SOFT MORPHOLOGICAL FILTER OPTIMIZATION USING A GENETIC ALGORITHM FOR NOISE ELIMINATION

by
Türker Erçal

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Engineering

Yeditepe University
2010

SOFT MORPHOLOGICAL FILTER OPTIMIZATION USING A GENETIC ALGORITHM FOR NOISE ELIMINATION

APPROVED BY:

Assist.Prof.Dr. Ender Özcan ……………………………………………
(Supervisor)

Assist.Prof.Dr. Esin Onbaşıoğlu ……………………………………………

Assoc.Prof.Dr. Cem Ünsalan ……………………………………………

DATE OF APPROVAL: …./…./….

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assist.Prof.Dr. Ender ÖZCAN for encouraging me from the beginning, his guidance and making this project a reality by giving new ideas always. I also would like to thank to my family for their endless support and love.

ABSTRACT

SOFT MORPHOLOGICAL FILTER OPTIMIZATION USING A GENETIC ALGORITHM FOR NOISE ELIMINATION

Digital image quality is an important issue in almost all image processing applications. Images can get corrupted for many reasons. Many different approaches have been proposed for restoring the image quality depending on the nature of the degradation. One of the most common problems that cause such degradation is impulse noise. Impulse noise causes extremely dark and bright specks spread over the image to emerge. In general, well known median filters are preferred for eliminating impulse noise. Soft morphological filters are recently introduced and have been in use for many purposes. In this study, a search is performed over soft morphological filters using a genetic algorithm as a supervised learning mechanism to obtain the best filter for eliminating impulse noise. The experiments yielded a detail preserving multi-stage morphological filter. The performances of the proposed filter and median filter are compared over a set of benchmark problem instances based on different criteria. Additionally, well-known filters in the literature are also compared to the proposed filter. The results indicate the proposed filter outperforms the median filter and gets very good result among the best filters known in the literature. As a last experiment, this best filter which is obtained using images with impulsive noise is tested over images with Gaussian noise. The experiments show that the proposed filter also generates significantly better results when compared to the median filter for eliminating Gaussian noise.

ÖZET

GENETİK ALGORİTMALAR KULLANILARAK, RESİMLER ÜZERİNDEKİ GÜRÜLTÜYÜ AZALTMA AMAÇLI BİR SOFT MORFOLOJİK FİLTRE OPTİMİZASYONU

Sayısal görüntü kalitesi, neredeyse bütün görüntü işleme uygulamalarında önemli bir sorundur. Resimler, pek çok nedenden dolayı bozulabilirler. Görüntü kalitesini onarmak için; bozulmanın niteliğine gore, pek çok değişik yaklaşım önerilmiştir. Bozulmaya yol açan en genel problem, "impulse noise" olarak isimlendirilen, gürültü türüdür. "Impulse noise", görüntünün üzerine dağılmış, aşırı derecede parlak ve karanlık noktalar oluşmasına sebep olur. Genelde, iyi bilinen median filtreleri, "impulse noise" giderimi için tercih edilirler. Hafif morfolojik filtreler, yakın zamanda ortaya çıkmışlardır ve pek çok çeşitli amaç için kullanılmaktadırlar. Bu çalışmada, "impulse noise" giderimi için, en uygun filtreyi bulma amaçlı olarak; kontrol edilen bir öğrenme yöntemi olarak, bir genetik algoritma kullanılarak, hafif morfolojik filtreler üzerinde bir arama yapılmıştır. Deneyler sonucunda, detayları da koruyan, çok bölümlü bir morfolojik filter elde edildi. Önerilen filtre ile median filtrelerinin sonuçları; değişik kriterlere gore, bazı örnek test problemleri üzerinde karşılaştırılmıştır. Ek olarak, literatürdeki bilinen filtreler ile, önerilen filtre karşılaştırılmıştır. Sonuçlar, önerilen filtrenin, median filtreden daha iyi sonuçlar aldığını ve literatürdeki iyi bilinen filtreler arasında da gayet iyi sonuçlar aldığı gösteriyor. Son bir deney olarak, "impulsive noise" kullanılarak elde edilmiş en iyi filtre, Gaussian Noise eklenmiş resimler üzerinde test edildi. Bu deneyler; önerilen filtrenin, Gaussian Noise filtrelemede de, median filtreden önemli derecede daha iyi sonuçlar aldığını göstermiş oldu.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

Noise reduction, also known as noise smoothing problem is an essential concern in image processing. There are many causes of noise in digital images. For example, transmitting or scanning an image can boost up the noise due to electrical interference from the devices used. There are different types of noises dealt with in image processing such as Gaussian noise, periodic noise or impulsive noise. Many researches have proposed a variety of approaches for the removal of such noises. Impulsive noise is often encountered during image transmission process [1] or malfunctioning in camera sensors or faulty memory locations in hardware [2]. For example, modeling of the Impulsive Noise is also studied in a paper by representing it as a discontinuous Markov process [3]. There are two common types of impulse noise; the salt-and-pepper noise and the random-valued noise. In an image which is corrupted by random valued noise; the noisy pixels can take any random value, but for an image with salt-and-pepper noise; the noisy pixels can take only the maximum and the minimum values in the dynamic range. In this study, impulsive noise elimination problem that requires restoration of the image quality which is reduced by white and black (salt & pepper) spots spread over the images is tackled. There are some previous studies on this topic using different search methods. In most of the studies, the search mechanisms are used as a trainer to optimize the filter, the candidate filters are applied on a corrupted image; to calculate the fitnesses of the candidates, and the original non-corrupted image is compared with the filtered corrupted images. So, by minimizing the difference in images, it is expected that an optimized filter will be generated after some generations.

At the beginning, 1 stage standard morphological filters are optimized for noise filtering. Standard Genetic Algorithms (GAs) are used in the design of standard morphological filters [4]. New adaptive morphological operators are proposed using GAs [5]. Adaptive immune algorithms are used to optimize standard morphological filters [6]. Also, parallel GAs are used for optimizing morphological filters [7].

After soft morphological filters are found, soft flat morphological filters are optimized for the same purpose. Some researchers expanded their previous researches into different classes of morphological filters [8]. Some studies are done on breakdown probability constraints and optimal soft morphology [9, 10]. Also, different optimization methods such as simulated annealing, genetic algorithms and tabu search are studied and compared for optimization of soft morphological filters [11, 12, 13]. Genetic algorithms are also used for optimizing stack filters based on minimum mean absolute error (MAE) criterion [14].

The recent studies are provided on different sizes or multiple stages of optimized morphological filters or filters with 3-d SEs for filtering of video sequences. A two-stage standard morphological filter is generated using GAs for noise elimination in [15]. Soft morphological filters are also used in a study for the removal of noise from CCTV Footage [16]. Multi-dimensional soft morphological filters are optimized using GAs for archive film material restoration [17, 18].

For noise removal problem, direct mathematical methods are also used some studies. To remove fixed & random valued impulsive noise on gray-scale images, edge information of the image is estimated to be used as prior information in order to apply different filters adaptively [19]. For removing random-valued impulsive noise on gray-scale images, an impulse detector is used and corrupted pixels are replaced using a median-type filter [20]. For impulsive noise removal on gray-scale images, difference-type noise detector is used and the corrupted pixels are processed by differentiating the cost-function, and a criterion called UQI (Universal Quality Index) is used for testing purposes [21]. A statistical method which does not require training is used for impulsive noise removal [22]. To remove fixed & random valued impulsive noise on gray-scale images, second order difference analysis is used for noise detection and cancellation [23]. A global-local noise detector is proposed and an adaptive median filter is applied on the corrupted pixels in this paper [24]. An adaptive noise detection algorithm and a non-linear low-pass filter are used to remove high-level impulsive noise on gray-scale images [25]. An adaptive vector median filtering method is studied to take the advantage of the optimal filtering situation and the robust order-statistic theory [26]. A two output non-linear filter is studied which is based on the subsequent activation of two recursive filtering algorithms that operate on different subsets of input data [27]. A modification of the median-filter using a noise estimation mask is presented in this paper [28]. A technique called triangular interpolants is used; this restoration technique is applied on the detected corrupted pixels on the gray-scale images [29]. A variation of the non-linear peak-and-valley filter is used for removing impulsive noise [30]. A statistical method called Jarque-Bera is used to find the corrupted pixels and the standard median filter is used to replace those pixels [31]. Statistical impulse detection and a nonlinear filter are used for impulsive noise removal; the non-linear filter uses an adaptive-network-based fuzzy inference system [32]. In another paper; to improve the impulsive noise removal capability of the switching median filter, they modified it by adding one more noise detector. The detection mechanism is based on the rank order arrangement of the pixels in the sliding window [33]. To remove random-valued impulsive noise, a two-phase median filter based iterative method is used in this study [34].

For the same purpose, neural networks and fuzzy methods are also used. To remove impulsive noise on color images, neural network and fuzzy logic are used. An impulse detector detects the corrupted pixels and the network generates a new pixel to replace it [35]. For removing impulsive noise on gray-scale images; and adaptive fuzzy neural network is used with the mean-squared-error (MSE) function used as the test criteria [36]. A hybrid filter which combines median filter and a neuro-fuzzy inference system is studied; the system is trained using computer generated images [37]. Fuzzy noise detection and fuzzy filtering is used to remove random valued impulsive noise on gray-scale images [38]. A trained neural network is used to detect the corrupted pixels and a recursive median filter is used on those pixels [39]. To improve the performances of impulse noise filters, a neuro fuzzy system is trained using a computer generated image, and it is applied to the output of an impulse filter [40]. And in another study; a neuro fuzzy system is trained using a computer generated image, and it is applied on the corrupted image to removing impulsive noise on gray-scale images [41]. An adaptive neural-network using unsupervised learning is studied [42]. Adaptive neuro-fuzzy methods are used for impulsive noise suppression from highly distorted images [43, 44]. To detect impulsive noise on images, a neuro-fuzzy inference system is used in this study and median filter is applied to the noises found on those images [45]. Again, for detection of the impulsive noise on images a feed forward neural network is used and a modified version of the arithmetic mean filter is used to remove the detected impulsive noises. Another difference of this study is the search metrics; they used False Alarm Ratio (FAR), Missed Noise (MN) pixels and Falsely Detected Noise (FDN) pixels [46]. To detect impulsive noise in colour images; a novel fuzzy detector based on a fuzzy metric is used. The fuzzy detector is inspired on the recent rank-ordered differences (ROD) statistic [47].

In this thesis, a Genetic Algorithm (GA) for optimizing a multi-stage non-flat soft morphological filter is presented. The aim is to eliminate salt & pepper noise on images while preserving the quality as much as possible. Different objectives are combined in a fitness function and an appropriate representation scheme is designed to cover a broad range of filter parameters.

The proposed filter is compared with the Median Filter and with table of results given in this paper; "A new method for impulsive noise suppression from highly distorted images by using Anfis" [43] which presents a filter called "Anfis-based impulsive noise removing filter (AIF). And, that table of compared results consists of these papers & filters: SMF [48], Iterative Median Filter (IMF) [49] and the complex structured impulsive noise removal filters: Progressive Switching Median Filter (PSM) [49], Signal Dependent Rank Order Mean Filter (SDROM) [50], Two-state Recursive Signal Dependent Rank Order Mean Filter (SDROMR) [50], Impulse Rejecting Filter (IRF) [51], Non-Recursive Adaptive-Center Weighted Median Filter (ACWM) [52] Recursive Adaptive-Center Weighted Median Filter (ACWMR) [52], Center Weighted Median Filter (CWM) [53], Yüksel's Anfis based filter (YÜKSEL) [41], Russo's fuzzy filter (RUSSO) [54] and Histogram Based adaptive fuzzy filter (HAF) [55].

In the next section, background information about soft mathematical morphology and genetic algorithms is given. In section 3, the representation scheme and other relevant components of the genetic algorithms are explained in detail. In section 4, the experimental settings used to obtain the best filter are explained and performance comparison of the generated filter and median filter is provided over 10 different images. Also, the filters generated for different levels of noise are compared with the latest filters presented in the literature and the improvement of the GA and the fitness function is shown visually and mathematically. Finally, conclusions are discussed in Section 5.

2. BACKGROUND

2.1. SOFT MATHEMATICAL MORPHOLOGY

Mathematical morphology was first introduced as an image processing methodology for binary images [56]. The basic operators used in mathematical morphology are ‘erosion’ and ‘dilation’ operators that accept the image itself and a ‘structuring element’ (SE), also referred to as ‘kernel’ as input. For more details on mathematical morphology, readers can refer to [57, 58]. Dilation causes regions of foreground pixels grow in size and holes within those areas shrink, while erosion has a reverse effect as illustrated in Figure 2.1.

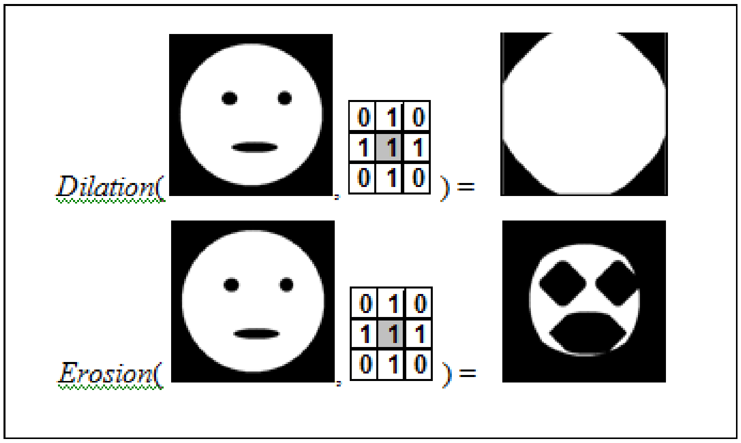
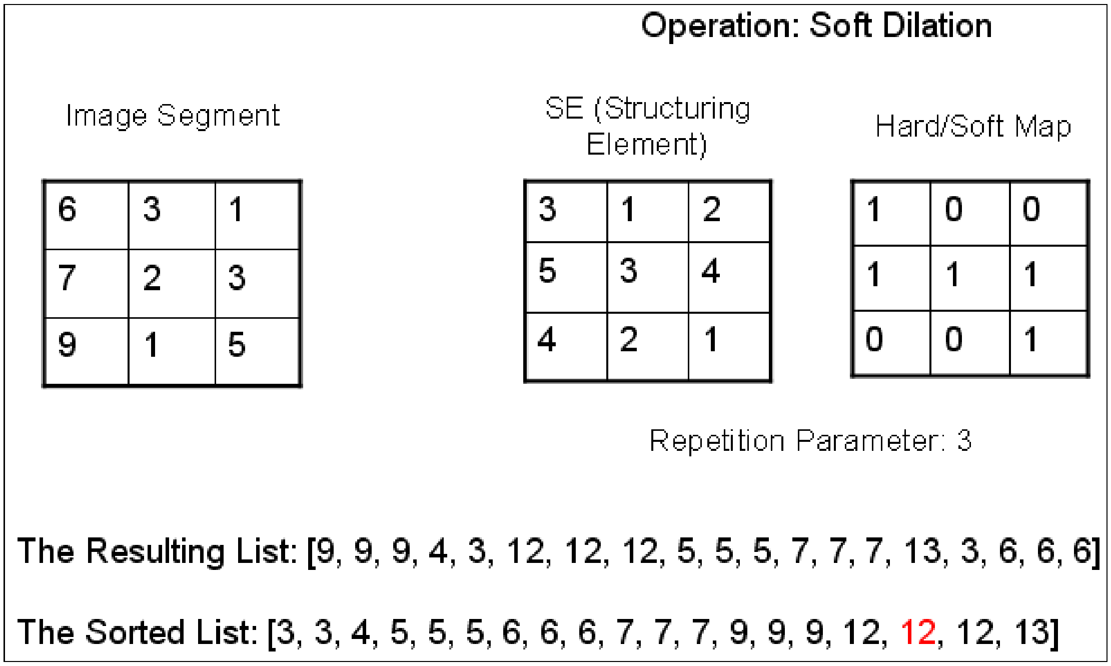


Figure 2.1. Repetitive dilation and erosion of a given binary image by a given structuring element having the origin in the middle for 40 iterations (1:on, 0:off)

Later, in gray-scale morphology erosion and dilation operators are replaced with ‘minimum’ and ‘maximum’ operators as the fundamental morphological operators, respectively. There are also ‘opening’ and ‘closing’ operators which requires application of dilation and erosion using the same structuring element in a specified order. Up to now, mathematical morphology has been used in many different image processing applications, ranging from noise suppression, feature extraction to object recognition.

Soft mathematical morphology introduced by Koskinen, Astola, & Neuvo [59] in 1991. In this approach, weighted order statistics is used instead of the minimum or maximum. The main difference from the standard morphology is the division of the structuring element into two parts; 'hard centre' and 'soft boundary'. The numbers in the hard centre part has weights greater than one, which is set by a parameter called 'rank' or 'repetition parameter'. The numbers in the soft boundary part of the SE has weights equal to one. An example soft dilation operation is illustrated in Figure 2.2.



Operation: Soft Dilation

Image Segment

6	3	1
7	2	3
9	1	5

SE (Structuring Element)

3	1	2
5	3	4
4	2	1

Hard/Soft Map

1	0	0
1	1	1
0	0	1

Repetition Parameter: 3

The Resulting List: [9, 9, 9, 4, 3, 12, 12, 12, 5, 5, 5, 7, 7, 7, 13, 3, 6, 6, 6]

The Sorted List: [3, 3, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 9, 9, 9, 12, 12, 12, 13]

Figure 2.2. An example soft dilation operation

Given a structural element B, it is divided into two subsets and, B is divided into two subsets: the hard centre structural element A and the soft structural element $B \setminus A$, where $A, B \subseteq Z^2$, $A \subseteq B \neq \varnothing$ and $\setminus$ denotes the set difference. Let $\oplus$ and $\ominus$ represent 'soft dilation' and 'soft erosion' operations. Soft dilation and soft erosion of an image f with rank-order i are defined as:

$$\begin{aligned}\oplus_{B,A,i}(f) = \max{}^{(i)}\{i\Diamond(f(z-\varepsilon)+A(\varepsilon)) \mid \varepsilon \in F_A\} \cup \\ \{f(z-\delta)+B(\delta) \mid \delta \in F_{B\setminus A}\}\end{aligned} \qquad (2.1)$$

$$\Theta_{B,A,i}(f) = \min^{(i)}\{i\lozenge(f(z+\varepsilon) - A(\varepsilon)) \mid \varepsilon \in F_A\} \cup \\ \{f(z+\delta) - B(\delta) \mid \delta \in F_{B\setminus A}\} \tag{2.2}$$

Where max(i) and min(i) denote the ith largest and smallest value in the set respectively; ◊ is the repetition operator and $i\lozenge f(v) = \{f(v), f(v)..., f(v)\}$ (i times); FA and FB\A represent the field of definition of A and $B \setminus A$, respectively. Consequently, soft opening (Λ) and soft closing (Δ) of an image f are defined as:

$$\Lambda_{B,A,i}(f) = \oplus_{B,A,i}(\ominus_{B,A,i}(f)) \tag{2.3}$$

$$\ominus_{B,A,i}\ \Delta_{B,A,i}(f) = (\oplus_{B,A,i}(f))(4) \tag{2.4}$$

It has been shown that soft morphological operations are more robust in noisy conditions and are less sensitive to additive noise and to small variations in object shape [60].

2.2. GENETIC ALGORITHMS

A genetic algorithm (GA) is a nature inspired and population based meta-heuristic used in search and optimization [61]. GAs have proven success in solving different classes of complex problems [62, 63, 64, 65]. The approach aims to improve a set of randomly generated initial candidate solutions, referred to as 'population' through Darwinian evolution as shown in the below list. A 'chromosome' denotes a candidate solution that is made up of 'genes', where each gene receives a value from a set of 'alleles'. For example, using binary representation, '00010' might represent a candidate solution for a problem requiring 5 genes. Other types of representation schemes are also allowed.

```
1.  Generate initial population of size N, $P_t(N)|_{t=0}$
2.  Evaluate chromosomes, $P_t(N)|_{t=0}$
3.  Repeat
4.  Select mates $M_t(K)$
5.  Apply Crossover, Obtain Offspring
6.  Mutate Offspring, $O_t(K)$
7.  Evaluate chromosomes, $O_t(K)$
8.  Replace current population, form next population
    // $P_{t+1}(N)$: select N from (N+K) chromosomes
9.  Until Termination Criteria are satisfied
10. Return the best chromosome
```

In an evolutionary cycle, a set of genetic operators, namely; ‘crossover’, ‘mutation’ and ‘replacement’ are applied to the chromosomes in that order. Good building blocks, possibly some part of an optimal solution are kept within the population, while the poor ones are eliminated based on a ‘fitness function’ denoting the quality of a given candidate solution. One point crossover (1PTX) is the traditional crossover used in GAs. Two mates are randomly selected within the population favoring better chromosomes having better fitness values. Then the genetic material is swapped at a randomly chosen locus. For example, assuming that ‘00010’ and ‘01001’ are selected as mates that will go through 1PTX, if the locus is randomly determined as 2, then two new chromosomes (offspring) are generated as follows: ‘00|010’ × ‘11|011’ → ‘00011’, ‘11010’. Next, mutation is applied to the offspring by perturbing the allele value at each locus to a different value in the allele set with a given probability. In binary representation, mutation is a bit-flip operation. For example, assuming a mutation probability of 1/5, the offspring ‘00011’ might change into ‘01010’.

