Düzce University Journal of Science & Technology, 6 (2018) 962-982

Düzce University
Journal of Science & Technology

Research Article

Image Denoising with Modified Grey Wolf Optimizer

Hüseyin Avni ARDAÇ [a,*], Pakize ERDOĞMUŞ [b]

[a] *Department of Computer Engineering, Institute of Science, Düzce University, Düzce, TURKEY*
[b] *Department of Computer Engineering, Faculty of Engineering, Düzce University, Düzce, TURKEY*
** Corresponding author's e-mail address: huseyinavniardac@duzce.edu.tr*

ABSTRACT

In this study, image denoising has been realized with with the one of the recent Nature-Inspired optimization algorithms, Grey Wolf Optimizer(GWO). GWO is one of the recent most studied continous optimization algorithm which performs better than the other algorithms. In this study, ten test images have been selected and gaussian noise has been added with some variance values. After the noisy images have been attained, these noisy images have been filtered with convulation in spatial domain. Filter coefficents have been trained with GWO, Modified Grey Wolf Optimizer(MGWO) and Genetic Algorithm(GA). Weiner filtering is also applied on the images for image denosing. The results show that Weiner Filter outperforms GWO trained filters on most of the images. MGWO performance is better then GWO and the results show that MGWO can also be used as an alternative method for image denoising. In the future studies, adaptive MGWO can be enhanced for much more succesfull image denoising process.

Keywords: *GWO, MGWO, GA, Image Denoising.*

Düzenlenmiş Gri Kurt Optimizasyon Algoritması ile Gürültü Temizleme

ÖZET

Bu çalışmada, yakın zamanda doğadan esinlenen optimizasyon algoritmalarından biri olan Gri Kurt Optimizasyonu(GWO) ile görüntülerdeki gürültülerin temizlenmesi gerçekleştirilmiştir. GWO, diğer algoritmalardan daha iyi performans gösteren, son zamanlarda en çok çalışılan sürekli optimizasyon algoritmasından biridir. Bu çalışmada on test görüntüsü seçilmiş ve bazı varyans değerleri ile gauss gürültüsü eklenmiştir. Gürültülü görüntüler elde edildikten sonra, bu gürültülü görüntüler uzamsal alanda konvülasyon ile filtrelenmiştir. Filtre katsayıları GWO, Düzenlenmiş Gri Kurt Optimizasyonu (MGWO) ve Genetik Algoritma (GA) ile eğitilmiştir. Elde edilen sonuçlara göre Weiner filter çoğu resimde daha başarılı sonuçlar vermiştir. MGWO'nun performansı GWO'dan daha iyidir ve sonuçlar MGWO'nun gürültü gidermede alternatif bir metot olarak kullanılabileceğini göstermiştir. Gelecekteki çalışmalarda daha başarılı gürültü temizleme işlemi için adaptif MGWO geliştirilebilir.

Anahtar Kelimeler: *GWO, MGWO, GA, Gürültü Temizleme.*

Received: 22/06/2018, Revised: 25/05/2018, Accepted: 27/06/2018

I. INTRODUCTION

In the recent years, with the exponential increasing of image and video transfer, image processing studies have also increased. Images are exposed to various types of noise distortion during the process of acquisition and transmission [1]. So, most of the studies in image processing are related with the image denoising.

There are different types of noised related with the transferring environment or acquisition. There are mainly two types of noises; additive and multiplicative noises. The primary reasons of noisy images are due to slow shutter speed, temperature, resistivity in solid state device and poor illumination [2]. Sampling and quantization errors also create noisy images. In order to enhance the quality of the images, image denoising is one of the pre-processing techniques. In most of the image processing studies, the artificial noisy images are created with some additive or multiplicative noise types. The most used noise types are White Noise, Gaussian Noise, Salt and Peppers, Red Noise and Speckle Noise. So, implementing an efficient noise removal technique is quite important in order to get every detail in the image processing applications.

In the literature, different types of methods have been used for image denoising. In the recent studies, deep convolutional neural networks have been used for this aim. Image denoising using the deep learning method has shown superior performance compared to conventional image denoising algorithms. In a study, improved by Lee at. al, convolutional neural network has been used for medical image denoising. They used 3000 chest radiograms for training their networks. In this study, they have introduced an image denoising technique based on a convolutional denoising autoencoder (CDAE) and evaluate clinical applications by comparing existing image denoising algorithms [3].

