Each noise NS do
    for Each image I do
        ANT – Add noise NS_i to I_i and save the noisy image I_{NS_i}
        I_N = F(I)
        for Each threshold t do
            AVA – Apply VisuShrink using t_i for the noisy image I_{NS_i}
            CPM – Calculate and save PSNR_i between I_i and I_{NS_i}
        end for
        SOM – Add t_i to the optimum matrix T_{NS_i} if it has the maximum PSNR
    end for
end for
CTM Randomly create training set images matrix T_R and testing images matrix T_S for each type
    of noise
```

3.2 Training phase

In this phase, features are extracted from noisy images and used to train different ML techniques (Algorithm 2). This phase proceeds as follows:

- Load the available training images created in the preprocessing phase (Algorithm 2, LTS)
- Features extraction. In this step, the feature matrix is constructed of SIFT descriptors. For each image, SIFT is used to detect a set of key points inside the image $P_1, P_2, .., P_n$ (Algorithm 2, DSP). SIFT is then used to calculate the descriptors vector around each point. This vector consists of 128 values appended in a descriptor matrix of $DT_i = n \times 128$, were n is the number of detected key points each represented in a separated row (Algorithm 2, CFD). SIFT usually detect a high number of points that are not all relevant and, therefore, need to be discarded. To do that, the mean of each descriptor is calculated and sorted (Algorithm 2, CDM). The descriptors of the highest m (e.g., $m = 10$) values of the mean are selected to be added to the feature matrix. Therefore, the final descriptors features of the current images is reduced from $DT_i = n \times 128$ to $D'T_i = m \times 128$, where m ™ n and appended to the final feature matrix F (Algorithm 2, PTD and AFM). As a result, F will be of dimension m $*$N $\times$ 128, were N is the number of images available in the training set.
- Train ML technique. In this step, the ML technique is trained using F as input the optimum threshold as output. The trained ML is saved in order to be used in the testing phase (Algorithm 2, TML). In this study, three Ml techniques are used as follows:
  1. Linear Regression (LR)
  2. Regression Tree (RT)
  3. Fuzzy Rules (FR)
- This process is repeated for each type of noise.

3.3 Testing phase

In this phase, the trained ML technique is used to predict the threshold of each image based on its SIFT features. This phase proceeds as follows:

Algorithm 2 Training and Testing Phases Algorithm

```
for Each noise type NS do
    ---------------Training Phase ---------------
    LTS – Load the training images set T_R
    for Each image I_T in T_R do
        DSP – Use SIFT to detect key points P_1, P_2, ..., P_n in I_{T_i}
        CFD – Calculate the SIFT descriptors of I_{T_i} (DT_i = n × 128), n is the number of SIFT point
        detected in I_{T_i}
        CDM – Calculate the mean μ_i of each descriptor
        PTD – Pick the top descriptors in terms of mean D'T_i = m × 128 (m ™ n)
        AFM – Append D'T_i to the general feature matrix F
    end for
    TML – Train the available ML technique ML_i with F as input and T_{NS_i} as a target output
    ---------------Testing Phase ---------------
    LSS – Load the testing images set T_S
    for Each image I_S in T_S do
        ETM – Calculate the SIFT descriptors of I_{S_i} (D_{S_i} = m × 128), m is the number of reduced
        SIFT point detected in I_{S_i}
        SML – Test the trained ML ML_i with D_{S_i} as input and save the predicted t_i of I_{S_i}
        DTI – De-noise I_{S_i} using VisuShrink with t_i and save the de-noised image I_{F_i}
        CPM – Calculate and save PSNR_i between I_i and I_{F_i}
    end for
end for
```

- Load the testing images set (Algorithm 2, LSS)
- Predict the VisuShrink Threshold. In this step, the ML technique is used to predict the threshold of the current image. For each image, SIFT is used to extract feature descriptors around a set of key points. This descriptor is calculated in the same way as described in the training phase (Algorithm 2, ETM). These descriptors are provided to the trained ML technique to predict the threshold t_i of the current image (Algorithm 2, SML).