3. GENETIC ALGORITHMS FOR GENERATING AN OPTIMAL FILTER

The need to search and try different combinations of operations, structuring elements and parameters expands the search space dramatically (2.14 x 1081) and makes the search for the optimal soft morphological filter well suited for GAs.

In this study GAs are used as a supervised learning mechanism to generate a filter for impulsive noise removal. The GA optimizes the parameters of the soft morphological filter by comparing the results of the filter over a set of training images.

3.1. IMPULSIVE NOISE

In this study, salt and pepper noise is studied. The usual and effective method for suppressing this kind of noise is generally by the use of median filter. Also, in this study, it is accepted that the noise is distributed on each pixel over the image with equal probability, and the generated filter will be best suitable for a trained noise level, but it can handle all different random distributions of that noise level.

3.2. REPRESENTATION

Each candidate solution coded by a chromosome represents a different filter. In this representation scheme, a soft morphological filter that has up to four stages can be encoded. Each of these four stages can have different morphological operations, parameters and structuring elements.

It is shown in a previous study [15] that having multiple stages gives better results compared to a one-staged morphological filter. The representation scheme used in this study can give every type of filter, so that we have a broader range of filters to choose.

In the first four genes of the chromosome, the type of the morphological operations for each stage are coded (0-No Operator, 1-Erosion, 2-Dilation, 3-Opening, 4-Closing).

By using a ‘No Operator’ type, the corresponding stage can be eliminated, or by using one of the other operations, the operator can be selected for the corresponding stage.

In the next four genes, the neighborhood type is coded. In this study, 4 and 8 neighborhood types are allowed. In a 4 neighborhood type, the corners of the SE are not used, but in an 8 neighborhood type, all the nine elements in the SE are used. First of all, these are the most common ones and some prior tests done showed that, enlarging the search space by allowing all types of SE shapes does not give better results. For this part of the chromosome, zero means a 4n type neighborhood; one means an 8n type neighborhood.

The next four genes are also binary like the previous four genes. They decode the information whether the SE is symmetric or not. In a symmetric SE, the numbers are symmetrically placed in all planes (horizontally, vertically and diagonally). But, in a non-symmetric SE, all the numbers are different. For this part of the chromosome, zero means a non-symmetric SE, one means symmetric SE.

The next 36 genes represent the numbers in the structuring elements of the four stages. 3x3 sized SEs are used in this study; which means each Structuring Element has nine numbers in them. Each slot in an SE has a range of (-255, 255). In a previous work [15], this range is used and it is shown that [4, 15] expanding the range of the number in the SEs gave better results. A grayscale image has a range of (0, 255); in order to change a black pixel (255) into a white pixel -255 have to be added, and a value of 255 have to be added for the opposite. Therefore, to cover all the values of a grayscale image, this range is used.

The last two parts are the parameters only used in soft morphology. The next 36 genes represent the ‘hard’ and ‘soft’ elements in the SEs. This part of the chromosome is also binary. Each gene works for a single number in an SE. When encoding, a zero maps to a soft element, and a one maps to a hard element.

The last four genes in a chromosome represent ‘the repetition parameter’ of ‘the rank parameter’ for each stage. The range of each gene in this part is (1 – 9). Using one in this field corresponds to a standard morphological filter, other values change the operation according to the rules defined in soft mathematical morphology. The values above eight give the same results, because the SEs used in this study has nine numbers in them. Therefore, the range has a maximum value of nine.

3.3. EVALUATION FUNCTION

The GA in this study is used like a training mechanism; a filter trainer for a specific purpose which is noise elimination. The fitness calculation gives us a value that represents the difference between the filtered noisy image and the original non-noisy image. Therefore, the lower the difference, the better the filter will be.

This calculation is composed of four parts. The first two comparing criteria (mean-absolute-error and mean-squared-error) are used in almost all the studies that are done on this topic before. Shape Error criteria is a relatively new method compared to that two error types and the brightness error is a comparison method that is used in this thesis and didn’t used in any of the papers researched in this study.

The mean-absolute-error calculates the difference of the pixels at the same coordinates in two images and adds all the differences and gives the mean average of that sum. It is a very standard error criterion.

The formulas are arranged for two 8-bit gray-scale images X and its restored image Y, with pixel indices “i” and “j”, i.e., image size is of M×N pixels.

$$f_1 = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| X(i,j) - Y(i,j) \right| \tag{3.1}$$

This criterion is like the mean-absolute-error, but has a difference. In here, the squares of the differences are taken before addition. Mean-squared-error criterion strengthens the effect of differences on the result. Each one is used in previous studies and none of them has precedence over the other. So, in this study, both of them are used together.

$$f_2 = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X(i,j) - Y(i,j) \right)^2 \tag{3.2}$$

The previous criteria check pixel by pixel comparisons, but do not take the whole image or shapes into account. In image reconstruction, that is not enough to preserve the shapes or the details in the image. So, the Shape Error is incorporated into the fitness function. It is first introduced in the paper [66].

$$f_3 = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{(i',j') \in w} \left| X(i,j) - X(i',j') - (Y(i,j) - Y(i',j')) \right|^{y} \tag{3.3}$$

M, N, X and Y represents the same things in previous equations, y is set to 2, (i', j') specifies the floating windows around (i, j) with the masking element w, which is a 3x3 window in this case.

After generating filters with the first three objectives and making some tests using the resulting filters; it is seen that the filtered image has the necessary properties such as smoothed noise and preserved details. But, it has a problem that the brightness of the image is a little bit different from the original image. According to my observations, this is because of using non-flat structuring elements. The big values can change the average pixel value of the image.

This problem is tried to be handled with two different methods, one of them is the Brightness Error Criteria. The other method will be explained in the Fitness Calculation Section. The Brightness Error Criteria simply calculates the absolute value of the difference between the average pixel value of the input image and the output image. By embedding this objective into the fitness function, the search is biased into a direction where the brightness of the resulting image will be close to the original image's brightness value.

$$f_4 = \left(\frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j) \right) - \left(\frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} Y(i,j) \right) \tag{3.4}$$

The first problem solved when combining these objectives in one fitness function is the scaling problem. The scaling is important to equalize the impact of the objectives on the fitness value. To scale them, each objective is divided by the maximum possible value they can give in an extreme case.

For Mean-Squared and Mean-Absolute-Error criteria, the extreme case is a comparison of two images; that if one image has a black pixel, the other one must have a white pixel at the same location and vice versa. The example images for this case are given in Figure 3.1. Mean-Squared-Error calculates a difference of 65025; Mean-Absolute-Error calculates 255 for this case. So, the values of these objectives are divided with these values in the fitness calculation.

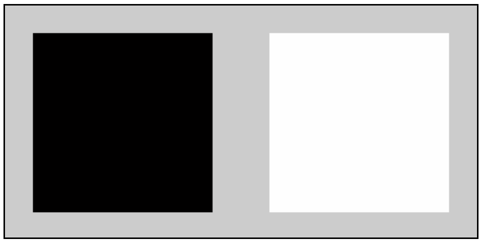


Figure 3.1. An example extreme case for MSE and MAE criteria

According to the tests done, the extreme case for the Shape Error Criteria is a case that each image has only horizontal or only vertical black lines on a white background with one pixel gap between them starting from one end and ending at the other end. But each image has the black lines where the other image has gaps (white lines). The example images for this case are given in Figure 3.2. This situation also fits the extreme case for the MSE and MAE criteria. The Shape Error Criteria calculates a difference of 1238 in a situation like this for two 100x100 images.

In this example; each image has a white background, but the first image has black lines in the even valued columns, whereas the second image has black lines in the odd valued columns. Therefore, these images are appropriate for the extreme cases of MSE, MAE and Shape Error criteria.

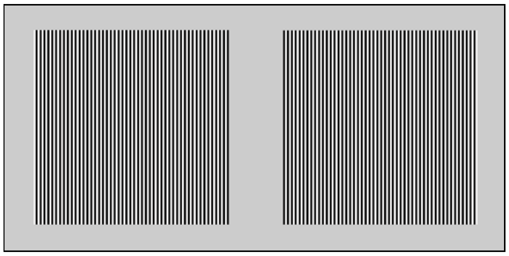


Figure 3.2. An example extreme case for the shape error criteria

The extreme case for Brightness Error Objective is a situation where one image is completely black and the other is completely white. This objective returns a value of 255 for this situation.

So the scaled fitness function looks like this:

$$f'(X,Y)=\sum_{i=1}^{4}s_i f_i \tag{3.5}$$

At the beginning of this study, a single non-noisy image with a size of 100x100 and a 100x100 20% impulsive noise added version of that image is given as an input into the GA. At each fitness calculation for each candidate filter, the filter is applied to the noisy image. The fitness value will be the difference of the original image and the resulted image according to the four objectives defined before.

But, using a single image gives a filter that is best suitable for an image with similar attributes like the training image. So, it is seen that to achieve a generalized filter, more than one image is needed for the training phase. In this study, three different 100x100 sized images are used for the fitness calculation. The fitness of each candidate filter is calculated with the same way described before, and the average value of these calculations made up the final fitness value.

$$f=\frac{\sum_{i=1}^{3}f'(X_i,Y_i)}{3} \tag{3.6}$$

Using more than a single image for the fitness calculation is the second method for solving the Brightness Error Problem. The three images are selected specially for this purpose. The average pixel value of a grayscale image ranges between (0, 255). If this range is divided into four equal parts, each part has a range of 64. Therefore, the three images used here have average pixel values of 64, 128 and 192 individually. The aim here is to adapt the filter for different brightness conditions incorporating with the brightness error criteria.

4. COMPUTATIONAL RESULTS

4.1. EXPERIMENTAL DATA AND SETTINGS

4.1.1. Training Set

The three Images used for the training of the filters are shown in the Figure 4.1. This figure only shows the images that have 20% level of noise. But for the last test which is explained in the Section 4.4, the same images are used but with noise levels ranging from 10% to 90%.

These three Images are cropped from the Flowers, Mandrill and Lena Images. Small Images are used in the training because; GA takes too much time when bigger sized Images are used. And the main difference between these Images is described at the end of the previous Section. They have different average brightness values (64, 128 and 192) which we think that will help the GA to generate a filter that is not dependent on the brightness of an image. The training Images and their source images are illustrated in Figures 4.1 – 4.5.

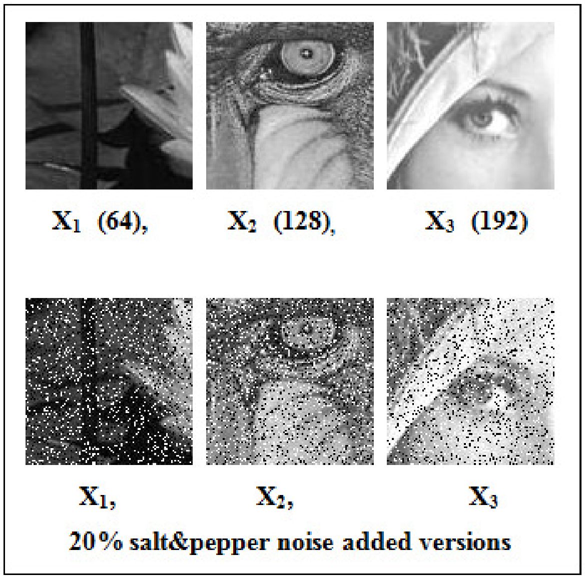


Figure 4.1. Three training images (noisy and clean versions)

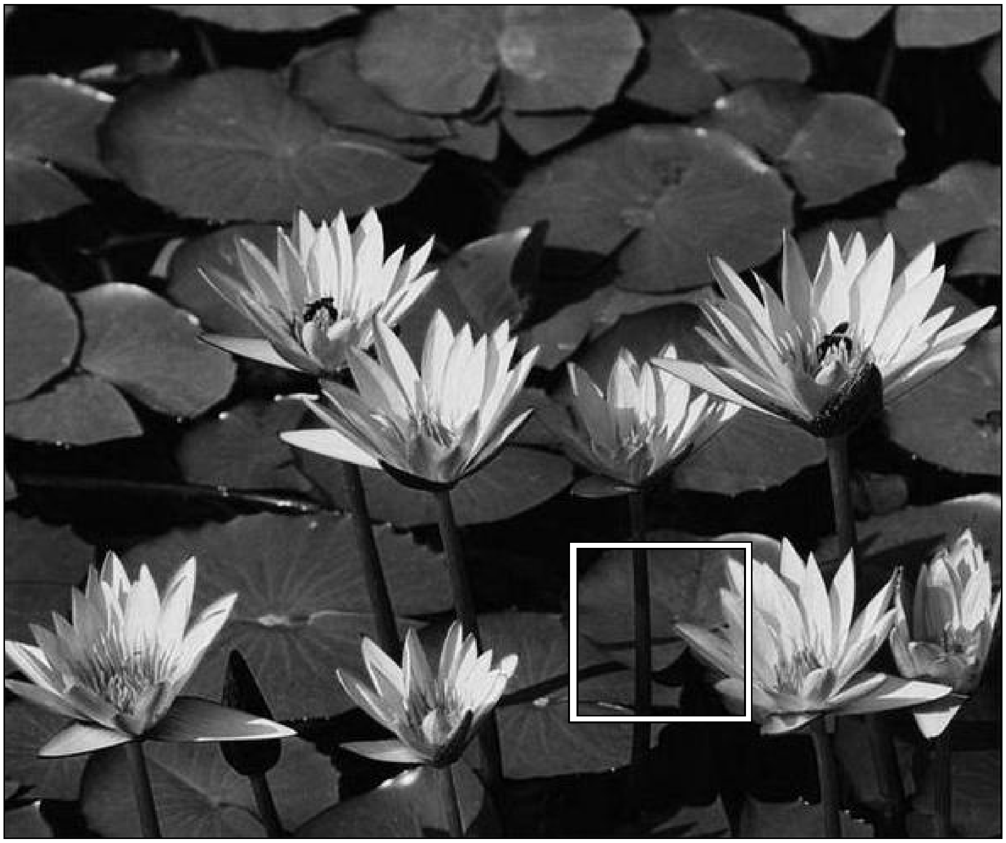


Figure 4.2. The cropped part of the Flowers image (335, 320) – (434, 419)

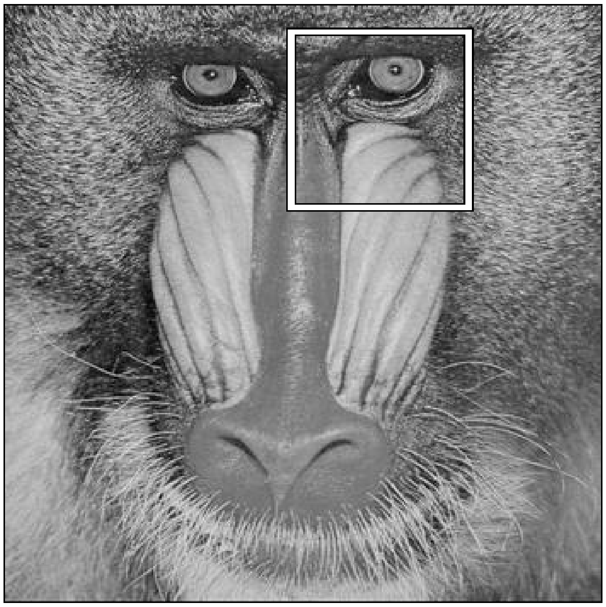


Figure 4.3. The cropped part of the Baboon image (170, 17) – (269, 116)

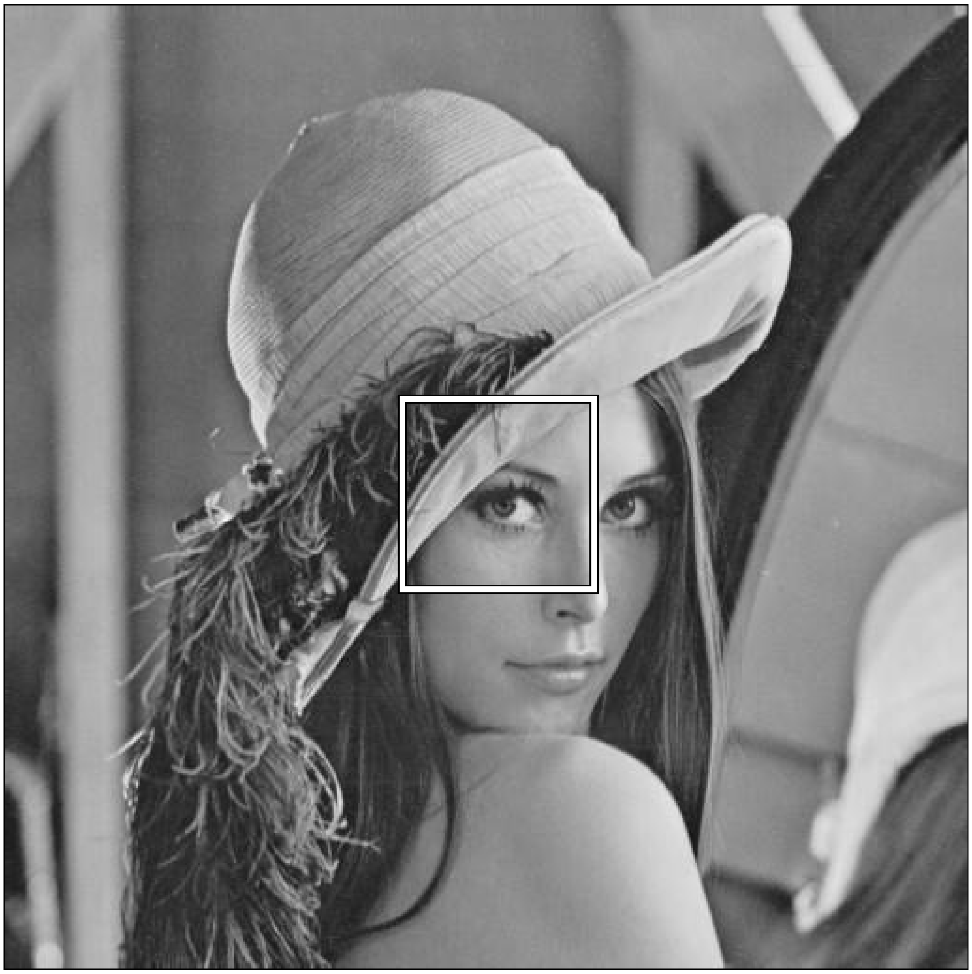


Figure 4.4. The cropped part of the Lena image (212, 213) – (311, 312)

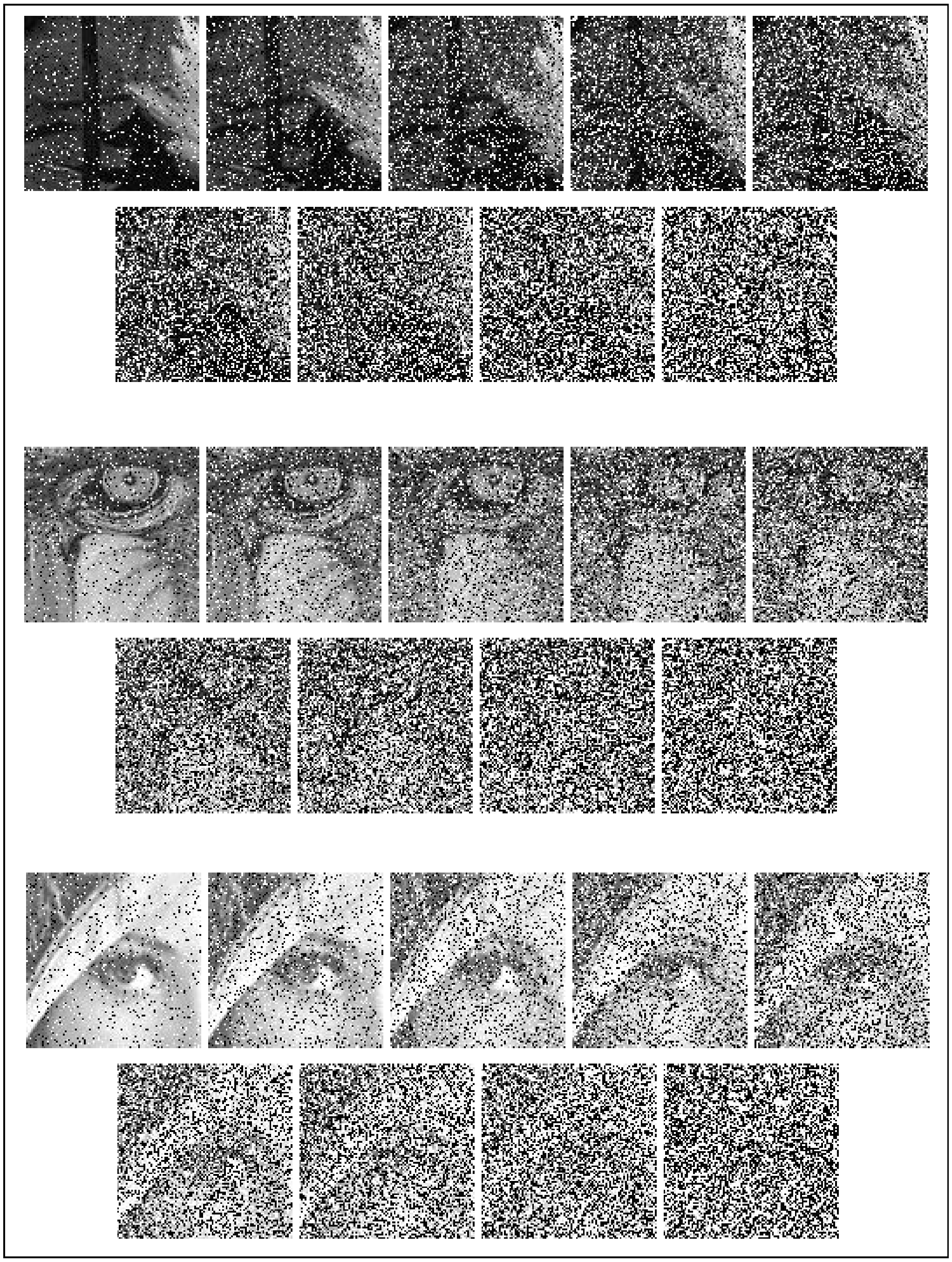


Figure 4.5. 27 (3x9) Noisy training images for all levels of noises used (10%-90%)

For the training phase; the GA parameters shown in the Table 4.1 are used. The MATLAB GA Library is used as the backbone of the GA, but the population generation operation, fitness function, cross-over and the mutation operators are written as customized functions and called from the GA Library.