Conventional smoothening and sharpening filters are the most popular filters for noise reduction in digital images. While smoothening filters are not very well for higher noise and it can harm to detail parts like edges owing to smoothness factor, sharpening filters reduce not only the noise but all the small details [4].

Filtering for image processing can be realized both in spatial domain and frequency domain. For denoising in spatial domain, linear and non-linear filters[5,6,7] anisotropic diffusion[8,9] and total variation methods[10], dictionary learning method[11,12], bilateral and non-local means (NLM) filters[13], Neural Networks[14,15] and deep learning algorithms[16,17,18] have been used. For transform domain filtering, Wavelet based denoising[19, 20], fourier based denoising[21, 22], curvelet based denosinig[23, 24], threshold estimation[25] and shrinkage rules[26, 27] have been used.

Linear and non-linear filters are used for image denoising. While mean filters are linear, median filtering is non-linear. Digital convolution using linear filters are one of the most used techniques for image processing. Linear filter coefficients are selected in such a way which minimizes the noise. So, finding the optimum filter coefficients is an optimization problem. In recent years, since nature-inspired optimization algorithms are easy to implement and require no derivative process, they have been popular. Starting from Genetic Algorithm, several kinds of nature-inspired optimization algorithms have been improved such as Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony Optimization, Whale Optimization, Grey Wolf Optimizer.

Grey Wolf Optimizer is one of the recent nature-inspired optimization algorithms developed by Mirjalili[28]. GWO has been applied successfully for the solutions of lot of real life optimization problems from computer engineering to biomedical engineering such as clustering,[29] training multilayer percepterons[30], classification[31], medical image fusion[32] and system parameter estimation[33]. So for the simplicity and successful denoising process, linear filters trained by GWO are used in this study.

Section II presents image denoising. Section III presents Nature-Inspierd Algorithms. Section IV presents Weiner filtering. Section V outlines the image denoising with linear filters trained by GWO algorithm. The results have been presented in Sections VI.

II. IMAGE DENOISING

A digital image f(x,y) is a function of intensities in spatial coordinates. While grey level digital image is two dimensional matrix, colored images are three dimensional matrix in general. But these images are not always high quality. Especially medical images from CT, MRI and ultrasound have been used for medical diagnosis. So, all the details must be observed in the images. After the process of image acquisition or transmission, images are generally noisy. So, the images are pre-processed with some image denoising techniques. For this aim, several approaches have been improved in the literature. In order to evaluate the performances of the algorithms, some artificial noises are added in the studies.

The best way to test the effect of noise on a standard digital image is to add a gaussian white noise[34]. Gaussian noise is statistical noise, which the noise values are taken from as the Gaussian distribution. The probability function of Gaussian distribution is as given in Equation 1.

$$PG(z)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{1}$$

In this equation, PG(z) presents the probability of Gaussian distribution, z presents the grey level, σ presents the standard deviation and μ presents the mean value.

Image denosing methods are classified into two main categories called as spatial domain and frequency domain. These methods are given in Figure 1.

The main methods in spatial domain are linear and non-linear filters. Anisotropic diffusion filter and non-local means filters are main non-linear filters.