- De-noise the image. In this step, the predicted threshold t_i is used through VisuShrink to de-noise the current image (Algorithm 2, SML). The resulted filtered image I_{F_i} is saved for evaluation (Algorithm 2, DTI).
- Calculate the PSNR between the filtered image I_{F_i} and the original image I_i (Algorithm 2, CPM)

4 Experiments & Results

In this section, a set of experiments along with their results are explained and discussed. Three different experiments (one for each noise type) are employed to evaluate the performance of the proposed framework. A set of 35 ultrasound images with sizes ranges from 249×346 to 759×584 were used are used in these experiments divided randomly into 60% for training and 40% for testing. The three experiments are constructed by adding salt and pepper, speckle, and Gaussian noises to each image separately. Each experiment is repeated n times (e.g., $n = 5$), and the average results were recorded and presented.

4.1 Evaluation Metrics

The proposed framework is evaluated by comparing its de-noised image with the original image. This comparison is performed by calculating the PSNR between the two images. The results of the proposed framework are compared with the results of VisuShrink using its fixed equation. Moreover, the visual comparison between the resulted image and the original image is also presented.

Three evaluation metrics are used to evaluate the performance of the proposed framework as follows:

1. Peak Signal to Noise Ratio (PSNR): PSNR is used to measure the ratio between the noise and the signal power affected by this noise [29]. It is presented by the mean square error (MSE) of an image I with dimensions $m \times n$ and a noisy image K. MSE is calculated using the following equation:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)^2 \tag{6}$$

   And PSNR is then calculated using the following equation:

$$PSNR = 20 * \log_{10} \left(\frac{MAXI}{\sqrt{MSE}} \right) \tag{7}$$

   where MAX_I is the maximum pixel value at the image I.

2. The structural similarity (SSIM) index: This metric is used to evaluate the quality of digital images [30]. SSIM between two images x and y with the same size $M \times N$ is calculated using the following equation:

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy}) + c_2}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2) + c_2}$$

   where μ_x and μ_y are the average of x and y, $sigma^2_x$ and σ^2_y are the variance of x and y, σ_{xy} is the co-variance of x and y, and c_1 and c_2 are division stabilizer variables.

3. Visual comparison. The images of the proposed approach using the three ML techniques are presented to be compared with the original image and the result of VisuShrink using a fixed equation (Fig. 3, Fig. 4).

4.2 Results and Discussions

In this section, the results of the enhancing salt and pepper, speckle, and Gaussian noise in the three mentioned experiments are presented. Each experiment is repeated five different times and their results, as well as their average, are presented.

For the images with salt and pepper noise, Table 1 presents the PSNR and the SSIM results of the three ML techniques used in the proposed approach and the results of the VisuShrink technique. It can be seen that the proposed approach achieved better results than VisuShrink for the three ML techniques in all runs. In specific, FR technique managed to achieve the best results among the three ML techniques with an average PSNR of 21.094 and SSIM of 0.399. As well, it is clear from Figure 2 that the proposed approach regardless of the ML technique used managed to enhance the salt and pepper noise in ultrasound images better than the VisuShrink technique.