Table 4.1. Genetic Algorithm Parameters

Population Size:	50
Termination Criteria:	Number of Generations (1000)
Selection Method:	Tournament Selection (Size 4)
Cross-Over Method:	Scattered Crossover
Mutation:	Simple Mutation (1 / ChromosomeLength = 1 / 88)

An Initial Population with a size of 50 individual is randomly generated according to the value ranges of each gene. To simply explain scattered cross-over; a vector with a size of 88 is formed with random values of 0 and 1. After this operation, the corresponding genes which have a value of 1 in the vector are swapped within parents. Therefore, the children are formed after this swapping operation.

To generate a final filter; the GA is run with the same parameters and with the same training images for 30 runs. At each run, a different random seed is used. After 30 runs are finished; the fitness values of the final filters for all runs are collected and the filter with the best fitness value is selected as the generated filter.

4.1.2. Test Set

In the Figure 4.6., the non-noisy Images used in the tests are shown. To fit them in a page; all of them are scaled. The original sizes are as follows:

Camera (256x256), Bridge (256x256), Peppers (512x512),
Reptile (441x331), Parrots (384x256), Airplane (512x512),
Mandril (350x350), Lena (512x512), Boat (512x512), GoldHill (512x512)

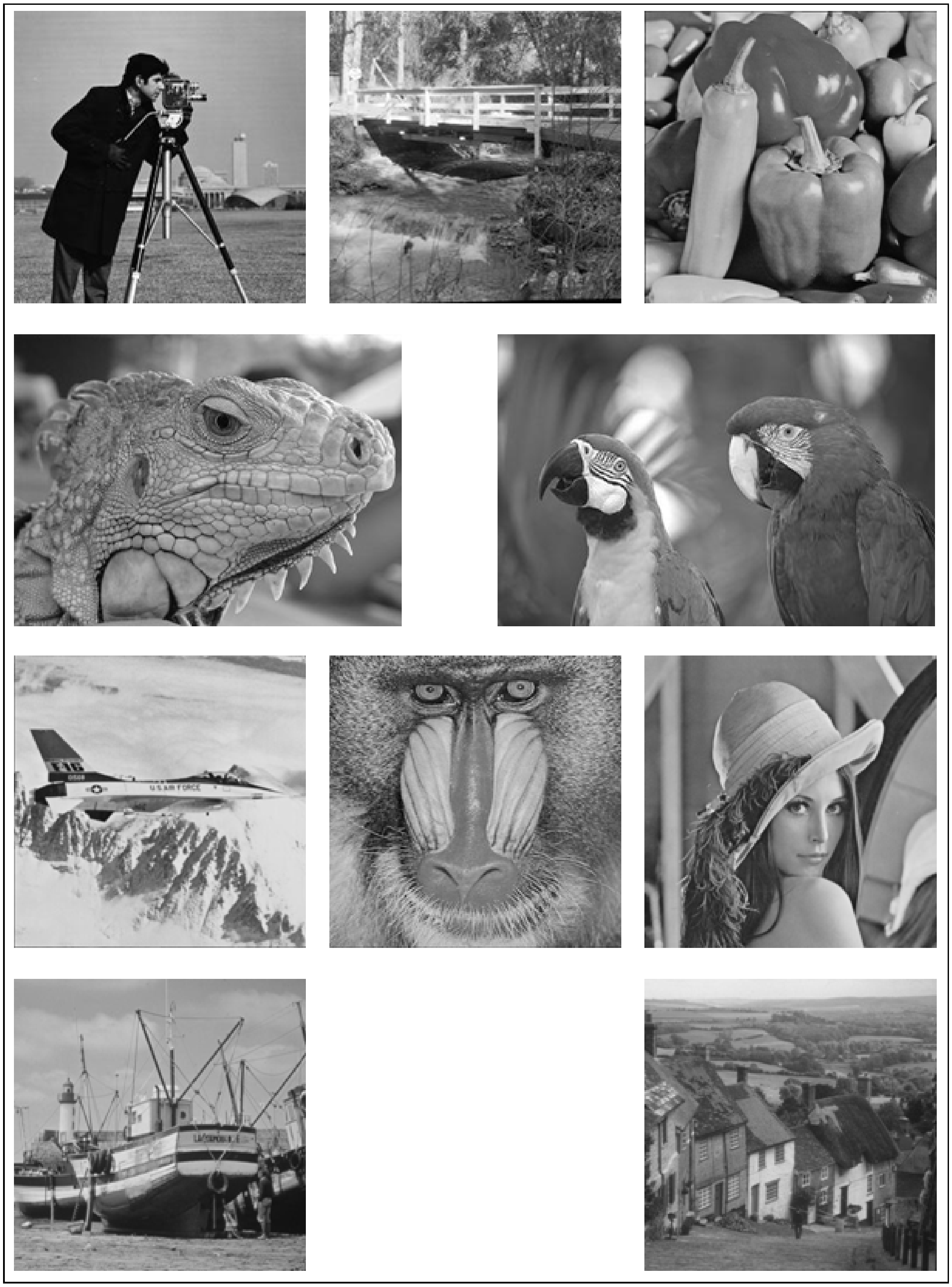


Figure 4.6. All 10 non-noisy test images

(Camera, Bridge, Peppers, Reptile, Parrots, Airplane, Mandrill, Lena, Boat, Goldhill)

4.1.3. Image Properties

In this part; the properties of both the training and test images are presented. Two types of image properties are calculated and showed here.

One is called histogram; the histogram of an image refers to a histogram of the pixel intensity values. The histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values.

The other critera is called Average Pixel Value; which is simply the average value of the sum of all the pixels of an image.

The main aim of showing these properties is to highlight the difference of the training images and test images. And it is shown that the cropped training images do not carry the same properties with the original test images as it can be seen from the histograms and average pixel values in the Table 4.2 and Figures 4.7 – 4.19.