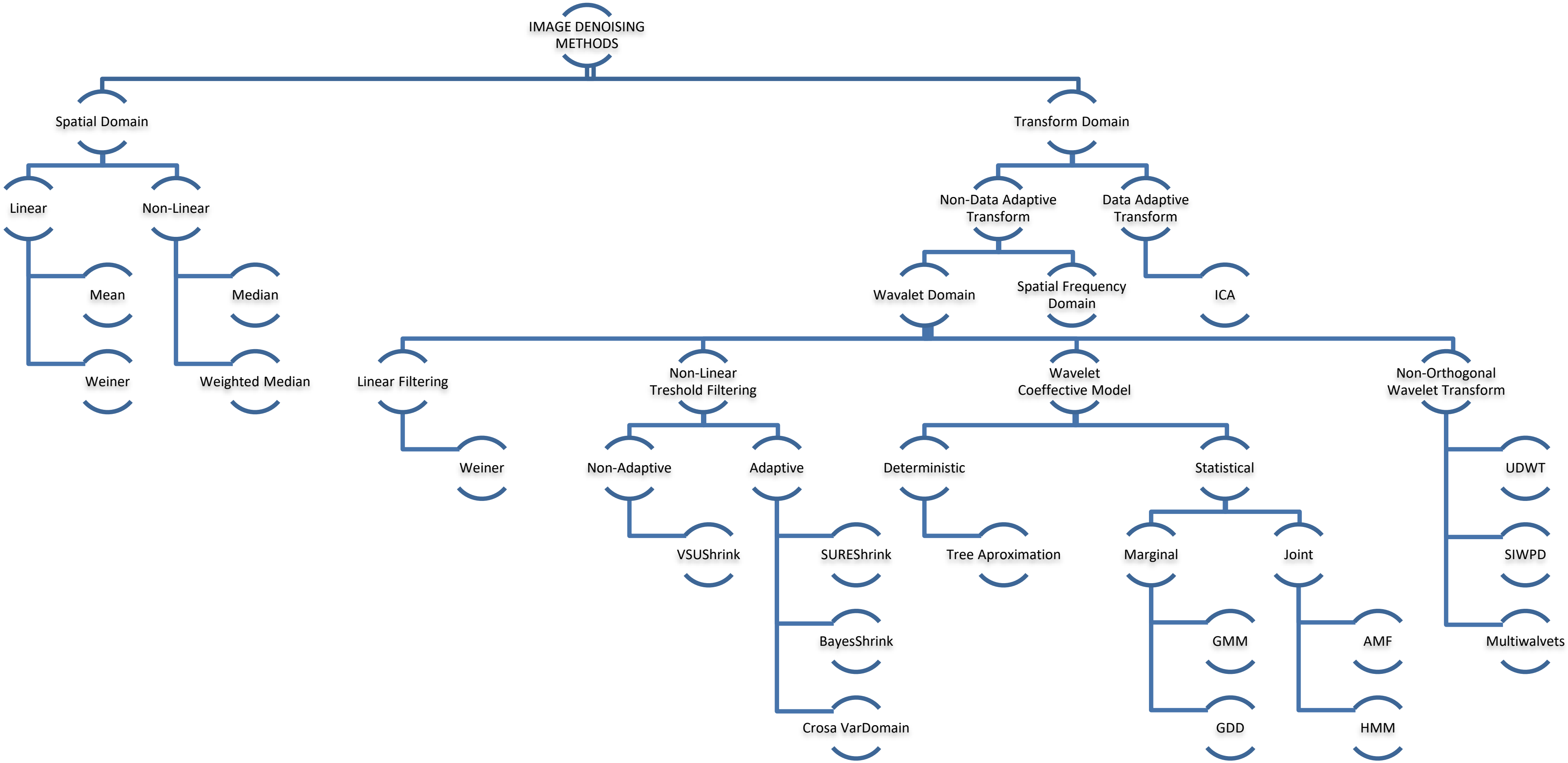


Figure 1. *Image denosing methods*

The main methods in frequency domain are based on curvelet, wavelet and contourlet transforms. In order to measure the performances of the denosing techniques, Means Square Error(MSE) and Peak Signal to Noise Ratio(PSNR) have been used. PSNR, is the ratio between the maximum possible power of a signal and the power of noise. Since signals have a very wide dynamic range, PSNR is usually expressed with logarithmic decibel scale.

MSE and PSNR values are calculated with the Equation 2 and 3.

$$MSE = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - J(j,i))^2 \quad (2)$$

$$PSNR = 10\log_{10}(\frac{{MAX_i}^2}{MSE}) \quad (3)$$

In the equation 2, I represents original image, J represents noisy image, M and N represent the size of the image. In the equation 3, MAX_i represents the maximum pixel.

III. NATURE-INSPIRED ALGORITHMS

In this study, image denoising has been realized with optimized filters. For this aim, GWO, MGWO and GA have been used. So GWO, MGWO and GA have been introduced below.