	Metrics	LR	RT	FR	VisuShrink
1st run	PSNR	20.755	20.746	20.745	20.127
	SSIM	0.388	0.388	0.388	0.364
2nd run	PSNR	21.175	21.163	21.160	20.481
	SSIM	0.416	0.416	0.416	0.383
3rd run	PSNR	21.502	21.501	21.486	20.701
	SSIM	0.407	.406	0.406	0.365
4th run	PSNR	21.318	21.310	21.302	20.476
	SSIM	0.418	0.418	0.418	0.379
5th run	PSNR	20.795	20.790	20.779	20.153
	SSIM	0.371	0.371	0.370	0.349
Average	PSNR	21.109	21.102	21.094	20.388
	SSIM	0.400	0.400	0.399	0.368

Table 1: PSNR and SSIM Results of ultrasound images with Salt & Pepper noise

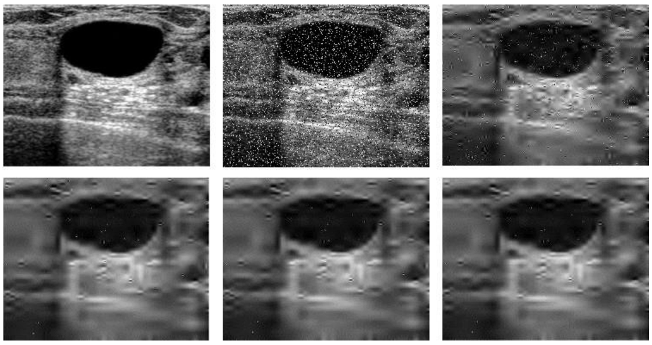

Figure 2: Results of an ultrasound image with salt & pepper noise from left to right: Original image, Noisy image, VisuShrink Result, Linear Regression Result, Regression Tree result, and Fuzzy Results

For the speckle noise, Table 2 summarizes the PSNR and the SSIM results of the proposed approach and the results of the VisuShrink technique. In all runs, LR, RT, and FR beat the VisuShrink technique in both PSNR and SSIM. RT ML technique came in the first place in enhancing speckle noise with PSNR of 24.030 and SSIM of 0.527. For the visual results, it could be seen from Figure 3 that the proposed approach for all ML techniques has better quality in enhancing the speckle than the VisuShrink technique.

	Metrics	LR	RT	FR	VisuShrinkS
1st run	PSNR	23.460	23.443	23.445	23.246
	SSIM	0.521	0.520	0.523	0.521
2nd run	PSNR	24.029	24.007	24.009	23.468
	SSIM	0.540	0.537	0.542	0.540
3rd run	PSNR	24.478	24.487	24.529	24.235
	SSIM	0.510	0.511	0.519	0.510
4th run	PSNR	24.629	24.627	24.704	24.026
	SSIM	0.544	0.544	0.553	0.544
5th run	PSNR	23.480	23.465	23.464	23.192
	SSIM	0.497	0.494	0.500	0.497
Average	PSNR	24.015	24.006	24.030	23.633
	SSIM	0.522	0.521	0.527	0.522

Table 2: PSNR and SSIM Results of ultrasound images with Speckle noise

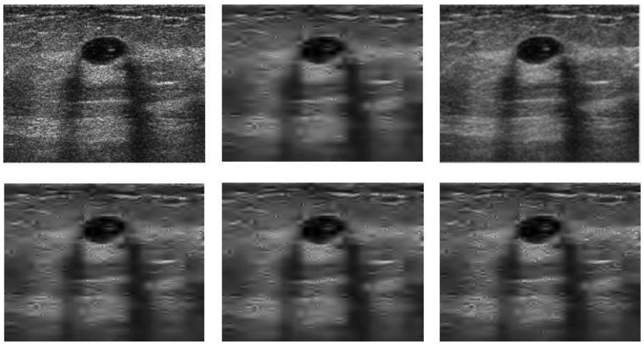

Figure 3: Results of an ultrasound image with Speckle noise from left to right: Original image, Noisy image, VisuShrink Result, Linear Regression Result, Regression Tree result, and Fuzzy Results

For the Gaussian noise, Table 3 summarizes the PSNR and the SSIM results of the proposed approach and the results of the VisuShrink technique. The proposed approach has the best results for all runs with a little preference to LR ML technique in enhancing the gaussian noise. As well, the visual results presented in Figure 4 prove that the proposed approach figures managed to enhance the gaussian results better than the fixed VisuShrink technique.