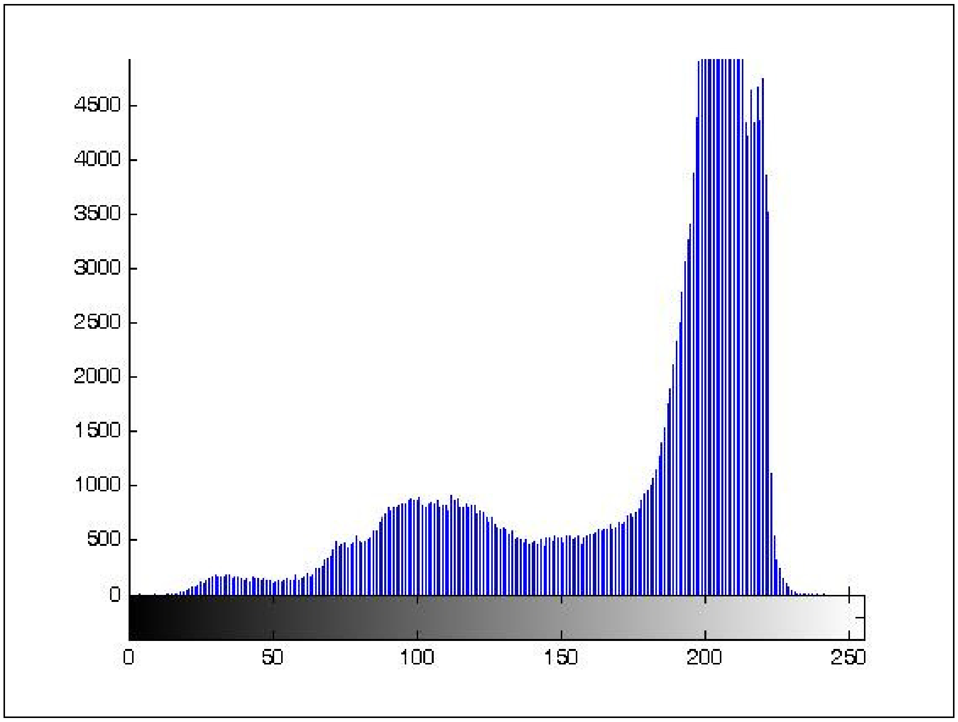


Figure 4.7. Histogram data of the Airplane image

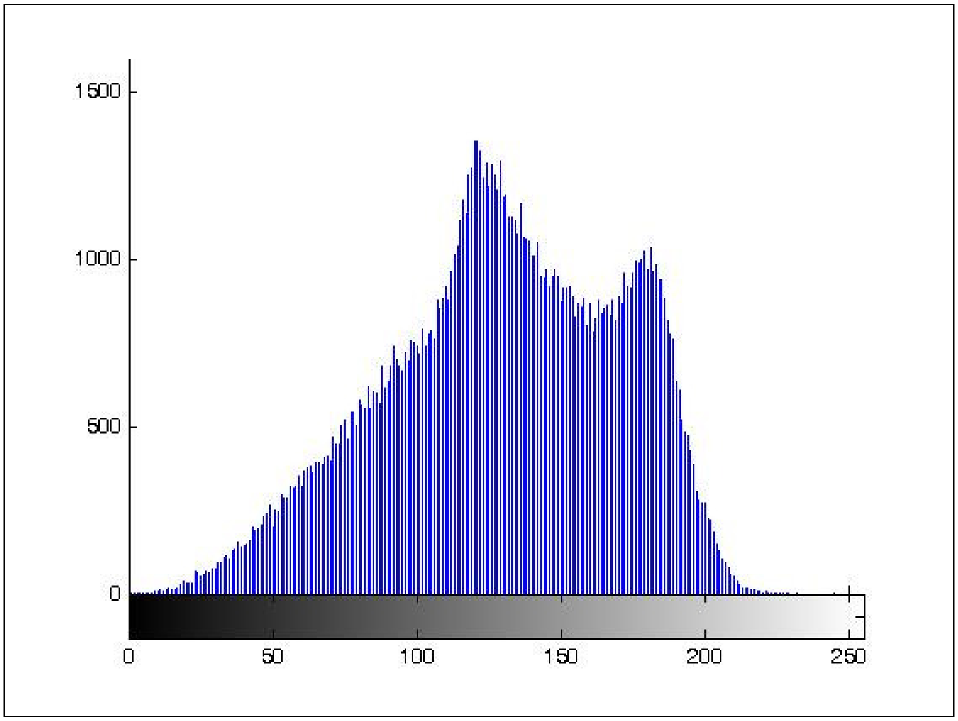


Figure 4.8. Histogram data of the Baboon image

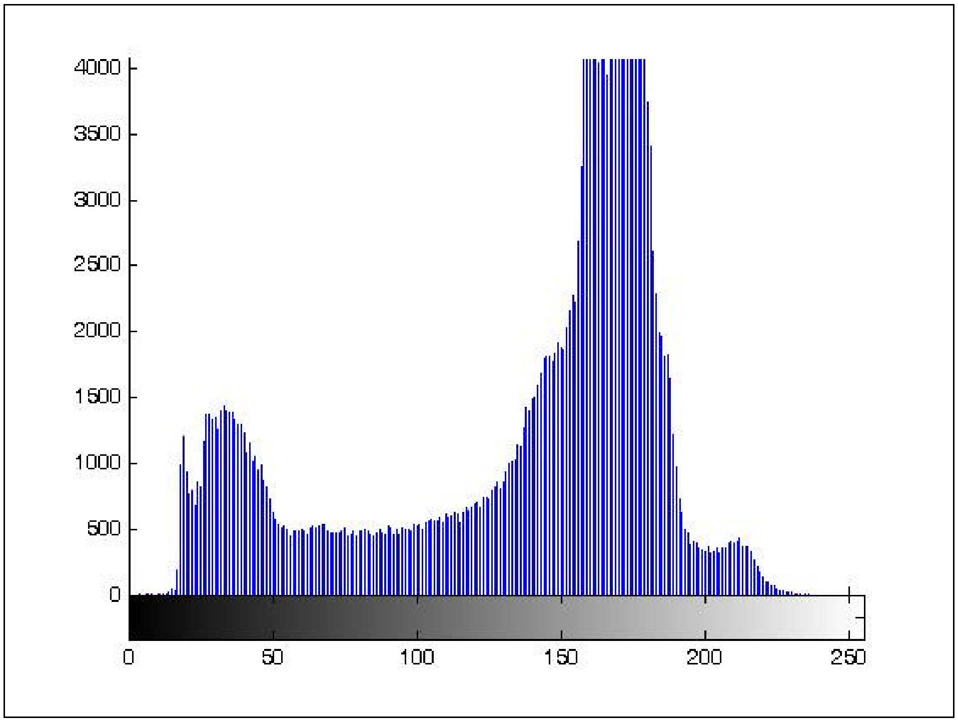


Figure 4.9. Histogram data of the Boat image

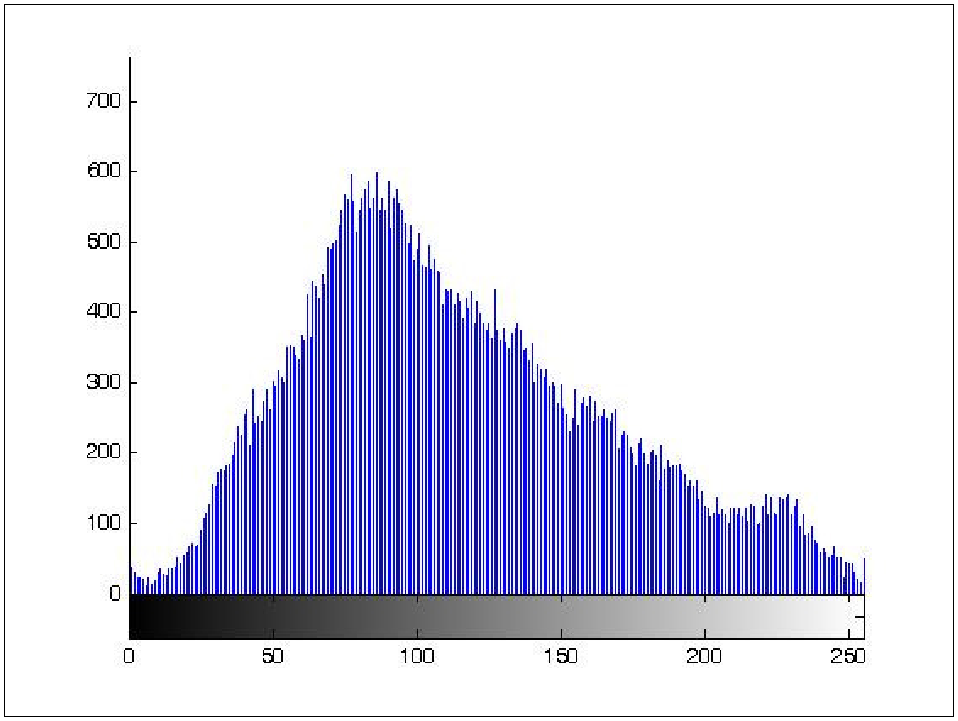


Figure 4.10. Histogram data of the Bridge image

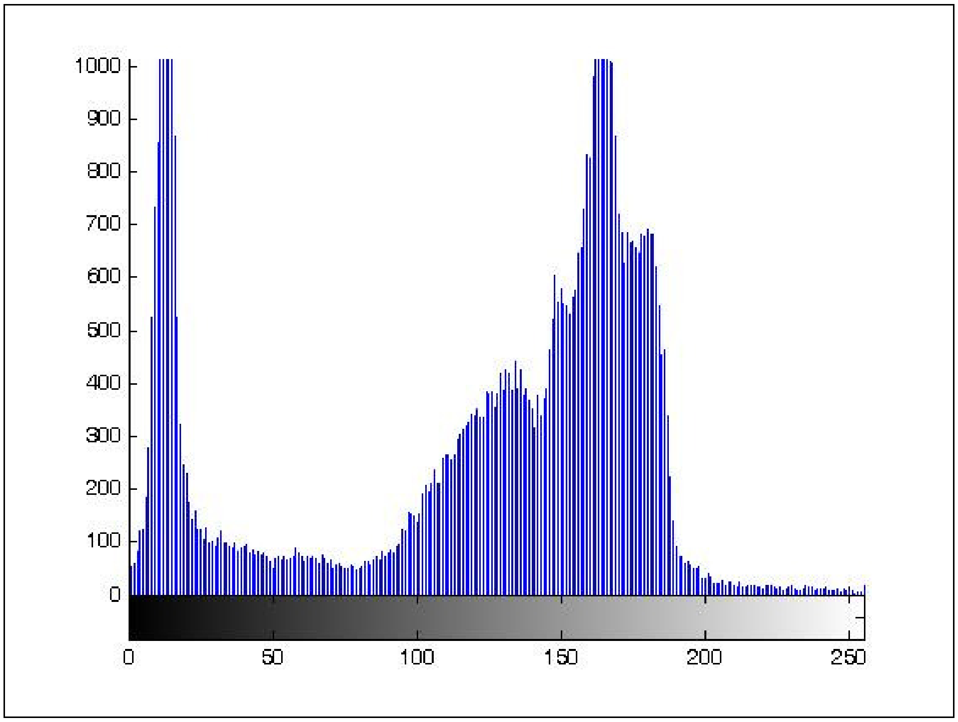


Figure 4.11. Histogram data of the Camera image

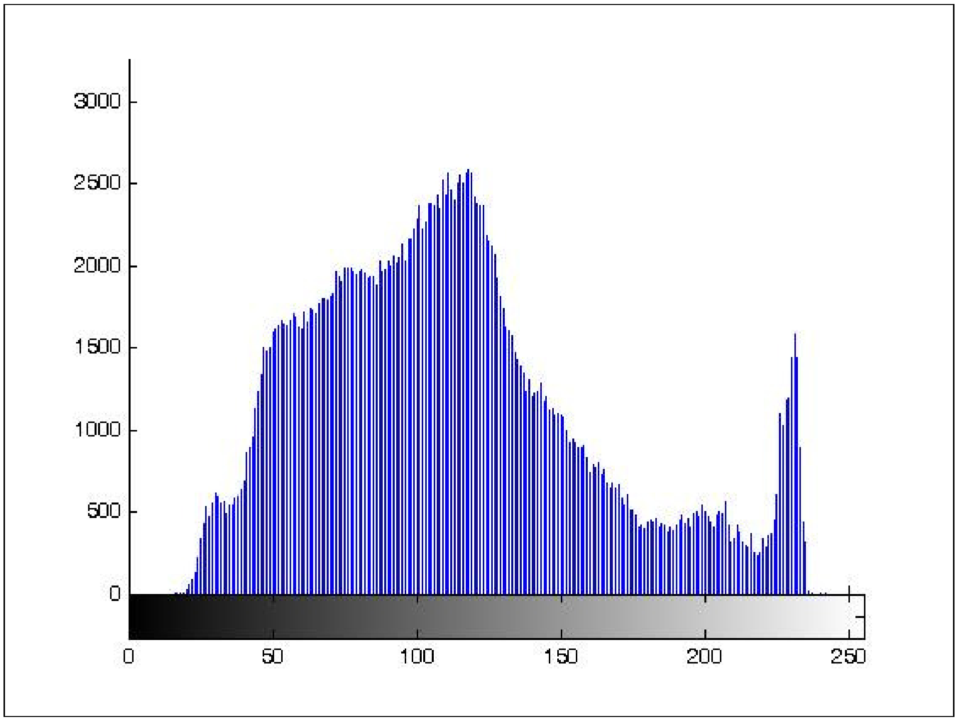


Figure 4.12. Histogram data of the Goldhill image

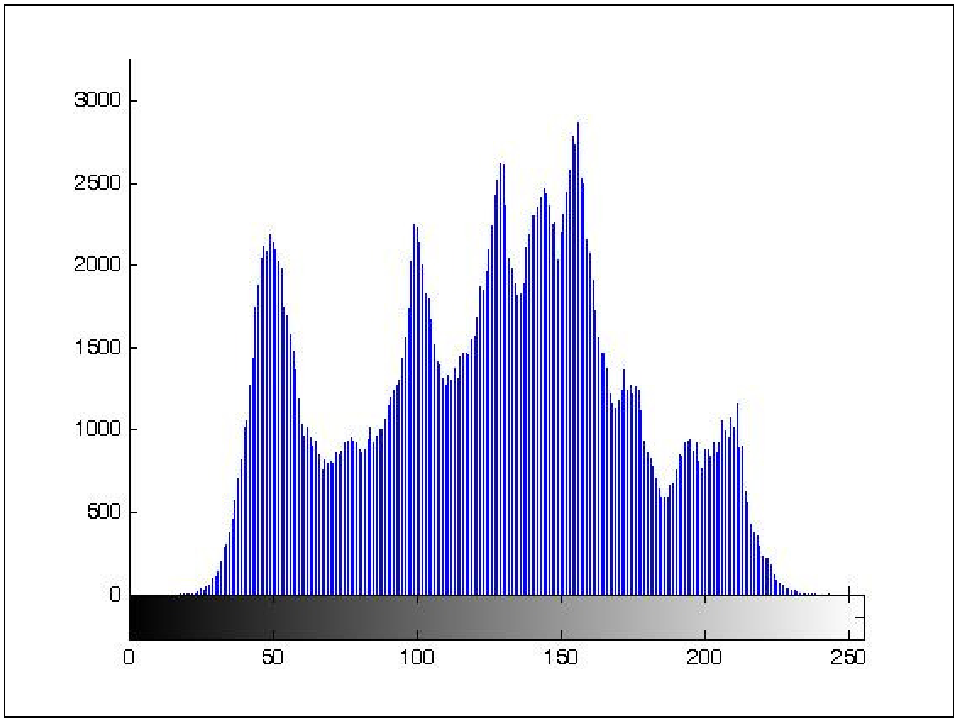


Figure 4.13. Histogram data of the Lena image

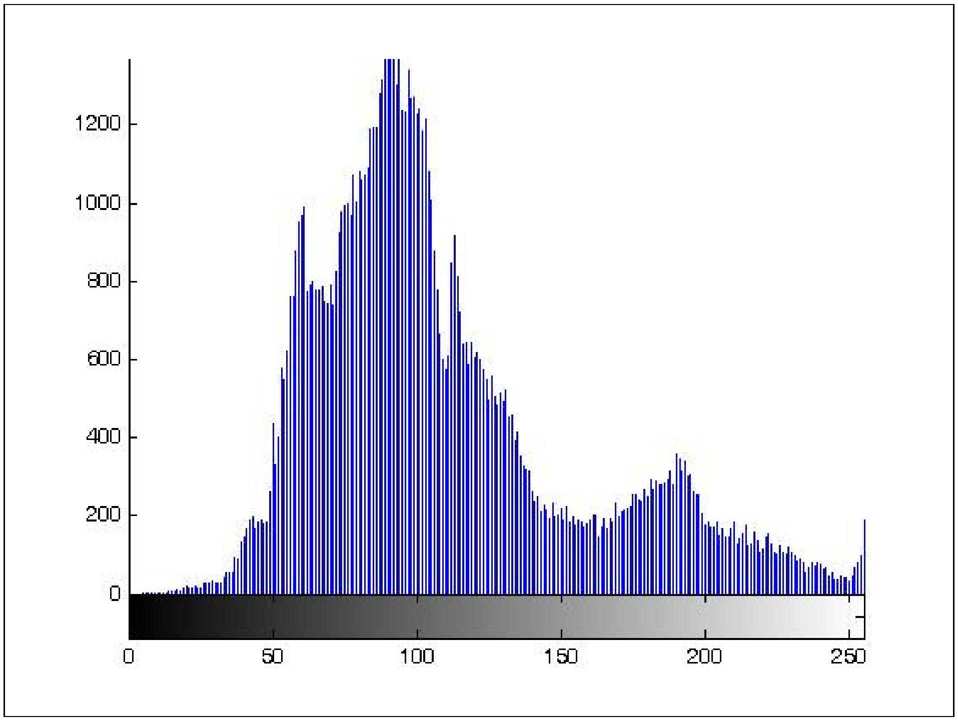


Figure 4.14. Histogram data of the Parrots image

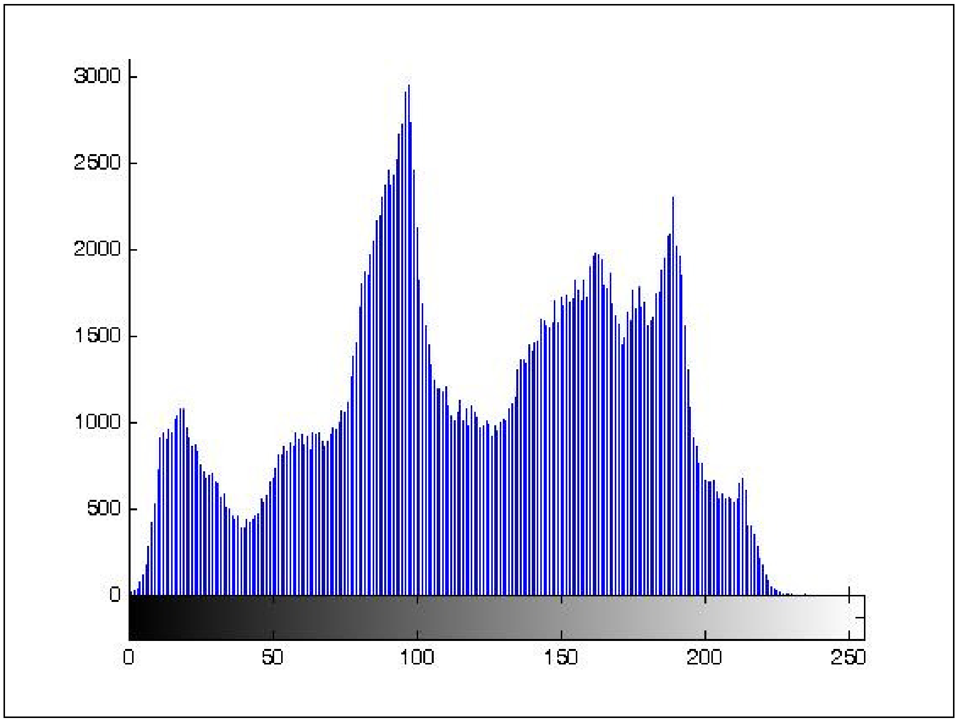


Figure 4.15. Histogram data of the Peppers image

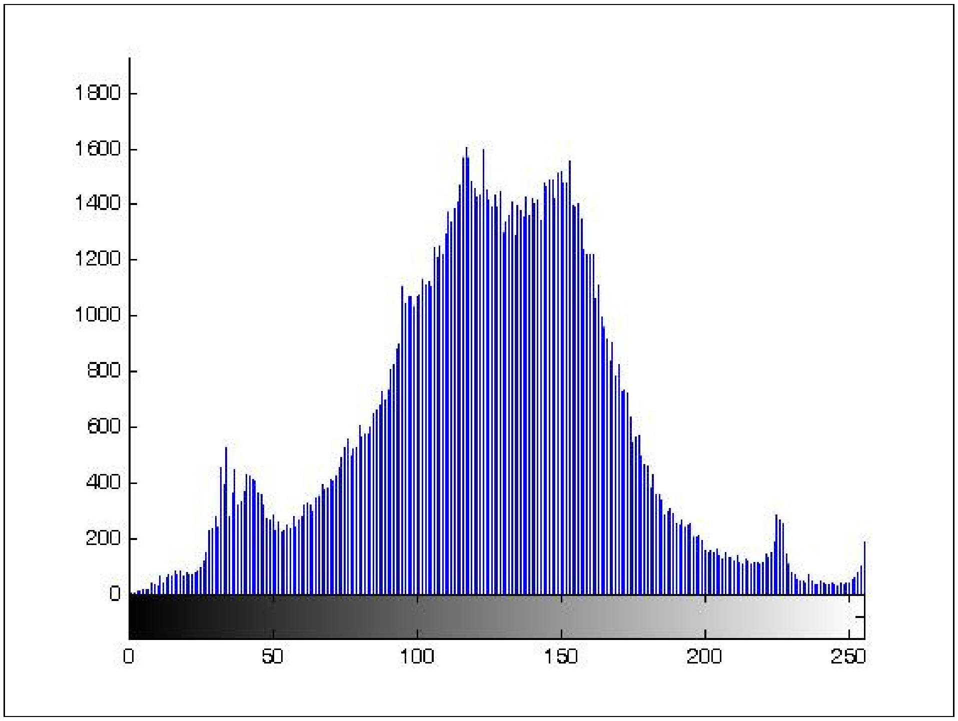


Figure 4.16. Histogram data of the Reptile image

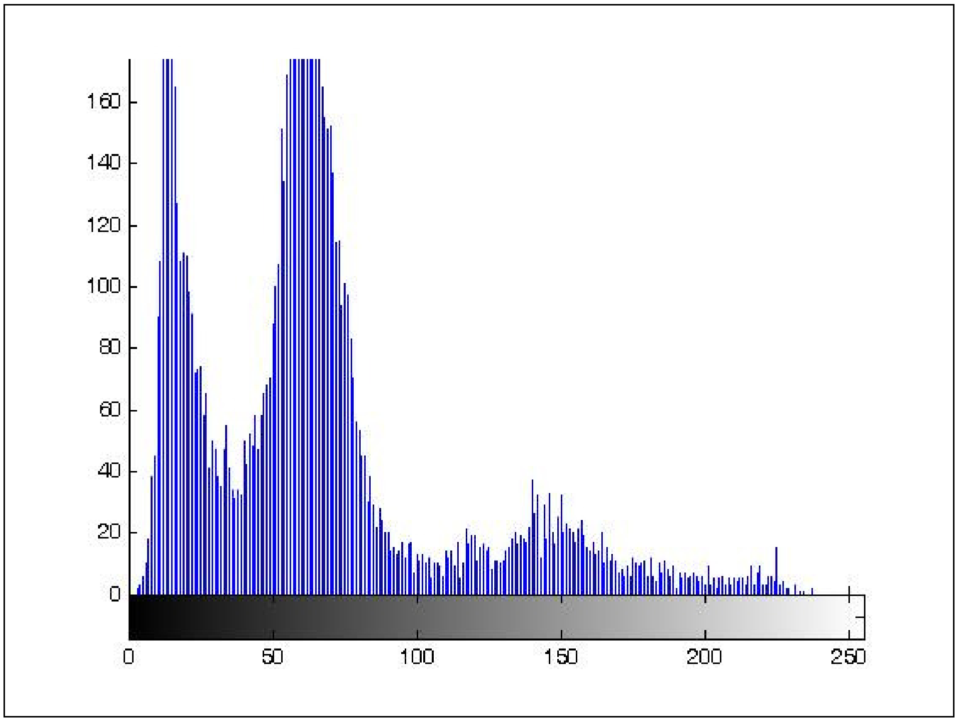


Figure 4.17. Histogram data of the X_1 training image

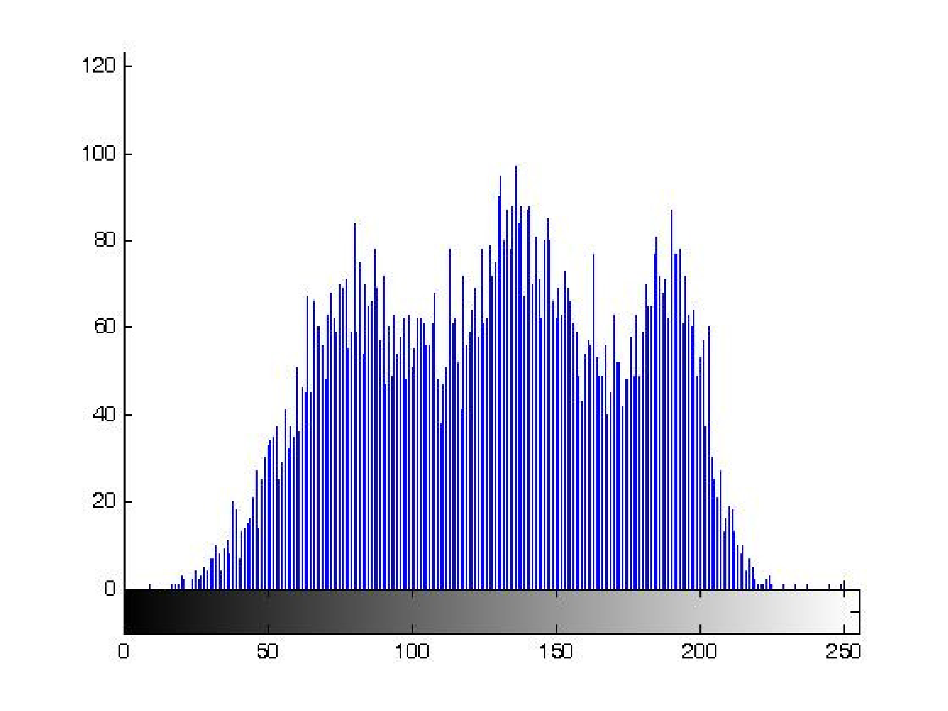


Figure 4.18. Histogram data of the X_2 training image

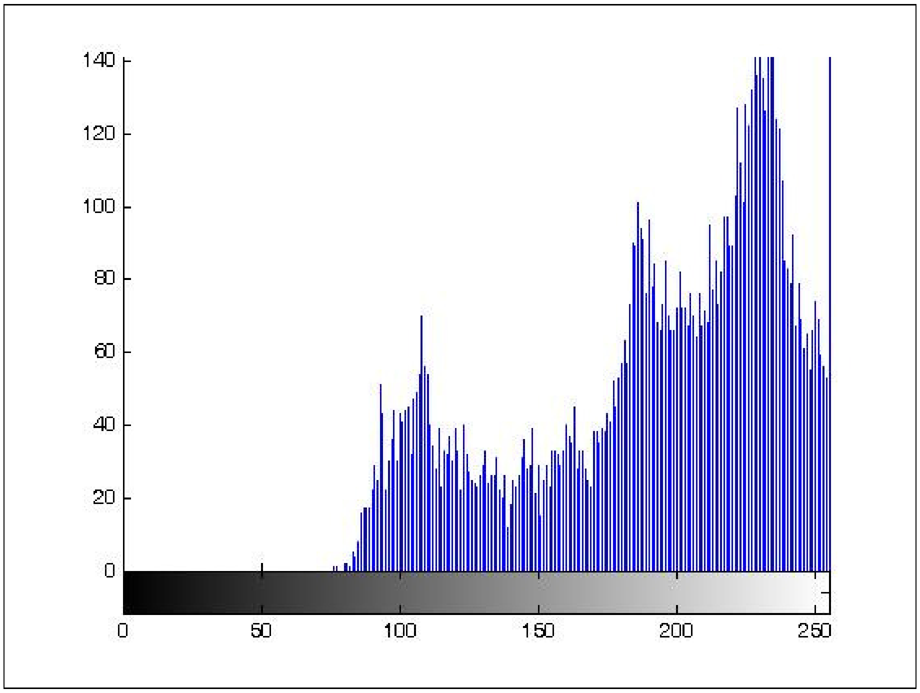


Figure 4.19. Histogram data of the X_3 training image

Table 4.2. Average pixel values (brightness values) for all images

Image Name	Average Pixel Value
airplane	178.68
baboon	129.81
boat	136.14
bridge	113.88
camera	118.72
goldhill	112.21
lena	124.04
parrots	109.56
peppers	120.29
reptile	125.47
X_1 training	63.77
X_2 training	128.61
X_3 training	192.53

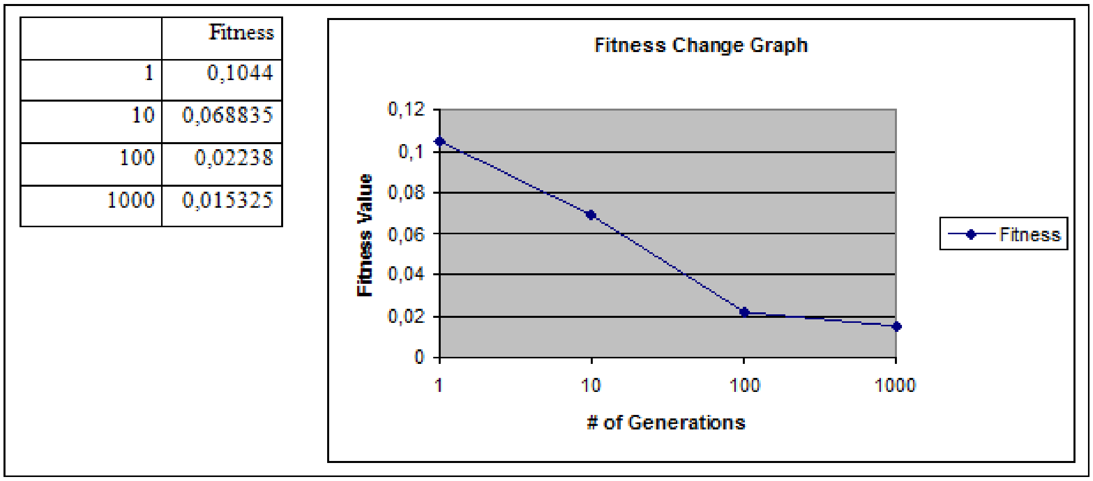
4.2. DOES GENETIC ALGORITHM IMPROVE THE FILTER?

In this part of the testing; what we aim is to prove that our GA and the fitness function really improve a filters capability of noise removal and shape preservation.

Therefore, to show this; we stop the GA runs at some intervals and take the soft morphological filter formed at those generations and applied them to a noisy Lena Image and calculated the MSE, MAE and Peak Signal to Noise Ratio (PSNR) Values of the filtered Images. The results of these experiments can be seen in Figures 4.20 – 4.25.

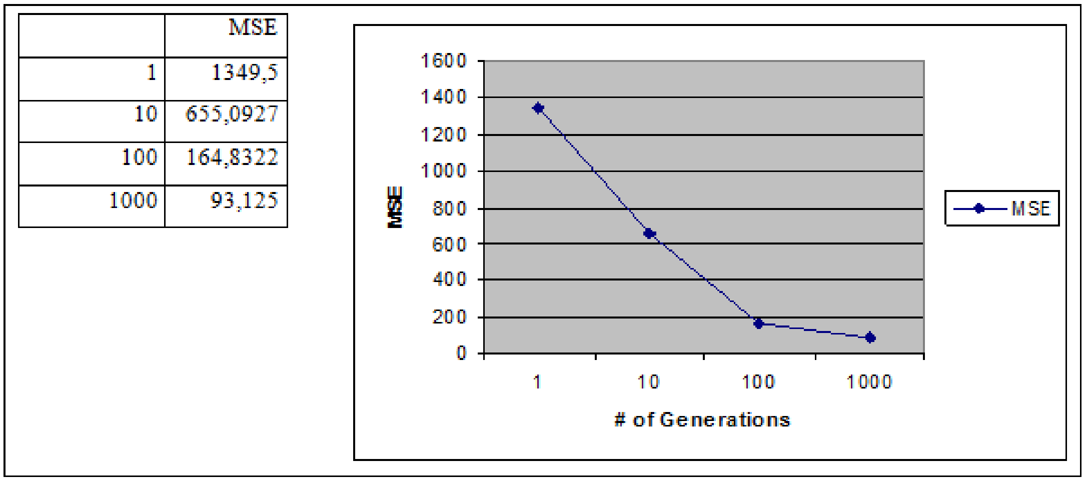
The results clearly showed that, the GA and the fitness function we had used improves a filter through the generations according to these criteria. Also, when we look at the images filtered by the filters from different generations of the GA; we can see clearly that the Image quality is increasing and the noise density is decreasing when the Filters from the future generations are used.

The Filter is generated with the same method and parameters used in the previous tests; by using three different Images with different brightness values and using the same fitness function. We had used 20% noisy versions of the training images, and also for testing the generated filters 256x256 sized Lena Image is used with a noise level of 20%. The MSE Value of the noisy Image is 3767,6.