A. GREY WOLF OPTIMIZER

Grey Wolf Optimizer was firstly developed by Mirjalili in 2014[28] GWO is one of the nature inspired optimization algorithm developed recently. GWO simulates the hunting behavior of Grey Wolves for finding near optimal solution to optimization problems. GWO is one of the population based optimization algorithms. GWO was originally developed for the solution of continuous optimization problems. But it was also applied for the solutions of discrete optimization problems [35]. In order to increase the performance of the GWO, it was combined with other heuristic algorithms such as genetic operators [36], particle swarm [37], pattern search[38] and Levy Flight[39].

Grey Wolves live as social groups. There is a hierarchy among them. So they are classified as alpha, beta, delta and omega. The wolf has some responsibility according to its level in the hierarchy. And their hunting strategy has three main phases. In the first phase, grey wolves track the prey. In the second phase, they encircle the prey until the prey stops moving and at last attack towards the prey. GWO mimics this hunting strategy for finding the optimal solution of an optimization problem.

Prey represents the optimal solution while each wolf represents a solution. While real wolves change their position in such a way that they encircle and attack the prey, random initial solutions changes their position from one iteration to another in such a way that they converges optimal solution.

For the encircling behavior of wolves, some equations are proposed in the article [28].

There is a social hierarchy among the wolves. The solutions with the best eligibility are accepted as wolves named alpha (α), beta (β), delta (δ) and omega (ω) respectively. For the encircling behavior of wolves, the following equations are proposed:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{4}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{AD} \tag{5}$$

X represents the position of grey wolf; Xp represents the position of prey, $\vec{A}$ and $\vec{C}$ are the coefficients, t represents current iteration and t+1 represents the next iteration. $\vec{A}$ and $\vec{C}$ are given:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - a \tag{6}$$

$$\vec{C} = 2\vec{r}_2 \tag{7}$$

Even if the real wolves see their prey, since it has not got any idea about the optimum point(prey) in the solution space, the hunting behavior is simulated with the three best solution found among the wolves. So the following equations are proposed for hunting behavior:

$$\overrightarrow{D_\alpha} = \left|\vec{C}_1.\overrightarrow{X_\alpha} - \vec{X}\right| \tag{8}$$

$$\overrightarrow{D_\beta} = \left|\vec{C}_2.\overrightarrow{X_\beta} - \vec{X}\right| \tag{9}$$

$$\overrightarrow{D_\delta} = \left|\vec{C}_3.\overrightarrow{X_\delta} - \vec{X}\right| \tag{10}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}(\overrightarrow{D_\alpha}) \tag{11}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}(\overrightarrow{D_\beta}) \tag{12}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}(\overrightarrow{D_\delta}) \tag{13}$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \tag{14}$$

Attacking prey is simulated decreasing the value of a. The components of a is reduced linearly from 2 to 0 during the iterations. The random vectors r_1 and r_2 allow the wolves to reach any point in the 2D and 3D space. At this stage, a value is reduced and therefore the range of change of A is reduced. When A has random values in the range [-1,1], the next location of the search agent will be anywhere between the current location and the location of the prey. The pseudo-code of the GWO is given in Figure 2.

Initialize the grey wolf population $X_i (i = 1,2, \ldots, n)$
Initialize a, A and C
Calculate the fitnessof search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
While (t < Max number of iterations)
  For each search agent
    Update the position of the current search agent by equation
  End for
  Update a, A and C
  Calculate the fitness of all search agents
  Update X_α, X_β and X_δ
  t = t +1
end while
return X_α

Figure 2. *Pseudo code of the GWO algorithm.*

B. MODIFIED GREY WOLF OPTIMIZER

MGWO is also based on GWO[40]. The main difference is the decreasing process of parameter a. While in the original GWO, a is decreased linearly from 2 to 0 using the update equation as given in equation 15.

$$a = 2(1 - \frac{t}{T}) \quad (15)$$

where T indicates the maximum number of iterations and t is the current iteration. MGWO employs exponential function for the decay of a over the course of iterations, as given in Equation 16.

$$a = 2(1 - \frac{t^2}{T^2}) \quad (16)$$

The parameter changing both in GWO and MGWO can be seen in Figure 3.