	Metrics	LR	RT	FR	VisuShrinkS
1st run	PSNR	18.923	18.917	18.905	18.729
	SSIM	0.352	0.352	0.351	0.339
2nd run	PSNR	19.224	19.224	19.211	19.046
	SSIM	0.382	0.382	0.381	0.370
3rd run	PSNR	19.619	19.616	19.606	19.456
	SSIM	0.375	0.374	0.373	0.364
4th run	PSNR	19.351	19.352	19.339	19.180
	SSIM	0.382	0.382	0.380	0.370
5th run	PSNR	19.016	19.016	19.007	18.838
	SSIM	0.339	0.339	0.338	0.326
Average	PSNR	19.226	19.225	19.213	19.050
	SSIM	0.366	0.366	0.365	0.354

Table 3: PSNR and SSIM Results of ultrasound images with Gaussian noise

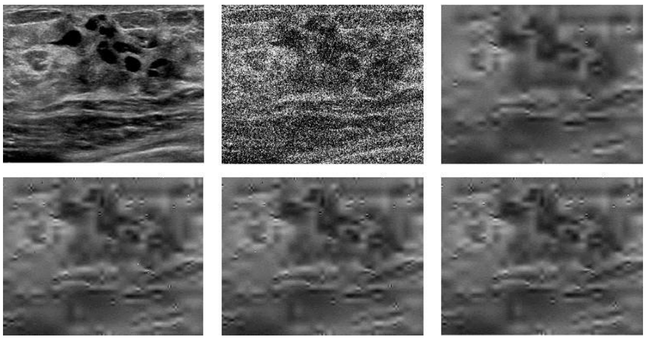

Figure 4: Results of an ultrasound image with Gaussian noise from left to right: Original image, Noisy image, VisuShrink Result, Linear Regression Result, Regression Tree result, and Fuzzy Results

5 Conclusion

Image denoising is a current challenging problem in image processing analysis. Several traditional techniques were used for solving this problem. These techniques use the same strategy to enhance different types of images including different types of noises. They do not pay any attention to the characteristics and the differences between these images. Therefore, these techniques may achieve good results for some images but not for all images. An intelligent de-noising technique was pro- posed in this paper to enhance different types of noises in medical ultrasound images. The framework uses the features of the image to train different ML techniques to predict the threshold of VisuShrink de-noising technique. The results of the framework in terms of PSNR and SSIM prove the efficiency of the framework in enhancing different types of noises compared with VisuShrink with a fixed equation. In future work, the proposed technique will be compared with the techniques employed to self estimate the visushrink threshod.

Bibliography

[1] Sonali Singh ,and Sulochana Wadhwani. Medical Image Denoising Using Sub Band Adaptive Thresholding Techniques Based on Wavelet 2D Transform, International Journal of Bio-Science and Bio-Technology, vol.7, no. 2, pp. 327-334, 2015.

[2] Hedaoo1 and Swati S Godbole. Wavelet Thresholding Approach For Image Denoising, International Journal of Network Security & Its Applications (IJNSA), vol.3, no.4, July 2011.

[3] Rong Bai, Wavelet Shrinkage Based Image Denoising using Soft Computing, 2008.

[4] D. L. Donoho, Denoising by soft-thresholding, IEEE Trans. Inf. Theory, vol. 41, no. 3, pp. 613-627, Mar. 1995.

[5] D. L. Donoho and I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika, vol. 81, no. 3, pp. 425-455, 1994

[6] David G. Lowe,Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision. pp. 1150-1157, Sept. 1999.

[7] Motwani,M.C., Gadiya,M.C., Motwani,R.C., Harris,F.C Jr. Survey of Image Denoising Techniques, IJCSMC, 2014.

[8] Kailath T., Equations of Wiener-Hopf type in filtering theory and related applications, in Norbert Wiener: Collected Work, vol.III, P.Masani, ed. Cambridge, MA: MIT Press, pp. 63-94, 1976.

[9] R. Kimiaefar, Siahkoohi H. R., Hajian A., Kalhor A., Random noise attenuation by Wiener-ANFIS filtering, Journal of Applied Geophysics. pp. 453-459, Volume 159, 2018.

[10] R.Gonzalez, and R.Woods,Digital Image Processing, Adison-Wesley, New York,1992.

[11] Sajjad Ahmed, Saiful Islam, Median filter detection through streak area analysis, Digital Investigation. pp. 100-106, Volume 26, 2018,

[12] James C Church YixinChen, and Stephen V Rice , A Spatial Median Filter for noise removal in digital images, IEEE Southeast Conference , 3-6 April 2008.

[13] M.Dentino, J.McCool, and B.Widrow, Adaptive filtering in the frequency domain, Proc. IEEE, vol. 66, no. 12, 1658-1659, Dec. 1978.