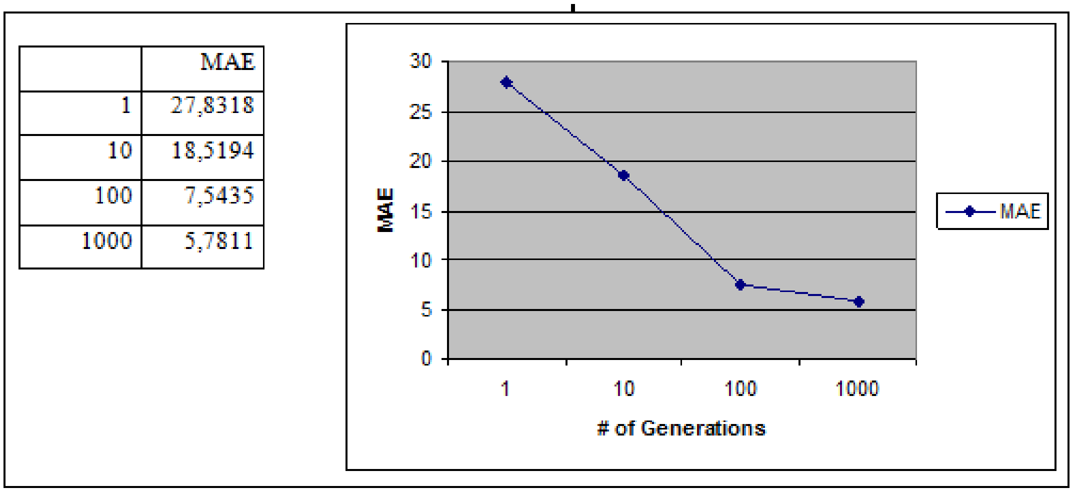
	Fitness
1	0,1044
10	0,068835
100	0,02238
1000	0,015325

Figure 4.20. Fitness change over generations



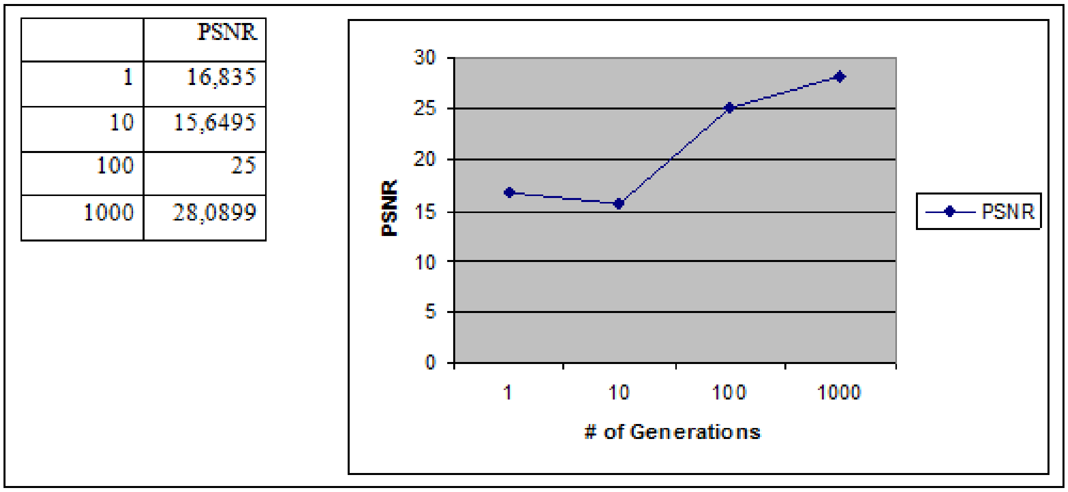
	MSE
1	1349,5
10	655,0927
100	164,8322
1000	93,125

Figure 4.21. MSE value change over generations



	MAE
1	27,8318
10	18,5194
100	7,5435
1000	5,7811

Figure 4.22. MAE value change over generations



	PSNR
1	16,835
10	15,6495
100	25
1000	28,0899

Figure 4.23. PSNR value change over generations

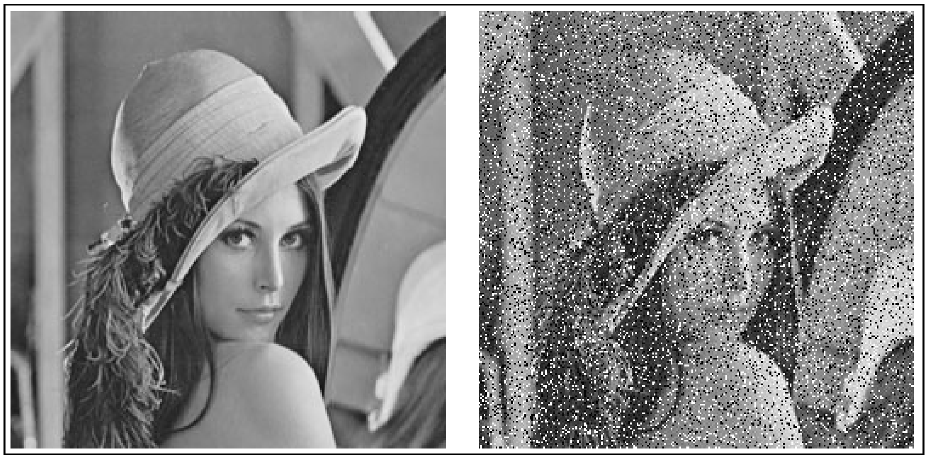


Figure 4.24. The original and the 20% noisy Lena images

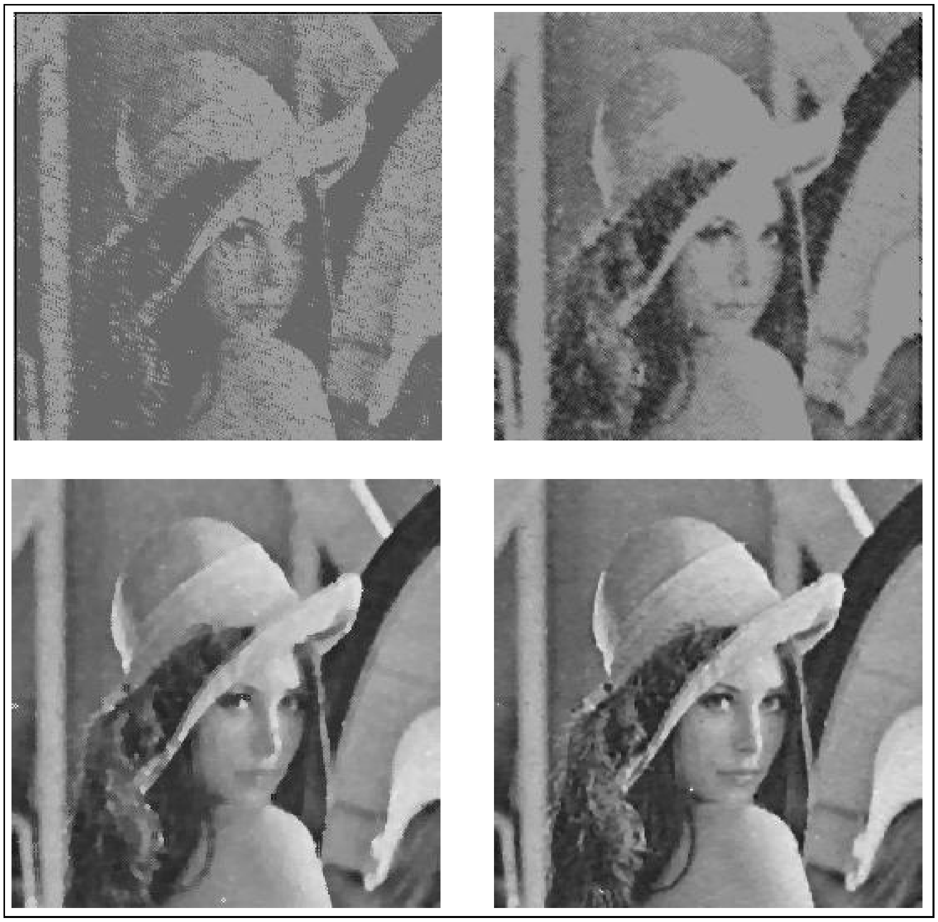


Figure 4.25. Filtered Lena image over generations of GA (1,10,100,1000th generations)

4.3. COMPARISON TO THE MEDIAN FILTER FOR A SINGLE NOISE LEVEL

To test the generated filter, it is applied on ten different images. The generated filter is illustrated in Tables 4.3 – 4.4. In this experiment; each image is corrupted with 20% of salt & pepper noise. But, to achieve statistically significant results, 30 different noisy variations of the images are generated; each having the same percentage of noise, but with different random noise distributions. The filter is applied on each of those images and the resulting images are compared with the original non-corrupted ones using the objectives defined in the fitness function.

The same operations are done using the 3x3, 4x4 and 5x5 median filters on the same images. For the performance test of the generated filter, the average and the best results taken from the median filters are compared with our filter's results and the outcomes of these experiments are shown in Tables 4.5 – 4.6.

The results showed that, according to mean-squared-error and shape error criteria, the generated filter gave better results in both average and best values for all images. According to mean-absolute-error criteria, the result is almost the same, for only one of the images; the 3x3 median filter gave a better result. According to brightness error criteria, for half of the images the generated filter gave better results than all sizes of median filters, but for the other images it could not. So, this is the worst criteria for our filter, but according to this criterion it can be considered as equal to the median filter. Therefore, when we look at the results in general, the generated filter outperforms the median filter in a significant way.

Table 4.3. The Chromosome of the generated filter for 20% impulsive noise

Operation Sequence	4	0	4	0					
Neighbourhood	0	0	0	1					
Symmetric	1	1	1	0					
Structuring Elements	156	-10	227	-13	-15	77	73	-194	17
	169	-43	231	206	-203	-153	-230	57	247
	102	-11	157	-33	-10	208	113	-248	104
	-35	-66	-250	-9	-128	122	-195	183	-204
Hard Soft Map	1	0	1	0	0	0	1	0	0
	1	1	0	0	1	1	1	1	1
	1	0	0	0	0	0	1	0	1
	1	0	1	1	1	0	0	0	1
Rank	3	3	2	8					

Table 4.4. The generated filter for 20% impulsive noise

1st Stage : Soft Closing				**2nd Stage : Soft Closing**		
	-10				-11	
-13	-15	-13		-33	-10	-33
	-10				-11	
1	0	1		1	0	0
0	0	0		0	0	0
1	0	0		1	0	1
Repetition Parameter : 3				Repetition Parameter : 2		

Table 4.5. Average values taken from the test done with 30 different noisy variations of the images (first 5 images)

(The training images are portions of the Lena, Mandrill and Flowers images)

	MAE		MSE		Shape Error		Brightness Error	
Baboon Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	10,72	0.02	300,25	1.80	64,66	0.12	0,32	0.07
3x3 Median	11,74	0.03	392,60	3.99	75,18	0.26	0,43	0.08
4x4 Median	14,43	0.03	491,62	2.70	77,68	0.12	0,64	0.07
5x5 Median	14,69	0.03	502,10	3.97	79,06	0.15	0,72	0.08
Boat Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	3,67	0.01	58,84	0.82	26,18	0.12	0,20	0.02
3x3 Median	4,22	0.02	123,17	3.06	38,74	0.41	0,12	0.04
4x4 Median	6,17	0.02	176,19	2.34	38,81	0.16	0,17	0.03
5x5 Median	5,76	0.03	151,44	3.35	37,84	0.26	0,04	0.02
Bridge Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	8,55	0.03	184,34	2.52	49,64	0.22	0,23	0.08
3x3 Median	9,76	0.05	289,72	6.65	62,90	0.57	0,39	0.10
4x4 Median	12,48	0.04	374,73	3.43	64,32	0.20	0,78	0.09
5x5 Median	12,47	0.04	373,07	4.70	64,06	0.26	0,90	0.09
Flowers Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	4,68	0.02	79,60	1.44	28,14	0.20	1,16	0.03
3x3 Median	4,01	0.02	119,47	3.47	38,60	0.58	0,09	0.03
4x4 Median	5,99	0.02	160,71	1.48	37,24	0.12	0,21	0.03
5x5 Median	5,44	0.02	131,49	1.64	35,19	0.16	0,49	0.03
Goldhill Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	3,95	0.01	49,56	0.80	25,04	0.13	0,06	0.02
3x3 Median	4,58	0.02	111,79	2.97	37,89	0.42	0,21	0.03
4x4 Median	6,15	0.02	122,13	1.61	34,36	0.13	0,41	0.02
5x5 Median	5,99	0.02	121,17	3.70	34,97	0.36	0,51	0.03

Table 4.6. Average values taken from the test done with 30 different noisy variations of the images (last 5 images)

(The training images are portions of the Lena, Mandrill and Flowers Images)

	MAE		MSE		Shape Error		Brightness Error	
Lena Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	2,52	0.01	31,91	0.66	18,34	0.14	0,03	0.02
3x3 Median	2,79	0.02	79,87	3.05	30,59	0.54	0,17	0.03
4x4 Median	4,53	0.02	120,77	1.85	29,15	0.17	0,31	0.03
5x5 Median	3,81	0.02	79,73	2.98	25,94	0.39	0,40	0.03
Peppers Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	2,53	0.01	31,21	1.22	18,21	0.27	0,16	0.02
3x3 Median	2,80	0.02	78,96	3.59	31,03	0.60	0,11	0.03
4x4 Median	4,41	0.01	131,34	2.44	30,40	0.21	0,12	0.02
5x5 Median	3,55	0.02	67,40	2.44	24,68	0.35	0,25	0.03
Parrots Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	3,05	0.02	77,08	1.73	31,40	0.24	0,13	0.04
3x3 Median	3,36	0.03	125,00	4.35	40,50	0.55	0,09	0.05
4x4 Median	4,69	0.03	150,92	3.79	39,89	0.30	0,29	0.04
5x5 Median	4,29	0.04	141,83	5.02	40,10	0.40	0,26	0.05
Camera Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	5,27	0.04	169,86	3.74	46,61	0.35	0,16	0.07
3x3 Median	5,94	0.07	276,02	10.15	59,91	0.89	0,58	0.09
4x4 Median	8,19	0.05	414,40	6.75	64,48	0.37	1,28	0.08
5x5 Median	7,77	0.06	373,53	8.06	62,85	0.48	1,20	0.09
Airplane Image	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
Our Filter	2,90	0.01	47,02	1.15	21,77	0.22	0,23	0.02
3x3 Median	3,21	0.02	110,46	3.46	35,58	0.54	0,20	0.04
4x4 Median	5,20	0.02	191,47	3.51	36,24	0.26	0,21	0.03
5x5 Median	4,52	0.02	128,10	3.22	32,54	0.34	0,14	0.04

4.4. COMPARISON OF THE EVOLVED FILTERS TO THE STATE-OF-THE-ART FILTERS

In the previous part of the tests; we had generated a filter for a single level of noise and test that filter for that specific level of noise which it is trained for. But, in this part of the testing; we had changed this approach.

In this part, for 9 different levels of noise (10% - 20% - 30% - ... – 90%), 9 different filters are generated. This is done by using the same 3 Images, different versions of those Images are generated with those levels of noise and they are given as inputs to the GA. And the filter is generated with the same fitness function and with the same methods as the previous tests. The GA is again ran for 30 separate times and the filter resulted from the best fit generation is used for testing purposes.

At first 3 images are generated for the 3 small test Images which has a 10% level of noise. And the GA is run for those 6 Images (3 non-noisy, 3 noisy) for 30 runs, after those runs; the filter from the best generation is taken out for testing.

For testing; 3 Images are used; Lena, Peppers, Mandrill which are shown in the Figure 4.26. These images are selected because they are the mostly used Images in the Literature and we have extensive test results for those Images in the Literature. But, at this point; we need to generate noisy versions of these Images which has the same Mean-Squared-Error Value.

Therefore, by using computational brute-force try-and-error approach, 9 different versions of those 3 Images with different levels of noise from 10% to 90% are generated. But those noisy Images are generated in such a way that they will have the same MSE Values with the Images used in the test results seen in the Literature.

After generating all those noisy Images for 9 levels of noise, the filter generated by using images with 10% level of noise is applied on each of them, and the 27 different results are collected for 3 Images x 9 different levels of noise.

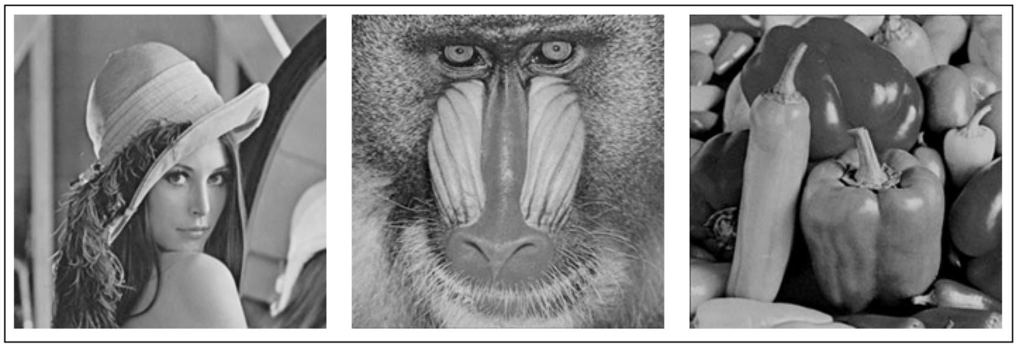


Figure 4.26. 512x512 Lena, Mandrill, Peppers images

These whole set of operations are also repeated for other levels of noise. Therefore, 9 different filters are generated by training the GA for 9 different levels of noise. And, those filters are applied on those 3 Images with noise levels ranging from (10% - 90%).

You can see all the results taken from this test in the Tables 4.7 – 4.12. At each page, the test results for a specific image are presented. In the “Our Method” part; you can see the MSE values of the images filtered by the filters generated by our GA with different training noise levels. For our method; the best filter results for each level of noise are marked as bold. And, in below, the MSE values of the images filtered by the filters in the literature can be seen.

From these results; there is more than one conclusion that we can perceive. The first one is; in most cases, the filters generated with our method give their best results at the noise level that they are generated.

The second conclusion is, our filter does not give the best results among all the filters, but if we look at the results of each for their generated noise level, it can be observed that our method gets a rank in the best 3 or 4 filters among the state-of-the-art filters in the literature as illustrated in Table 4.13 and Figure 4.27. The values in Table 4.13 are formed with the values taken from the Tables A.1 – A.3. Each approach is ranked from 1 (best) to 14 (worst) based on the MSE values, and then the average ranking of each approach is plotted for each test image considering different levels of noise. This of course requires prior knowledge about the level of the noise that we have for a given filter.

Table 4.7. Comparison of our approach to the other approaches from literature based on MSE values for the Lena image - Part I

(The training images are portions of the Lena, Mandrill and Flowers images)

	10	20	30	40	50
Noisy image	1852,70	3767,60	5563,60	7451,50	9268,40
Our Method	63,96	93,12	129,28	162,34	206,68
10%	**63,96**	125,41	336,90	823,43	1717,71
20%	67,64	**93,12**	151,24	296,04	659,00
30%	79,71	101,04	136,61	224,36	447,25
40%	95,13	108,01	**129,28**	171,60	275,60
50%	128,16	138,57	157,62	188,62	257,80
60%	120,30	130,94	142,82	**162,34**	**206,68**
70%	230,67	233,72	254,33	260,33	305,43
80%	215,40	220,46	227,01	243,98	265,07
90%	281,29	291,88	306,34	338,26	387,36
AIF	12,46	24,13	47,44	71,68	114,05
IMF	124,05	140,06	158,84	203,36	261,87
SMF - 3x3	87,09	148,82	353,46	958,83	2046,10
PSM	54,92	85,76	132,48	272,63	647,30
SDROM	60,52	136,35	408,80	1105,70	2339,10
SDROMR	56,79	91,21	171,88	287,33	553,76
IRF	61,15	127,84	366,24	981,06	2098,80
ACWM	51,97	121,51	367,27	993,73	2119,40
ACWMR	46,95	85,67	170,71	299,57	536,23
CWM	76,27	259,94	727,17	1729,30	3273,40
YÜKSEL	63,63	94,21	151,67	256,73	417,24
RUSSO	51,45	111,90	208,57	346,21	605,95
HAF	12,53	24,66	49,61	83,16	121,66

Table 4.8. Comparison of our approach to the other approaches from literature based on MSE values for the Lena image - Part II

(The training images are portions of the Lena, Mandrill and Flowers images)

	60	70	80	90	Average
Noisy image	11152,00	13078,00	14769,00	16804,00	9300,76
Our Method	356,48	629,36	1235,10	2356,58	
10%	3417,09	5735,92	8579,54	12308,65	3310,87
20%	1668,65	3487,04	6238,84	10310,34	2297,21
30%	1171,02	2744,55	5369,45	9653,81	1992,81
40%	600,56	1650,92	3627,65	7965,03	1462,42
50%	519,31	1330,86	3280,31	8206,04	1420,78
60%	370,18	1022,40	2851,88	7606,07	1261,42
70%	381,12	842,92	2045,64	5043,67	959,85
80%	**356,48**	**629,36**	1332,46	3726,48	721,75
90%	521,77	762,78	**1235,10**	**2356,58**	648,23
AIF	153,18	204,08	276,45	383,37	142,98
IMF	491,57	1402,70	3971,50	10140,00	1877,11
SMF - 3x3	3856,60	6769,10	9938,00	14269,00	4269,67
PSM	1938,30	5036,00	9495,10	14862,00	3613,83
SDROM	4322,30	7404,70	10619,00	14831,00	4580,83
SDROMR	1071,80	2452,10	5615,90	13152,00	2605,86
IRF	3918,50	6821,70	9982,40	14290,00	4294,19
ACWM	3951,00	6859,10	10015,00	14311,00	4310,00
ACWMR	1007,80	2042,30	4230,20	9975,40	2043,87
CWM	5431,60	8350,40	11291,00	15060,00	5133,23
YÜKSEL	647,59	970,07	1375,40	1968,40	660,55
RUSSO	1114,90	2175,50	4372,30	8848,00	1981,64
HAF	154,23	221,77	463,38	491,17	180,24

Table 4.9. Comparison of our approach to the other approaches from literature based on MSE values for the Mandrill image - Part I

(The training images are portions of the Lena, Mandrill and Flowers images)

	10	20	30	40	50
Noisy image	1766,50	3585,30	5351,70	7093,50	8898,20
Our Method	193,38	250,13	308,52	361,50	423,27
10%	**193,38**	273,47	486,69	928,09	1909,83
20%	216,13	**250,13**	311,09	452,69	858,00
30%	248,72	270,17	**308,52**	392,61	646,38
40%	279,36	296,53	320,37	**361,50**	487,28
50%	335,46	347,61	366,95	386,44	448,94
60%	332,73	340,80	354,34	370,92	**423,27**
70%	403,74	409,45	420,37	430,45	472,66
80%	391,25	398,22	408,75	420,01	455,22
90%	410,60	417,88	431,38	449,19	496,96
AIF	31,16	70,37	113,88	164,86	220,92
IMF	324,10	340,02	358,99	386,60	456,02
SMF - 3x3	271,04	329,47	560,00	1094,30	2184,30
PSM	108,39	147,80	205,12	335,73	736,27
SDROM	165,39	259,65	544,67	1187,70	2443,10
SDROMR	171,74	224,09	312,82	429,68	692,46
IRF	171,56	252,63	497,92	1071,00	2190,20
ACWM	125,94	215,00	470,31	1052,30	2180,10
ACWMR	115,20	176,27	263,32	381,32	620,87
CWM	184,00	348,50	836,76	1795,20	3319,70
RUSSO	46,95	85,67	170,71	299,57	536,23
YÜKSEL	76,27	259,94	727,17	1729,30	3273,40
HAF	31,77	72,04	115,01	172,19	227,46

Table 4.10. Comparison of our approach to the other approaches from literature based on MSE values for the Mandrill image - Part II

(The training images are portions of the Lena, Mandrill and Flowers images)

	60	70	80	90	Average
Noisy image	10623,00	12415,00	14201,00	15998,00	8881,36
Our Method	521,01	749,35	1115,79	1856,10	
10%	3407,13	5459,72	8321,25	11616,21	3259,59
20%	1753,92	3251,77	5943,98	9561,00	2259,89
30%	1310,97	2605,18	5182,08	8980,67	1994,56
40%	835,22	1777,41	3818,16	7306,28	1548,25
50%	650,10	1334,54	3438,57	7207,29	1451,64
60%	573,06	1165,95	3019,14	6854,64	1343,55
70%	555,06	875,49	2027,21	4792,20	1038,73
80%	**521,01**	778,69	1443,19	3269,92	808,71
90%	571,95	**749,35**	**1115,79**	**1856,10**	650,01
AIF	293,77	370,09	461,06	599,19	258,37
IMF	657,79	1653,40	4082,50	9688,00	1994,16
SMF - 3x3	3836,80	6583,00	9739,10	13720,00	4257,56
PSM	1932,40	5146,60	9665,40	14527,00	3644,97
SDROM	4264,30	7185,30	10451,00	14272,00	4530,35
SDROMR	1122,90	2482,90	5432,60	12614,00	2609,24
IRF	3856,00	6614,30	9786,60	13742,00	4242,47
ACWM	3851,70	6620,20	9798,60	13749,00	4229,24
ACWMR	975,77	1888,50	3713,40	8619,00	1861,52
CWM	5272,30	8037,50	10960,00	14346,00	5011,11
RUSSO	1007,80	2042,30	4230,20	9975,40	2043,87
YÜKSEL	5431,60	8350,40	11291,00	15060,00	5133,23
HAF	301,71	387,19	482,01	611,48	266,76

Table 4.11. Comparison of our approach to the other approaches from literature based on MSE values for the Peppers image - Part I

(The training images are portions of the Lena, Mandrill and Flowers images)

	10	20	30	40	50
Noisy image	1940,10	3988,80	5897,70	7914,50	9960,80
Our Method	59,68	95,49	131,67	179,93	232,53
10%	**59,68**	141,56	335,98	896,96	2002,49
20%	65,95	**95,49**	141,92	330,03	799,81
30%	78,51	101,45	**131,67**	233,20	516,94
40%	112,00	125,67	142,25	190,90	313,82
50%	142,05	162,01	176,23	213,34	276,59
60%	132,36	143,66	152,65	**179,93**	**232,53**
70%	254,87	267,27	270,68	306,71	350,39
80%	363,50	369,63	376,93	399,49	423,48
90%	759,92	769,86	778,24	811,31	876,05
AIF	12,95	28,77	52,32	83,96	123,28
IMF	123,82	149,80	190,54	249,05	335,41
SMF - 3x3	77,69	160,17	389,36	1021,90	2161,00
PSM	55,24	94,58	159,79	301,41	714,25
SDROM	63,70	195,43	482,43	1240,30	2528,90
SDROMR	61,34	149,97	268,08	474,06	750,02
IRF	60,14	182,08	437,96	1126,40	2299,90
ACWM	55,70	181,99	442,58	1139,30	2317,90
ACWMR	54,51	142,14	259,92	463,44	727,67
CWM	84,55	303,75	818,32	1860,70	3557,80
YÜKSEL	228,79	321,47	467,54	670,77	946,99
RUSSO	72,12	160,29	297,64	507,32	820,94
HAF	18,03	39,12	67,14	96,36	137,03

Table 4.12. Comparison of our approach to the other approaches from literature based on MSE values for the Peppers image - Part II

(The training images are portions of the Lena, Mandrill and Flowers images)

	60	70	80	90	Average
Noisy image	11929,00	13993,00	15930,00	17908,00	9940,21
Our Method	403,68	829,00	1806,19	3267,65	
10%	3695,31	6464,19	9521,83	13342,63	3646,07
20%	1851,95	4166,26	7234,87	11588,18	2627,47
30%	1225,95	3183,28	6046,20	10606,69	2212,42
40%	657,04	1880,93	4304,51	8589,78	1631,73
50%	517,54	1604,82	4129,07	9010,14	1623,23
60%	**403,68**	1260,99	3487,42	8291,24	1428,51
70%	470,06	946,16	2570,58	5990,11	1142,75
80%	514,31	**829,00**	**1806,19**	4365,73	944,91
90%	990,60	1355,43	1938,14	**3267,65**	1154,81
AIF	171,87	233,66	331,48	487,90	169,58
IMF	667,77	1693,60	4329,20	10742,00	2053,47
SMF - 3x3	4151,50	7271,20	10690,00	15164,00	4565,20
PSM	1952,20	5280,30	9673,30	15099,00	3703,34
SDROM	4674,00	7960,80	11444,00	15787,00	4930,73
SDROMR	1568,00	3167,30	6270,00	13218,00	2880,75
IRF	4309,50	7444,10	10835,00	15306,00	4666,79
ACWM	4333,40	7471,50	10859,00	15324,00	4680,60
ACWMR	1395,30	2753,00	4820,70	10357,00	2330,41
CWM	5858,40	8999,30	12226,00	16049,00	5528,65
YÜKSEL	1343,80	1852,70	2468,70	3267,10	1285,32
RUSSO	1420,30	2528,40	4767,70	9732,40	2256,35
HAF	219,60	288,96	378,12	520,49	196,09

Table 4.13. Average rank of each filter over all noisy images

(The training images are portions of the Lena, Mandrill and Flowers images)

	Lena	Mandrill	Peppers
Our Method	4,44	5,11	3,56
AIF	1,00	1,00	1,00
IMF	6,78	7,11	5,78
SMF - 3x3	10,56	10,33	9,78
PSM	7,56	6,89	6,33
SDROM	12,11	11,11	12,44
SDROMR	7,22	7,44	7,56
IRF	10,56	9,89	10,44
ACWM	10,78	9,11	10,89
ACWMR	5,67	5,11	5,78
CWM	13,78	13,11	13,67
YÜKSEL	5,33	4,56	8,33
RUSSO	7,22	12,22	7,44
HAF	2,00	2,00	2,00

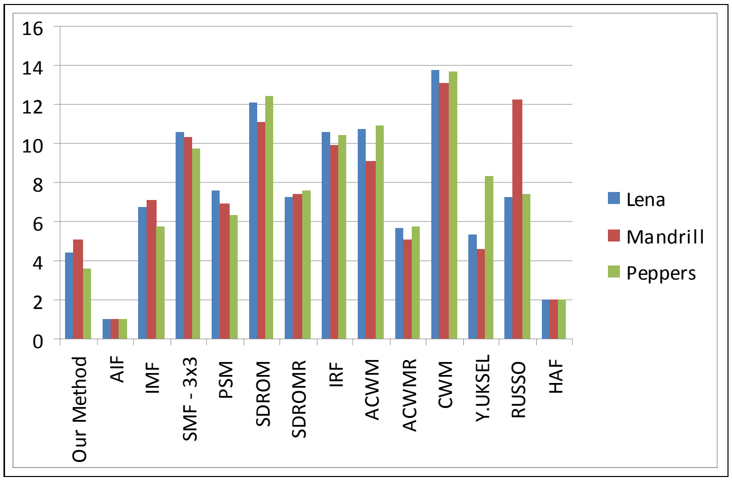


Figure 4.27. Comparison of our approach to the other approaches in literature based on the average rankings over all noisy data using MSE values for a given image

(Here, the filters of our method are used on the noise level that they are trained for)

In general, there is no prior knowledge regarding the noise level for a given image, hence we compare the performance of different filters generated using our approach at different noise levels as illustrated in Table 4.14 and Figure 4.28. The below table is formed with the values taken from the Tables A.4 – A.6. The results indicate that the filter obtained by using training instances with 60% noise yields a better average performance.

Table 4.14. Average rank of each filter over all noisy images

(The training images are portions of the Lena, Mandrill and Flowers images)

	Lena	Mandrill	Peppers
10%	15,44	15,00	14,11
20%	12,56	12,33	11,11
30%	10,78	11,33	9,44
40%	8,33	9,67	7,67
50%	9,44	9,89	8,78
60%	7,56	8,67	6,67
70%	10,67	10,56	9,56
80%	9,11	9,33	10,22
90%	11,44	10,67	13,22
AIF	1,00	1,00	1,00
IMF	10,44	10,78	9,67
SMF - 3x3	17,56	17,11	16,44
PSM	12,22	10,89	10,89
SDROM	18,78	17,33	19,11
SDROMR	11,56	11,33	12,33
IRF	17,11	16,11	17,11
ACWM	17,22	15,22	17,56
ACWMR	9,78	7,89	10,11
CWM	20,89	19,89	20,78
YÜKSEL	7,56	7,56	13,00
RUSSO	11,56	18,44	12,22
HAF	2,00	2,00	2,00

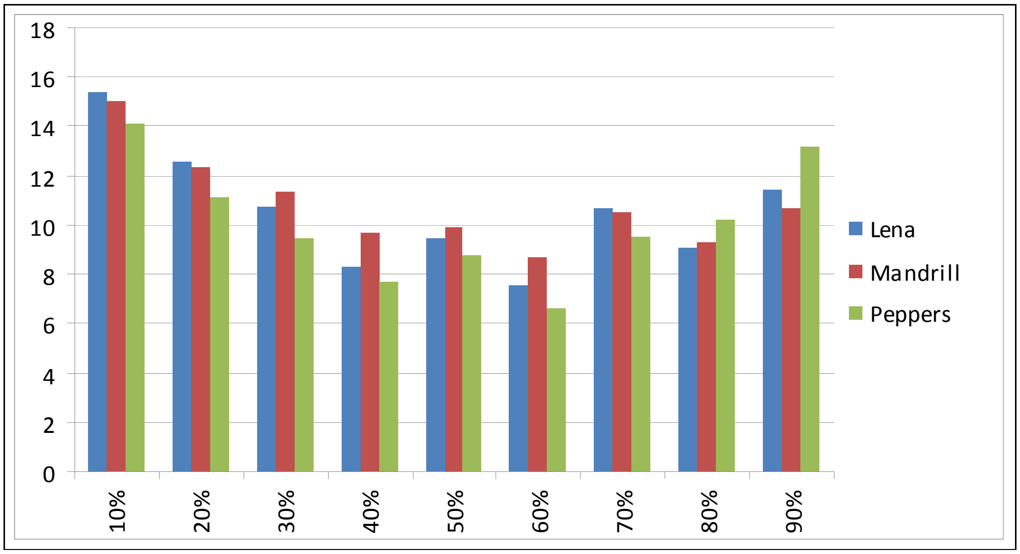


Figure 4.28. Average performance ranking of the filters, which are obtained by using the training instances with a given noise level, over all noisy data

And as the last part of the comparisons; as shown in Table 4.15 and Figure 4.29 which are formed with the values taken from the Tables A.7 – A.9. A comparison using the filter generated by training over images with 60% noise yields a comparable performance to the existing approaches. Still, the state of the approaches cannot be beaten.

Table 4.15. Average rank of each filter over all noisy images

(The training images are portions of the Lena, Mandrill and Flowers images)

	Lena	Mandrill	Peppers
Our Method - 60%	5,33	6,00	4,56
AIF	1,00	1,00	1,00
IMF	6,78	6,89	5,67
SMF - 3x3	10,44	10,11	9,67
PSM	7,56	6,89	6,33
SDROM	12,11	11,00	12,33
SDROMR	7,22	7,33	7,44
IRF	10,44	9,78	10,33
ACWM	10,67	9,11	10,89
ACWMR	5,67	5,11	5,67
CWM	13,67	13,11	13,56
YÜKSEL	5,00	4,56	8,22
RUSSO	7,11	12,11	7,33
HAF	2,00	2,00	2,00

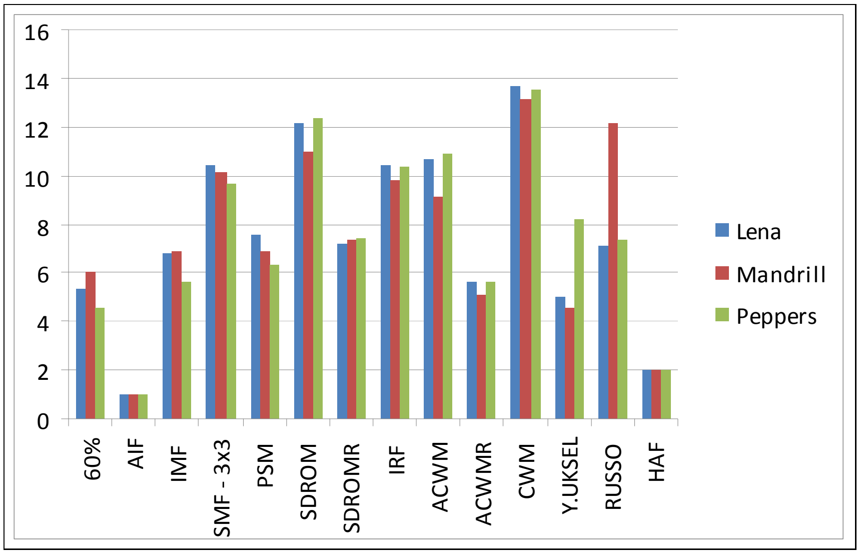


Figure 4.29. Comparison of approaches using average ranking of that are obtained after training for each image with the given noise level

4.5. PERFORMANCE OF THE BEST EVOLVED FILTER

As we had concluded from the results presented at the end of the previous Section. According to the average ranking between the filters generated with training data which has noise level in the range (10% - 90%). It is seen that the filter trained with images that has a noise level of 60% gets the best average ranking result through the other filters generated. The filter is illustrated in the Tables 4.16 – 4.17.

Therefore, in this Section, the filter generated with 60% level of noise is presented and the Clean, Noisy and Filtered Versions of the three test Images processed by this filter are shown in the Figures 4.30 – 4.32.

Table 4.16. The Chromosome of the generated filter for 60% impulsive noise

Operation Sequence	0	3	4	4					
Neighbourhood	0	1	1	1					
Symmetric	0	1	1	1					
Structuring Elements	169	103	-17	122	78	242	228	-105	18
	-6	-4	-25	-5	-1	-156	-190	-36	152
	9	11	163	13	-229	-101	128	-250	-4
	4	4	-250	38	12	62	55	-177	82
Hard Soft Map	0	1	0	0	0	0	0	1	1
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
Rank	8	5	5	4					

Table 4.17. The generated filter for 60% of impulsive noise

1st Stage : Soft Opening		
-6	-4	-6
-5	-1	-5
-6	-4	-6
0	0	0
0	0	0
0	0	0
Repetition Parameter : 5		
2nd Stage : Soft Closing		
9	11	9
13	-229	13
9	11	9
0	0	0
0	0	0
0	0	0
Repetition Parameter : 5		
3rd Stage : Soft Closing		
4	4	4
38	12	38
4	4	4
0	0	0
0	0	0
0	0	0
Repetition Parameter : 4		

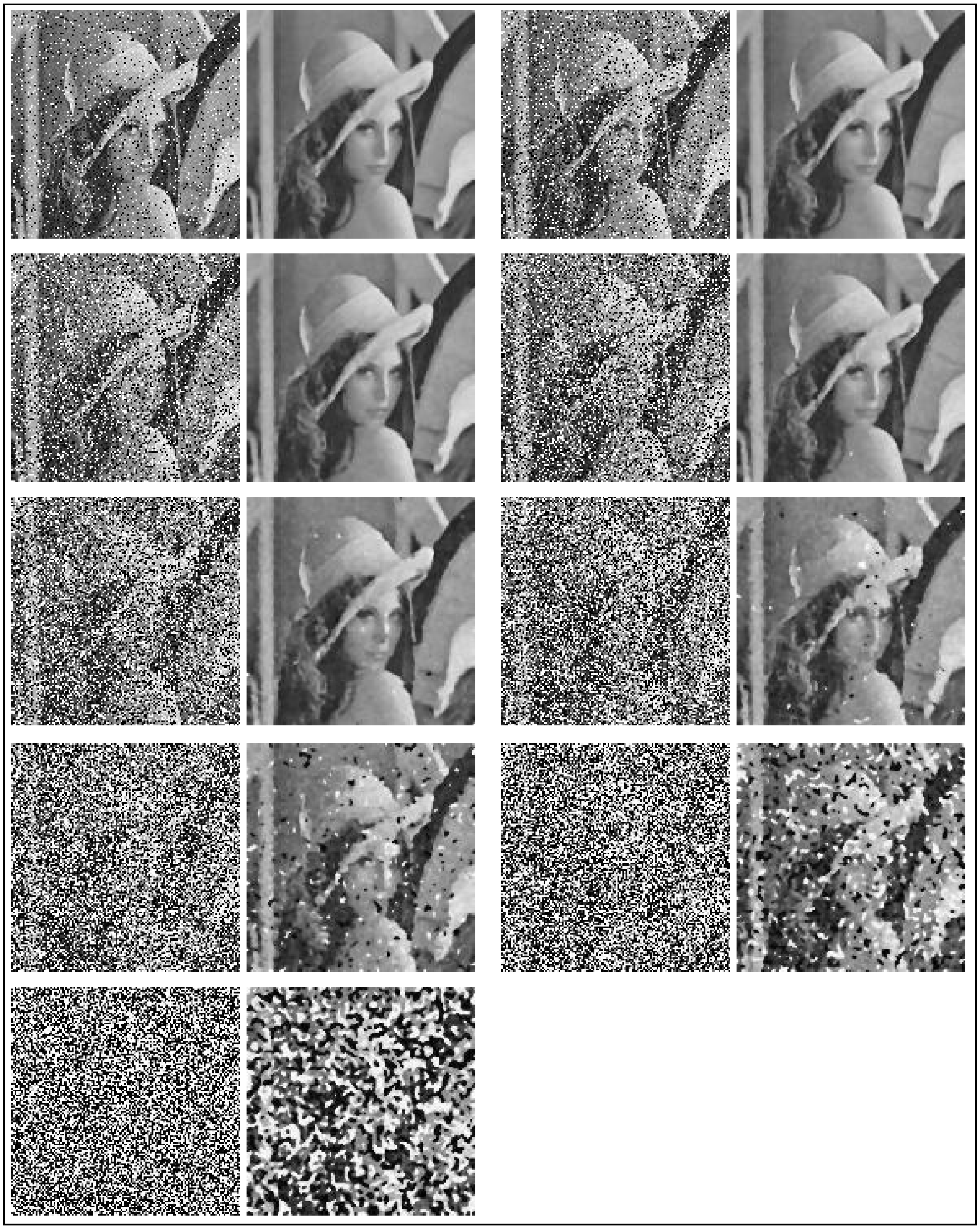


Figure 4.30. Filtered Lena images with noise levels (10% - 90%) with the filter trained with 60% level of noise