[14] Florian Luisier, Thierry Blu, Brigitte Forster, and Michael Unser, Which Wavelet bases are best for Image Denoising, SPIE Proceedings, vol.5914, pp. 100-111, Sept. 17 , 2005.

[15] Roy, Vandana. Spatial and Transform Domain Filtering Method for Image De-noising: A Review, I.J.Modern Education and Computer Science. pp. 41-49, Volume 7, 2013.

[16] Wikipedia contributors. Linear regression, In Wikipedia, The Free Encyclopedia. Retrieved 14:23, March 30, 2019, from https://en.wikipedia.org/w/index.php?title=Linear_regression& oldid=889485632.

[17] P.S. Hiremath, Prema T. Akkasaligar, and Prema T. Akkasaligar, Lin- ear Regression Model for Gaussian Noise Estimation and Removal for Medical Ultrasound Images, International Journal of Computer Appli- cations (0975 8887) vol. 50, no.3, July 2012.

[18] Dinh Hoan Trinh, Marie Luong, Jean-Marie Rocchisani, Canh Duong Pham, and Francoise Dibos, Adaptive Medical Image Denoising Using Support Vector Regression, CAIP 2011.

[19] Dinh Hoan Trinh, Marie Luong, Jean-Marie Rocchisani, Canh Duong Pham, and Francoise Di- bos, Medical Image Denoising Using Kernel Ridge Regression, IEEE international conference, 2011.

[20] Remya Ravi Nair and Ramaprasad Poojary. Image Denoising using Decision Tree Based Method, SSRG International Journal of Electronics and Communication Engineering (SSRG- IJECE), Vol. 3 Issue 6 , ISSN: 2348 8549, June 2016.

[21] V. Rivas, J. Merelo, I. Rojas, G. Romero, P. Castillo, and J. Carpio, Evolving two dimensional fuzzy systems, Fuzzy Sets and Systems, Vol. 138, no. 2, pp. 381–398, 2003.

[22] Fuzzy Logic for Image Processing: Definition and Applications of a Fuzzy Image Processing Scheme(chapter title), Advanced Fuzzy Logic Technologies in Industrial Applications(book ti- tle), pp. 101-113, Springer London, 2006.

[23] Mansi Pathak, and Dr.G.R.Sinha, A Survey of Fuzzy Based Image De- noising Techniques, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), e-ISSN: 2278-2834, p-ISSN: 2278-8735. vol. 9, no. 4, ver.I , pp. 27-36, Jul - Aug. 2014.

[24] Giovanni Palma, Isabelle Bloch, Serge Muller, and Razvan Iordache, Fuzzifying images using fuzzy wavelet denoising, IEEE International Conference on fuzzy systems, Aug. 2009.

[25] Monika Sharma and SarabjitKaur, Fuzzy Logic Based Adaptive Image Denoising, International Journal Of Computer Technology & Applica- tions, vol. 4, no.3, pp. 502-507, May-June 2013.

[26] R.Pushpavalli , and G.Sivarajde, Image Enhancement Using Adaptive Neuro-Fuzzy Inference System, International Journal Of Scientific & Technology, Research volume 2, ISSUE 6, JUNE 2013.

[27] Nguyen Minh Thanh, and Mu-Song Chen," Image Denoising Using Adap- tive NeuroFuzzy System", IAENG International Journal of Applied Mathematics (IJAM), 2007.

[28] Amaninder Kaur Brar, and Vikas Wasson, Image Denoising Using Im- proved Neuro-Fuzzy Based Algorithm: A Review, International Journal of Computer Technology and Applications, vol. 5, no. 5, pp.1772-1779, April 2014.

[29] Wikipedia contributors. Peak signal-to-noise ratio. Wikipedia, The Free Encyclopedia. March 14, 2019. Available at: https://en.wikipedia.org/w/index.php?title=Peak_signalto-noise_ratio&oldid=887764757. Accessed April 2, 2019.

[30] Wikipedia contributors. Structural similarity. Wikipedia, The Free Encyclopedia. March 31, 2019. Available at: https://en.wikipedia.org/w/index.php?title=Structural_similarity&oldid=890371981. Accessed April 2, 2019.