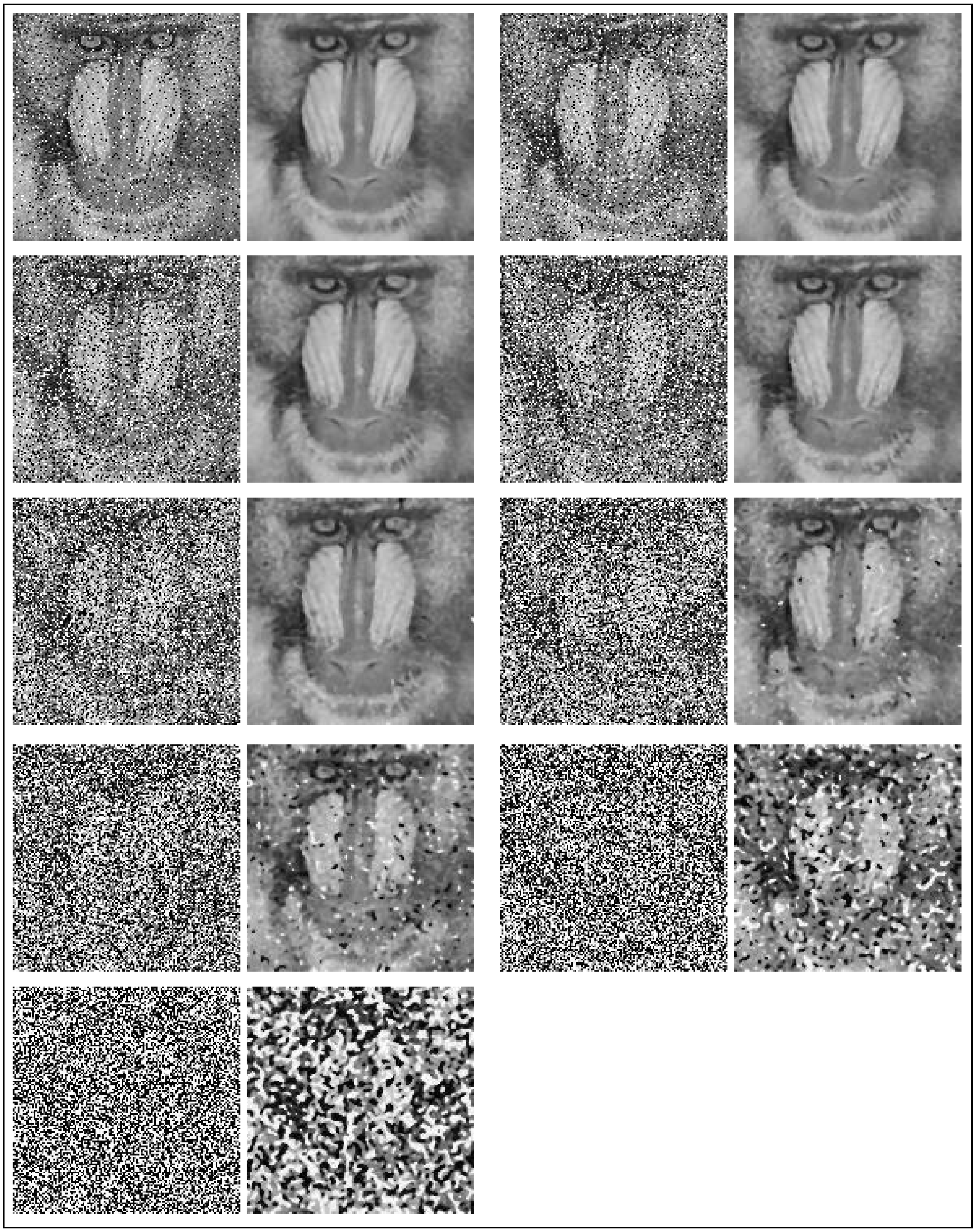


Figure 4.31. Filtered Mandrill images with noise levels (10% - 90%) with the filter trained with 60% level of noise

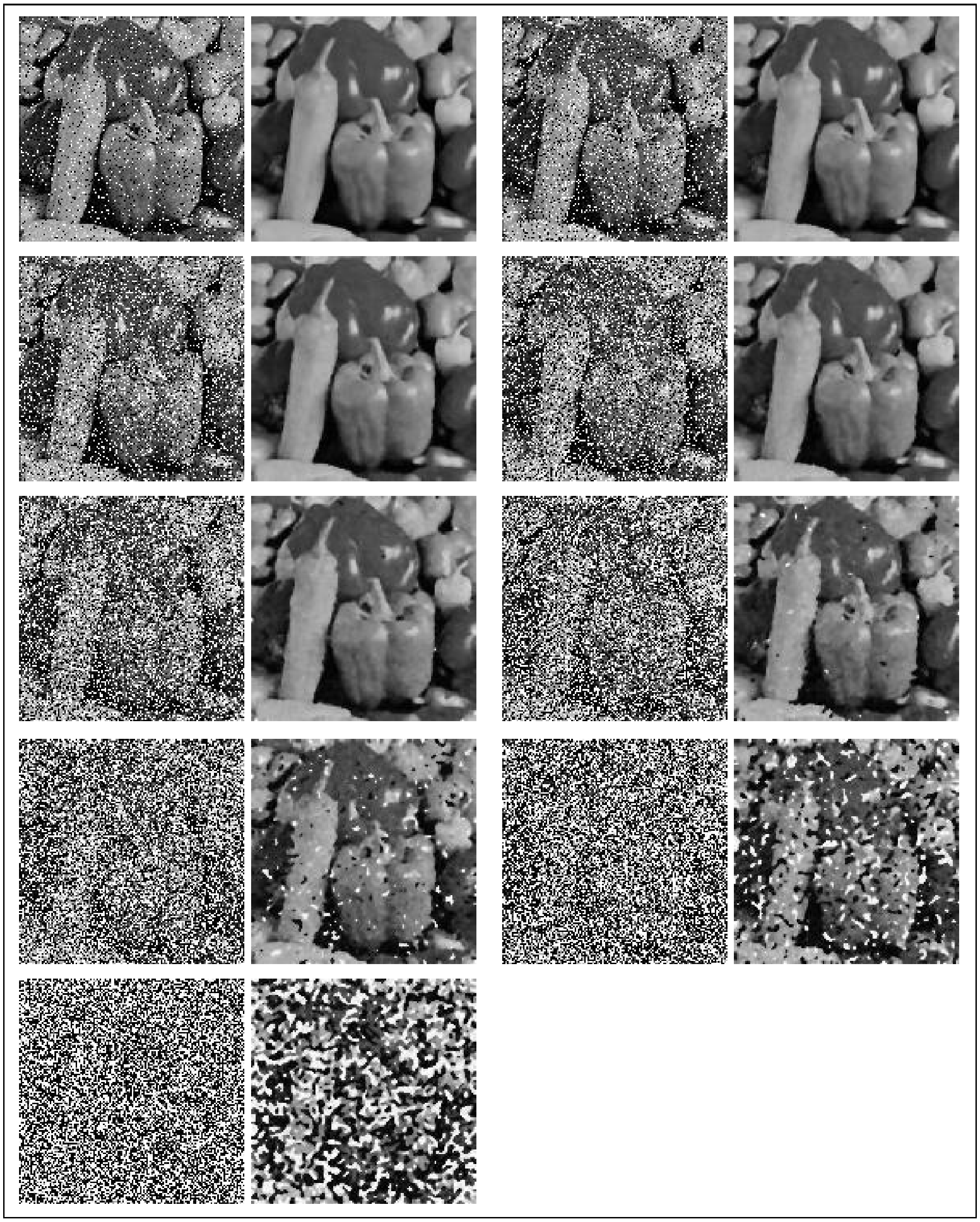


Figure 4.32. Filtered Peppers images with noise levels (10% - 90%) with the filter trained with 60% level of noise

4.6. PERFORMANCE OF THE BEST FILTER OVER THE TRAINING IMAGES

In this part; the selected best filter which is trained at 60% Noise Level is applied to the 3 training Images. With this test; our aim is to show that the trained filter is also got good results with the training images itselves. Since, those images are used to rank and evolve the filter during generations of the GA; it is expected to have good results when the filter is applied on the training images. And from the results seen in the Tables 4.18 and Figure 4.33; the result of the filtering of the training images is really good both in its own context and also compared to other images.

First, salt&pepper noise ranging between (10%-90%) is applied on the training images which give us nine different noisy images for each training image. After generating the noisy Images, the select filter is applied on each of them. In the below table and figure; the difference between the filtered and the original images are listed with mean-squared-error criteria. Also, the results for the Lena, Mandrill and Peppers Images are also listed here.

Table 4.18. Selected filter results on the training images & Lena, Mandrill, Peppers images (MSE criteria)

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	X_1 (64)	X_2 (128)	X_3 (192)	Lena	Mandrill	Peppers
10%	67,96	398,44	117,84	120,30	332,73	132,36
20%	73,82	409,86	135,27	130,94	340,80	143,66
30%	81,44	428,39	147,90	142,82	354,34	152,65
40%	123,53	471,58	163,77	162,34	370,92	179,93
50%	142,09	523,40	194,39	206,68	423,27	232,53
60%	246,46	812,96	319,86	370,18	573,06	403,68
70%	1443,61	1280,15	1015,76	1022,40	1165,95	1260,99
80%	3482,46	3095,32	3112,49	2851,88	3019,14	3487,42
90%	8607,60	7454,19	8238,67	7606,07	6854,64	8291,24
Average	1585,44	1652,70	1493,99	1401,51	1492,76	1587,16

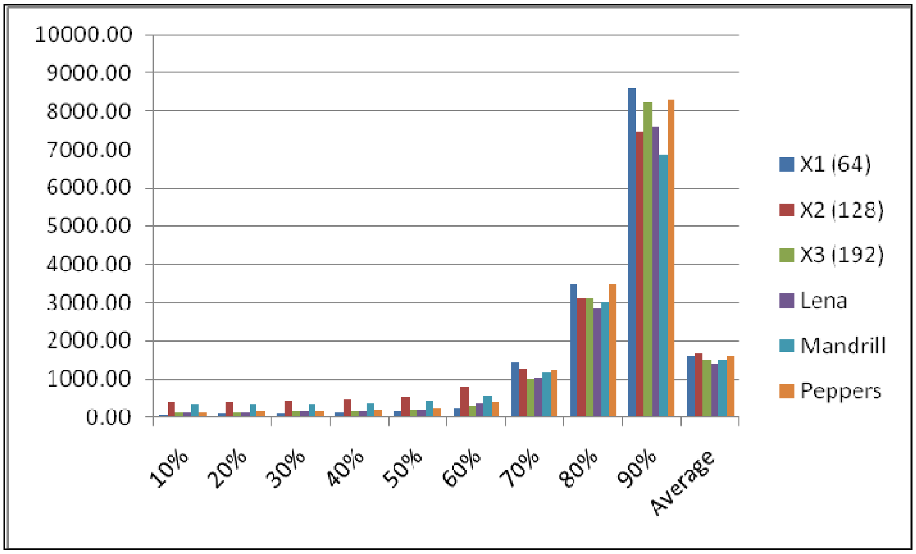


Figure 4.33. Selected filter results on the training images & Lena, Mandrill, Peppers images (MSE criteria)

4.7. PERFORMANCE OF THE BEST FILTER OVER THE IMAGES WITH GAUSSIAN NOISE

In this part; a completely different noise type is used. Although our filter is trained on impulsive noise added images; we want to test and see the capabilities of the generated filter for Gaussian noise added images, also.

In salt and pepper noise (impulsive noise), pixels in the image are very different in color or intensity from their surrounding pixels; the defining characteristic is that the value of a noisy pixel bears no relation to the color of surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, hence the term salt and pepper noise. Typical sources include flecks of dust inside the camera, or with digital cameras, faulty CCD elements.

In Gaussian noise, each pixel in the image will be changed from its original value by a (usually) small amount. A histogram, a plot of the amount of distortion of a pixel value against the frequency with which it occurs, shows a normal (Gaussian) distribution of noise. While other distributions are possible, the Gaussian distribution is usually a good model, due to the central limit theorem that says that the sum of different noises tends to approach a Gaussian distribution.

Gaussian noise is applied on the 10 test images used in the previous Sections and the 3 training images. The Variance range of the applied Gaussian noise is ranging from 10% to 90%. After that; the 3 by 3 median filter and the selected filter which is trained at 60% Noise Level is applied to all these images and the results are recorded according to mean-squared-error criteria. The outcomes of this experiment are illustrated in the Tables 4.19 – 4.22

The noise is generated using MATLAB Application with the imnoise function, the default mean value is used and the variance value is given within the range of (10 – 90). The test results show that, our method also can eliminate Gaussian noise and gets significantly better results than the median filter for all images and all noise levels.

Table 4.19. Selected filter results on the Gaussian Noise added training images & 10 test images (MSE criteria)

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Variance	10%	20%	30%	40%	50%	60%	70%	80%	90%	Average
img64	302.72	559.33	747.73	754.74	1202.27	1298.77	1488.67	1215.65	1668.23	1026.46
img128	661.18	906.44	1147.19	1375.37	1624.70	1757.60	1895.79	2307.44	2209.89	1542.85
img192	359.50	535.06	727.85	914.04	1141.13	1152.28	1468.60	1683.09	1572.46	1061.56
airplane	383.56	584.42	759.72	928.64	1076.05	1245.39	1369.31	1567.58	1718.49	1070.35
baboon	669.45	907.42	1117.78	1358.77	1558.14	1722.10	1985.51	2123.05	2301.10	1527.03
Boat	374.43	597.75	816.50	983.35	1174.31	1382.08	1561.87	1721.68	1904.98	1168.55
Bridge	595.42	828.47	1062.75	1223.51	1407.05	1656.60	1790.70	1962.27	2159.26	1409.56
camera	545.62	756.17	1010.60	1187.30	1337.88	1531.73	1705.44	1916.04	2114.30	1345.01
goldhill	325.73	543.98	747.81	953.63	1118.93	1323.80	1475.47	1684.21	1824.55	1110.90
Lena	308.85	527.90	723.42	933.92	1119.14	1313.05	1510.73	1668.58	1833.52	1104.35
parrots	357.41	572.78	771.17	961.73	1186.96	1399.18	1551.92	1702.00	1931.74	1159.43
peppers	331.10	544.61	752.96	955.84	1124.76	1294.10	1496.64	1634.10	1807.25	1104.59
Reptile	452.73	686.63	912.97	1127.09	1278.18	1543.82	1690.18	1891.15	2096.32	1297.67
Average	435.98	657.77	869.11	1050.61	1257.65	1432.35	1614.68	1775.14	1934.01	

Table 4.20. Median filter (3x3) results on the Gaussian Noise added training images & 10 test images (MSE criteria)

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Variance	10%	20%	30%	40%	50%	60%	70%	80%	90%	Average
img64	909.55	1748.22	2452.59	2734.47	3828.18	4074.30	4691.99	4568.32	5436.86	3382.72
img128	1450.47	2418.05	3142.61	3914.83	4552.98	4990.00	5520.60	6044.87	6033.31	4229.75
img192	1090.06	1871.08	2583.58	3195.85	4127.17	4353.75	5200.18	5635.15	5783.14	3760.00
airplane	997.83	1774.03	2420.66	3051.72	3615.02	4177.64	4618.51	5091.66	5506.52	3472.62
Baboon	1381.58	2371.85	3169.12	3879.83	4476.66	5020.10	5534.81	5859.28	6181.43	4208.30
Boat	1098.32	1973.75	2753.08	3395.05	3999.15	4544.46	5011.39	5449.46	5853.64	3786.48
Bridge	1272.02	2148.68	2923.09	3574.60	4134.96	4707.06	5118.28	5523.57	5890.17	3921.38
Camera	1195.71	2083.94	2911.73	3550.43	4140.22	4666.16	5012.47	5522.32	5975.54	3895.39
goldhill	1078.75	1988.35	2741.33	3418.36	4008.90	4530.13	4973.23	5402.85	5803.29	3771.69
Lena	1076.21	1976.67	2747.91	3426.39	4005.04	4514.01	5031.98	5421.99	5847.27	3783.05
Parrots	1145.40	2015.13	2779.76	3403.56	4069.62	4608.53	5019.09	5412.01	5901.68	3817.20
peppers	1056.29	1928.59	2674.93	3355.15	3923.60	4388.27	4888.37	5276.31	5718.97	3690.05
Reptile	1186.30	2157.44	2974.18	3692.85	4237.71	4855.69	5285.85	5694.84	6097.56	4020.27
Average	1149.12	2035.06	2790.35	3430.24	4086.09	4571.55	5069.75	5454.05	5848.41	

Table 4.21. Comparison of the average values of Median Filter (3x3) and selected filter results for each Gaussian Noise variance (MSE criteria)

(These values are taken from the Table 4.14 & 4.15)

(The training images are portions of the Lena, Mandrill and Flowers images)

	Our Method	Median Filter
10%	435.98	1149.12
20%	657.77	2035.06
30%	869.11	2790.35
40%	1050.61	3430.24
50%	1257.65	4086.09
60%	1432.35	4571.55
70%	1614.68	5069.75
80%	1775.14	5454.05
90%	1934.01	5848.41

Table 4.22. Comparison of the average values of Median Filter (3x3) and selected filter results for each image (MSE criteria)

(These values are taken from the Table 4.14 & 4.15)

(The training images are portions of the Lena, Mandrill and Flowers images)

	Our Method	Median Filter
img64	1026.46	3382.72
img128	1542.85	4229.75
img192	1061.56	3760.00
airplane	1070.35	3472.62
baboon	1527.03	4208.30
Boat	1168.55	3786.48
Bridge	1409.56	3921.38
Camera	1345.01	3895.39
goldhill	1110.90	3771.69
Lena	1104.35	3783.05
Parrots	1159.43	3817.20
peppers	1104.59	3690.05
Reptile	1297.67	4020.27

5. CONCLUSION

In this thesis, an evolutionary approach is used to generate a noise filter which is based on soft mathematical morphology. The generated filter is best suitable for a level of salt & pepper noise, but it can also be classified as a generalized filter, too; because it can be used for every different image having different shapes and tones.

The aim in this study is to fulfill two goals. The first goal is to minimize the level of noise; the second one is to preserve the details. Both of them are achieved by using an appropriate representation which allows multi-stage filters with a wide range of structuring elements and by combining four different objectives in the fitness function.

And, also in the first of the testing, we had proved that the GA and the fitness function we had used improve a soft morphological filter both visually and also by means of some most used image comparison criteria.

The first comparisons are done with the median filter which is the most known filter for impulsive noise suppression. The results show that the filter generated in this study outperforms the median filer significantly.

In the next part of the testing, we had generated filters for different levels of noise and compared those filters with the best filters known in the literature. The results showed that the method we had used helps us to generate filters that are best suitable for specific level of noise. Our filter didn't out perform all the filters in the literature, but it got a rank in the best 3 or 4 filters according to the MSE criterion.

Although our filter is specifically trained for impulsive noise, we had tested and shown the results of the effects of the generated filter on Gaussian Noise added images. And, the selected filter also outperformed the Median Filter significantly for all images and noise levels tested.

Within this study; we had seen the effectiveness of the soft morphological filters for impulsive noise problem on images and we had also seen the effect of Genetic Algorithms on generating a multi-stage filter. A new fitness function with different criterions is used and its effects on the generated filters are tested.

At this point, it can be accepted that this method has proved itself. It can be used to as a tool for impulsive noise suppression on images. The generated filters for different noise levels can be embedded into an application for the users to select. Or those filters can be embedded into an application which detects the level of noise on images and select the best one from the list of filters that our GA generated.

APPENDIX A: PERFORMANCE RANKING TABLES

Table A.1. Comparison of our approach to the other approaches in the literature based on the average rankings over all noisy data using MSE values for the Lena image

(Here, the filters of our method are used on the noise level that they are trained for)

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method	11	6	4	3	3	3	3	3	4	3	4,44	2,65
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	14	12	6	4	4	4	5	5	7	5	6,78	3,70
SMF - 3x3	13	13	10	10	10	10	10	10	9	10	10,56	1,42
PSM	6	4	3	6	9	9	9	9	13	9	7,56	3,09
SDROM	8	11	13	13	13	13	13	13	12	13	12,11	1,69
SDROMR	7	5	8	7	7	7	8	8	8	8	7,22	0,97
IRF	9	10	11	11	11	11	11	11	10	11	10,56	0,73
ACWM	5	9	12	12	12	12	12	12	11	12	10,78	2,39
ACWMR	3	3	7	8	6	6	6	6	6	7	5,67	1,66
CWM	12	14	14	14	14	14	14	14	14	14	13,78	0,67
YÜKSEL	10	7	5	5	5	5	4	4	3	4	5,33	2,06
RUSSO	4	8	9	9	8	8	7	7	5	6	7,22	1,72
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.2. Comparison of our approach to the other approaches in the literature based on the average rankings over all noisy data using MSE values for the Mandrill image *(Here, the filters of our method are used on the noise level that they are trained for)* *(The training images are portions of the Lena, Mandrill and Flowers images)*

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method	12	8	6	5	3	3	3	3	3	3	5,11	3,14
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	14	13	8	7	4	4	4	5	5	5	7,11	3,89
SMF - 3x3	13	12	12	11	10	9	9	9	8	11	10,33	1,73
PSM	5	4	4	4	8	8	8	8	13	8	6,89	2,98
SDROM	8	10	11	12	12	12	12	12	11	12	11,11	1,36
SDROMR	10	7	7	8	7	7	7	7	7	7	7,44	1,01
IRF	9	9	10	10	11	11	10	10	9	10	9,89	0,78
ACWM	7	6	9	9	9	10	11	11	10	9	9,11	1,69
ACWMR	6	5	5	6	6	5	5	4	4	4	5,11	0,78
CWM	11	14	14	14	14	13	13	13	12	13	13,11	1,05
YÜKSEL	3	3	3	3	5	6	6	6	6	6	4,56	1,51
RUSSO	4	11	13	13	13	14	14	14	14	14	12,22	3,23
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.3. Comparison of our approach to the other approaches in the literature based on the average rankings over all noisy data using MSE values for the Peppers image

(Here, the filters of our method are used on the noise level that they are trained for)

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method	6	4	3	3	3	3	3	3	4	3	3,56	1,01
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	13	6	5	4	4	4	4	5	7	5	5,78	2,91
SMF - 3x3	11	8	9	10	10	10	10	10	10	10	9,78	0,83
PSM	4	3	4	5	5	9	9	9	9	9	6,33	2,60
SDROM	9	12	13	13	13	13	13	13	13	13	12,44	1,33
SDROMR	8	7	7	7	7	8	8	8	8	8	7,56	0,53
IRF	7	11	10	11	11	11	11	11	11	11	10,44	1,33
ACWM	5	10	11	12	12	12	12	12	12	12	10,89	2,32
ACWMR	3	5	6	6	6	6	7	7	6	7	5,78	1,20
CWM	12	13	14	14	14	14	14	14	14	14	13,67	0,71
YÜKSEL	14	14	12	9	9	5	5	4	3	4	8,33	4,30
RUSSO	10	9	8	8	8	7	6	6	5	6	7,44	1,59
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.4. Average performance ranking of the filters, which are obtained by using the training instances with a given noise level, over all noisy data for the Lena image *(The training images are portions of the Lena, Mandrill and Flowers images)*

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
10%	11	12	17	17	17	17	17	16	15	17	15,44	2,35
20%	12	6	7	13	16	15	15	15	14	14	12,56	3,64
30%	14	8	5	7	11	14	14	13	11	13	10,78	3,38
40%	16	9	3	4	7	9	10	9	8	9	8,33	3,74
50%	19	16	9	5	4	7	8	8	9	8	9,44	4,93
60%	17	14	6	3	3	4	7	7	7	7	7,56	4,85
70%	21	20	15	10	8	5	5	6	6	6	10,67	6,40
80%	20	19	14	8	6	3	3	4	5	5	9,11	6,79
90%	22	22	16	15	9	8	4	3	4	4	11,44	7,55
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	18	17	10	6	5	6	9	10	13	10	10,44	4,72
SMF - 3x3	15	18	18	18	18	18	18	18	17	18	17,56	1,01
PSM	6	4	4	11	15	16	16	17	21	16	12,22	6,24
SDROM	8	15	21	21	21	21	21	21	20	21	18,78	4,49
SDROMR	7	5	12	12	13	12	13	14	16	15	11,56	3,43
IRF	9	13	19	19	19	19	19	19	18	19	17,11	3,62
ACWM	5	11	20	20	20	20	20	20	19	20	17,22	5,45
ACWMR	3	3	11	14	12	11	11	11	12	12	9,78	3,96
CWM	13	21	22	22	22	22	22	22	22	22	20,89	2,98
YÜKSEL	10	7	8	9	10	10	6	5	3	3	7,56	2,51
RUSSO	4	10	13	16	14	13	12	12	10	11	11,56	3,40
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.5. Average performance ranking of the filters, which are obtained by using the training instances with a given noise level, over all noisy data for the Mandrill image *(The training images are portions of the Lena, Mandrill and Flowers images)*

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
10%	12	13	17	16	16	16	16	15	14	15	15,00	1,66
20%	13	8	7	15	15	14	14	14	11	13	12,33	3,00
30%	14	12	6	10	12	13	13	12	10	12	11,33	2,40
40%	16	14	9	5	8	9	9	9	8	8	9,67	3,32
50%	19	18	12	8	4	7	7	7	7	7	9,89	5,30
60%	18	17	10	6	3	6	6	6	6	6	8,67	5,32
70%	21	21	14	13	7	4	5	5	5	5	10,56	6,93
80%	20	20	13	11	5	3	4	4	4	4	9,33	6,96
90%	22	22	15	14	9	5	3	3	3	3	10,67	7,89
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	17	16	11	9	6	8	8	10	12	10	10,78	3,70
SMF - 3x3	15	15	20	19	18	17	17	17	16	19	17,11	1,69
PSM	5	4	4	4	14	15	15	16	21	16	10,89	6,60
SDROM	8	10	19	20	20	20	20	20	19	20	17,33	4,77
SDROMR	10	7	8	12	13	12	12	13	15	14	11,33	2,55
IRF	9	9	18	18	19	19	18	18	17	18	16,11	4,08
ACWM	7	6	16	17	17	18	19	19	18	17	15,22	5,04
ACWMR	6	5	5	7	11	10	10	8	9	9	7,89	2,26
CWM	11	19	22	22	22	21	21	21	20	21	19,89	3,48
RUSSO	3	3	3	3	10	11	11	11	13	11	7,56	4,39
YÜKSEL	4	11	21	21	21	22	22	22	22	22	18,44	6,46
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.6. Average performance ranking of the filters, which are obtained by using the training instances with a given noise level, over all noisy data for the Peppers image *(The training images are portions of the Lena, Mandrill and Flowers images)*

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
10%	6	7	14	17	17	17	17	16	16	17	14,11	4,43
20%	10	4	4	10	13	15	15	15	14	15	11,11	4,48
30%	13	5	3	6	9	10	14	13	12	13	9,44	3,97
40%	15	6	5	4	5	7	10	9	8	9	7,67	3,39
50%	18	14	8	5	4	6	7	8	9	8	8,78	4,49
60%	17	9	6	3	3	3	5	7	7	7	6,67	4,42
70%	20	18	12	9	7	4	4	6	6	4	9,56	5,92
80%	21	21	15	11	8	5	3	3	5	3	10,22	7,24
90%	22	22	21	16	15	9	6	4	4	5	13,22	7,63
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	16	10	9	7	6	8	8	10	13	10	9,67	3,12
SMF - 3x3	12	12	16	18	18	18	18	18	18	18	16,44	2,60
PSM	4	3	7	8	10	16	16	17	17	16	10,89	5,71
SDROM	9	17	20	21	21	21	21	21	21	21	19,11	4,01
SDROMR	8	11	11	13	12	14	13	14	15	14	12,33	2,12
IRF	7	16	17	19	19	19	19	19	19	19	17,11	3,95
ACWM	5	15	18	20	20	20	20	20	20	20	17,56	5,00
ACWMR	3	8	10	12	11	12	12	12	11	12	10,11	2,98
CWM	14	19	22	22	22	22	22	22	22	22	20,78	2,73
YÜKSEL	19	20	19	15	16	11	9	5	3	6	13,00	6,30
RUSSO	11	13	13	14	14	13	11	11	10	11	12,22	1,48
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.7. Comparison of approaches for the Lena image using average ranking of that are obtained after training for each image with the given noise level

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
60%	13	10	4	3	3	3	4	4	4	4	5,33	3,61
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	14	12	6	4	4	4	5	5	7	5	6,78	3,70
SMF - 3x3	12	13	10	10	10	10	10	10	9	10	10,44	1,24
PSM	6	4	3	6	9	9	9	9	13	9	7,56	3,09
SDROM	8	11	13	13	13	13	13	13	12	13	12,11	1,69
SDROMR	7	5	8	7	7	7	8	8	8	8	7,22	0,97
IRF	9	9	11	11	11	11	11	11	10	11	10,44	0,88
ACWM	5	8	12	12	12	12	12	12	11	12	10,67	2,50
ACWMR	3	3	7	8	6	6	6	6	6	7	5,67	1,66
CWM	11	14	14	14	14	14	14	14	14	14	13,67	1,00
YÜKSEL	10	6	5	5	5	5	3	3	3	3	5,00	2,18
RUSSO	4	7	9	9	8	8	7	7	5	6	7,11	1,69
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.8. Comparison of approaches for the Mandrill image using average ranking of that are obtained after training for each image with the given noise level

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
60%	14	13	7	5	3	3	3	3	3	3	6,00	4,47
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	13	12	8	7	4	4	4	5	5	5	6,89	3,48
SMF - 3x3	12	11	12	11	10	9	9	9	8	11	10,11	1,45
PSM	5	4	4	4	8	8	8	8	13	8	6,89	2,98
SDROM	8	9	11	12	12	12	12	12	11	12	11,00	1,50
SDROMR	10	7	6	8	7	7	7	7	7	7	7,33	1,12
IRF	9	8	10	10	11	11	10	10	9	10	9,78	0,97
ACWM	7	6	9	9	9	10	11	11	10	9	9,11	1,69
ACWMR	6	5	5	6	6	5	5	4	4	4	5,11	0,78
CWM	11	14	14	14	14	13	13	13	12	13	13,11	1,05
RUSSO	3	3	3	3	5	6	6	6	6	6	4,56	1,51
YÜKSEL	4	10	13	13	13	14	14	14	14	14	12,11	3,30
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

Table A.9. Comparison of approaches for the Peppers image using average ranking of that are obtained after training for each image with the given noise level

(The training images are portions of the Lena, Mandrill and Flowers images)

Noise Level	10	20	30	40	50	60	70	80	90	Rank	Average	STDEV
Our Method												
60%	13	5	3	3	3	3	3	4	4	4	4,56	3,24
AIF	1	1	1	1	1	1	1	1	1	1	1,00	0,00
IMF	12	6	5	4	4	4	4	5	7	5	5,67	2,60
SMF - 3x3	10	8	9	10	10	10	10	10	10	10	9,67	0,71
PSM	4	3	4	5	5	9	9	9	9	9	6,33	2,60
SDROM	8	12	13	13	13	13	13	13	13	13	12,33	1,66
SDROMR	7	7	7	7	7	8	8	8	8	8	7,44	0,53
IRF	6	11	10	11	11	11	11	11	11	11	10,33	1,66
ACWM	5	10	11	12	12	12	12	12	12	12	10,89	2,32
ACWMR	3	4	6	6	6	6	7	7	6	7	5,67	1,32
CWM	11	13	14	14	14	14	14	14	14	14	13,56	1,01
YÜKSEL	14	14	12	9	9	5	5	3	3	3	8,22	4,44
RUSSO	9	9	8	8	8	7	6	6	5	6	7,33	1,41
HAF	2	2	2	2	2	2	2	2	2	2	2,00	0,00

REFERENCES

1. Pitas, I. and A. N. Venetsanopoulos, Nonlinear Digital Filters - Principles and Applications, Boston: Kluwer Academic Publisher, 1990.

2. Bovik, A., Handbook of Image and Video Processing, New York: Academic Press, 2000.

3. Kontorovich, V. and V. Lyandres, “Impulsive noise: A nontraditional approach”, Signal Processing, 51, pp. 121-132, 1996.

4. Harvey, N. R. and S. Marshall, “Using genetic algorithms in the design of morphological filters”, In J. Serra, & Pierre S., Mathematical Morphology and Its Application to Image Processing, France: Kluwer Academic Publishers, pp. 53-59, 1994.

5. Romulos, T., B. Monica, B. Yuan, L. Olivier and B. Pierre, ”Adaptive filtering using morphological operators and genetic algorithms”, In Proceedings of the 6th International Conference on Signal Processing, Beijing, China, pp. 853-857, 2002.

6. Tongslieng, S., L. Soiigtslo and Z. Xiaodong, “Optimal design of morphological filters based on adaptive immune algorithm”, In Proceedings of the 7th International Conference on Signal Processing, Beijing, China, pp. 1064-1067, 2004.

7. Kraft, P., S. Marshall, J.J. Soraghan and N.R. Harvey, “Parallel genetic algorithms for optimizing morphological filters”, In Proceedings of the Fifth International onference on Image Processing and Its Applications, Edinburgh, UK, pp. 762-766, 1995.

8. Harvey, N. R. and S. Marshall, “The design of different classes of morphological filter using genetic algorithms”, In Proceedings of the Fifth International Conference on Image Processing and Its Applications, Edinburgh, UK, pp. 227-231, 1995.

9. Koivisto, P., H. Huttunen and P. Kuosmanen, ”Optimal soft morphological filtering

under breakdown probability constraints", In Proceedings of the International Symposium on Circuits and Systems, Atlanta, pp. 112-115, 1996.

10. Koivisto, P., H. Huttunen and P. Kuosmanen, ”Breakdown points and optimal soft morphological filtering", In Proceedings of 7th IEEE Signal Processing Workshop, Loen, Norway, pp. 490-493, 1996.

11. Chunhui, Z., S. Shenghe and H. Junying, "Optimization of soft morphological filters by simulated annealing", In Proceedings of the 5th International Conference on Signal Processing, Beijing, China, pp. 494-498, 2000.

12. Chunhui, Z., "Optimization design of soft morphological filters based on improving genetic algorithm", In Proceedings of the International Conference on Neural Networks & Signal Processing, Nanjing, China, pp. 491-494, 2003.

13. Chun-hui, Z., L. Gang-jian and N. Hai-chun, "Optimization of soft morphological filter based on tabu search", In Proceedings of the First International Conference on Innovative Computing, Information and Control, Beijing, China, pp. 611-614, 2006.

14. Zhao, C. and W. Zhang, "Using genetic algorithm optimizing stack filters based on MMSE criterion", Image and Vision Computing, 23, pp. 853–860, 2005

15. Jelodar, M. S., S.M. Fakhraie and M.N. Ahmadabadi, "Two-stage morphological filter design using genetic algorithm", In Proceedings of the IEEE International Conference on Engineering of Intelligent Systems, Islamabad, Pakistan, pp. 129-133, 2006.

16. Hough, K. and S. Marshall, "Soft morphological filters applied to the removal of noise from CCTV footage", In Proceedings of the IEE International Symposium on Imaging for Crime Detection and Prevention, London, UK, pp. 61-66, 2005.

17. Harvey, N. R. and S. Marshall, "Restoration of archive film material using multi-dimensional soft morphological filters", In Proceedings of the IEEE-EURASIP

Workshop on Nonlinear Signal and Image Processing, Antalya, Turkey, pp. 811-815, 1999.

18. Hamid, M. S., N.R. Harvey and S. Marshall, “Genetic algorithm optimization of multidimensional grayscale soft morphological filters with applications in film archive restoration”, IEEE Transactions on Circuits and Systems for Video Technology, 13(5), pp. 406-416, 2003.

19. Lin, T. N. and K.J. Chan, “Adaptive-hierarchical filtering approach for noise removal”, Displays, 29, pp. 209–213, 2008.

20. Lin, T. C., “Progressive decision-based mean type filter for image noise suppression”, Computer Standards & Interfaces, 30, pp. 106–114, 2008.

21. Yuan, S. Q., Y.H. Tan and H.L. Sun, ”Impulse noise removal by the difference-type noise detector and the cost function-type filter”, Signal Processing, 87, pp. 2417–2430, 2007.

22. Luo, W., “An efficient algorithm for the removal of impulse noise from corrupted Images”, Int. J. Electron. Commun. (AEÜ), 61, pp. 551 – 555, 2007.

23. Dang, D. and W. Luo, “Impulse noise removal utilizing second-order difference analysis”, Signal Processing, 87, pp. 2017–2025, 2007.

24. Yuan, S. Q. and Y.H. Tan, ”Impulse noise removal by a global–local noise detector and adaptive median filter”, Signal Processing, 86, pp. 2123–2128, 2006.

25. Hsia, S. C., “A fast efficient restoration algorithm for high-noise image filtering with adaptive approach”, J. Vis. Commun. Image R., 16, pp. 379–392, 2005.

26. Lukac, R., “Adaptive vector median filtering”, Pattern Recognition Letters, 24, pp. 1889–1899, 2003

27. Russo, F., “Impulse noise cancellation in image data using a two-output nonlinear filter”, Measurement, 36, pp. 205–213, 2004.

28. Cabrera, L. G., M.J.G. Salinas, P.L.L. Escamilla, J.M. Aroza, J.F.G. Lopera and R.R. Roldan, “Median-type filters with model-based preselection masks”, Image and Vision Computing, 14, pp. 741-752, 1996.

29. Çivicioğlu, P. and M. Alçı, “Impulsive Noise Suppression from Highly Distorted Images with Triangular Interpolants”, Int. J. Electron. Commun. (AEÜ), 58, pp. 311–318, 2004.

30. Alajlan, N., M. Kamel and E. Jernigan, ”Detail preserving impulsive noise removal”, Signal Processing: Image Communication, 19, pp. 993–1003, 2004.

31. Beşdok, E. and M.E. Yüksel, “Impulsive noise suppression from images with Jarque-Bera test based median filter”, Int. J. Electron. Commun. (AEÜ), 59, pp. 105 – 110, 2005.

32. Beşdok, E., P. Çivicioğlu and M. Alçı, “Using Anfis with circular polygons for impulsive noise suppression from highly distorted images”, Int. J. Electron. Commun. (AEU), 59, pp. 213 – 221, 2005.

33. Kang, C. C. and W.J. Wang, “Modified switching median filter with one more noise detector for impulse noise removal”, AEU - International Journal of Electronics and Communications, 63 (11), pp. 998-1004, 2009.

34. Zhang, J., “An efficient median filter based method for removing random-valued impulse noise”, Digital Signal Processing, Article in Press, Corrected Proof, 2009.

35. Li, Y., F. L. Chung and S. Wang, “A robust neuro-fuzzy network approach to impulse noise filtering for color images”, Applied Soft Computing, 8, pp. 872–884, 2008.

36. Çivicioğlu, P., “Using neighborhood-pixels-information and ANFIS for impulsive noise suppression”, Int. J. Electron. Commun. (AEÜ), 61, pp. 657 – 664, 2007.

37. Yüksel, M. E., “A median/ANFIS filter for efficient restoration of digital images corrupted by impulse noise”, Int. J. Electron. Commun. (AEÜ), 60, pp. 628 – 637, 2006.

38. Schulte, S., V. DeWitte, M. Nachtegael, D.V. DerWeken and E.E. Kerre, “Fuzzy random impulse noise reduction method”, Fuzzy Sets and Systems, 158, pp. 270 – 283, 2007.

39. Majhi, B. and P.K. Sa, ”FLANN-based adaptive threshold selection for detection of impulsive noise in images”, Int. J. Electron. Commun. (AEÜ), 61, pp. 478 – 484, 2007.

40. Yüksel, M. E., “A simple neuro-fuzzy method for improving the performances of impulse noise filters for digital images”, Int. J. Electron. Commun. (AEÜ), 59, pp. 463 – 472, 2005.

41. Yüksel, M. E. and A. Baştürk, “Efficient Removal of Impulse Noise from Highly Corrupted Digital Images by a SimpleNeuro-Fuzzy Operator”, Int. J. Electron. Commun. (AEÜ), 57(3), pp. 214–219, 2003.

42. Kong, H. and L. Guan, “A Neural Network Adaptive Filter for the Removal of Impulse Noise in Digital Images”, Neural Networks Letter, 9(3), pp. 373-378, 1996.

43. Beşdok, E., “A new method for impulsive noise suppression from highly distorted images by using Anfis”, Engineering Applications of Artificial Intelligence, 17, pp. 519–527, 2004.

44. Beşdok, E., P. Çivicioğlu and M. Alçı, “Using an adaptive neuro-fuzzy inference system-based interpolant for impulsive noise suppression from highly distorted images”, Fuzzy Sets and Systems, 150, pp. 525–543, 2005.

45. Türkmen, I., “Efficient impulse noise detection method with ANFIS for accurate image restoration”, AEU - International Journal of Electronics and Communications, Article in Press, Corrected Proof, 2010.

46. Kaliraja, G. and S. Baskarb, “An efficient approach for the removal of impulse noise from the corrupted image using neural network based impulse detector”, Image and Vision Computing, 28 (3), pp. 458-466, 2009.

47. Camarena, J. G., V. Gregori, S. Morillas and A. Sapena, ”Two-step fuzzy logic-based method for impulse noise detection in colour images”, Pattern Recognition Letters, Article in Press, Corrected Proof, 2010.

48. Tukey, J. W., ”Nonlinear (nonsuperposable) methods for smoothing data”, Congress Records EASCON’74, 673, 1974.

49. Wang, Z. and D. Zhang, ”Progressive switching median filter for the removal of impulse noise from highly corrupted images”, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, 46 (1), pp. 78–80, 1999.

50. Abreu, E., M. Lightstone, S.K. Mitra and K. Arakawa, ”A new efficient approach for the removal of impulse noise from highly corrupted images”, IEEE Transactions on Image Processing 5 (6), pp. 1012–1025, 1996.

51. Chen, T. and H.R. Wu, ”A new class of median based impulse rejecting filters”, IEEE International Conference on Image Processing 1, pp. 916–919, 2000.

52. Chen, T., and H.R. Wu, ”Adaptive impulse detection using center weighted median filters”, IEEE Signal Processing Letters 8 (1), pp. 1–3, 2001.

53. Muneyasu, M., N. Nishi and T. Hinamoto, ”A new adaptive center weighted median filter using counter propagation networks”, Journal of the Franklin Institute 337, pp. 631–639, 2000.

54. Russo, F. and G. Ramponi, ”A fuzzy filter for images corrupted by impulse noise”, IEEE Signal Processing Letters 6 (3), pp. 168–170, 1996.

55. Wang, J. H. and W.J. Liu, L.D. Lin, “Histogram-based fuzzy filter for image restoration”, IEEE Transactions on Systems, Man, and Cybernetics, Part B 32 (2), pp. 230–238, 2002.

56. Matheron, G., Random sets and integral geometry. New York: Wiley, 1975.

57. Serra, J., Image analysis and mathematical morphology, Vol. 1. London: Academic Press, 1982.

58. Serra, J., Introduction to mathematical morphology, Comp. Vision Graph. Image Proc., 35(3), pp. 283-305, 1986.

59. Koskinen, L., J. Astola and Y. Neuvo, ”Soft morphological filters”, In Proceedings of the SPIE Symp, Image Algebra and Morphological Image Processing, San Diego, CA, pp. 262-270, 1991.

60. Kuosmanen, P. and J. Astola, “Soft morphological filtering”, Journal Mathematical Imag. Vision, 5(3), pp. 231-262, 1995.

61. Holland, J. H., Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press, 1975.

62. Goldberg, D. E., Genetic algorithms in search, optimization, and machine learning. New York: Addison-Wesley, 1989.

63. Ozcan, E. and C. K. Mohan, “Partial shape matching using genetic algorithms”, Pattern Recognition Letters, 18, pp. 987-992, 1997.

64. Ozcan, E., “Memetic algorithms for nurse rostering”, Lecture Notes in Computer Science, 3733, pp. 482-492, 2005.

65. Ozcan, E. and E. Onbasioglu, “Memetic algorithms for parallel code optimization”, International Journal of Parallel Programming, 35(1), 33-61, 2007.

66. Kuosmanen, P., P. Koivisto, H. Huttunen and J. Astola, ”Shape preservation criteria and optimal soft morphological filtering”, J. Math. Imag. Vision, 5(4), pp. 319-335, 1995.