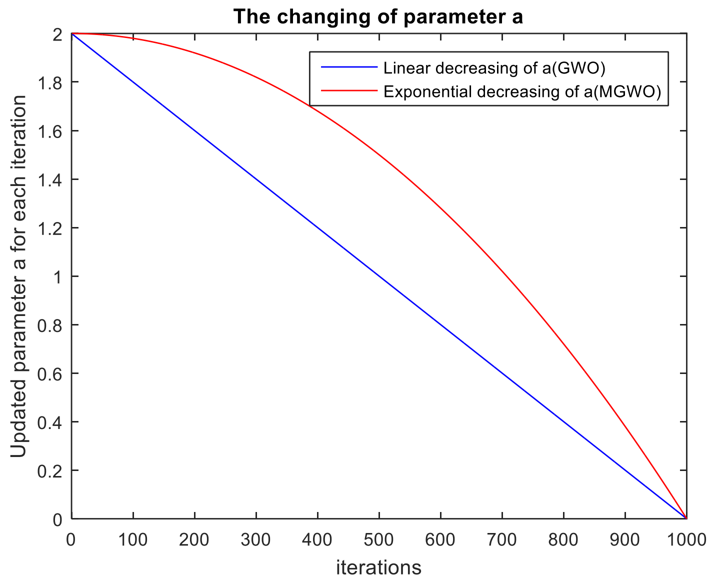


Figure 3. *Changing of parameter a in GWO and MGWO*

Using this exponential decay function, the numbers of iterations used for exploration and exploitation are 70% and 30%, respectively in MGWO[40].

C. GENETIC ALGORITHM

GA is developed by Holand[41]. GA is a population-based nature-inspired algorithm inspired by the theory of evolution. Survival of the fittest explains the Dawinian evolutionary theory. Since the optimization algorithms aim is to search of the fittest, evolutionary theory has been applied to the solution of the optimization problems very well. So, GA is one of the most successful optimization algorithms even today. In the algorithm, best properties are transferred from the generation to generation with crossover and elitism. Algorithm starts some initial random solutions of the optimization problem called individuals. GA uses some genetic operators such as crossover, mutation, and elitism in order to find the optimum solution. In each generation, fitness function values for each individual are calculated. The best individuals are selected for the next generation by some methods such as tournament selection or roulette wheel. After the selection, elitism, crossover, and mutation are applied to the population [42].

IV. WEINER FILTERING

The Wiener filter is stationary linear filter for additive noisy images. Weiner filter aims to find optimal MSE. Wiener filters are usually applied in the frequency domain. If it is assumed that x(m,n) noisy image and X(u,v) is the Discrete Fourier Transform (DFT) of the image x. The original image spectrum is estimated by taking the product of X(u,v) with the Wiener filter G(u,v) as given in the equation 17.

$$Y(u,v)=G(u,v)X(u,v) \tag{17}$$

The inverse DFT is then used to obtain the image estimate from its spectrum. The Wiener filter is defined as given below;

H(u,v): Fourier transform of point spread function(PSF)

P_s(u,v): Power spectrum of signal process, obtained by taking the fourier transform of noise autocorrelation

P_n(u,v): Power spectrum of noise process, obtained by taking the fourier transform of noise autocorrelation

The Wiener filter is:

$$G(u,v) = \frac{H(u,v)P_s(u,v)}{|H(u,v)|^2 P_s(u,v) + P_n(u,v)} \tag{18}$$

V. Image Denoising With Nature-Inspired Algorithms

In this study, image denoising has been realized with convolution. For this aim, Gaussian noises have been added with some variance and mean values. After this process, the noisy images have been convolved with the optimized filters trained by GWO, MGWO and GA. MSE value is used as fitness function for the algorithms. Since all the optimization algorithms aim to find minimum value of fitness function, the filter coefficients giving the minimum MSE, are accepted the best filter coefficients for these type of noise. Having found the coefficients, this filter is applied the noisy images and denoised the images. The images used for testing are given in Figure 4.

Figure 4. *Test images used for image denoising (fabric, kobi, lighthouse, onion, pears, gantrycrane, yellowlily, wagon, trailer, strawberries)*

Gaussian noise has been added to the pictures with some variance and mean values as given in the Table 1.

Table 1. *Means and Variance Values of the Gaussian Noises*

Gaussian Noise Statistical parameters	Test 1	Test 2	Test 3	Test 4
Mean	0	0.1	0	0
Variance	0.01	0.02	0.02	0.04
PSNR	20	16	17	14.5

The original images and noisy images after filtering have been compared with each other according to the PSNR value. Weiner filtering is also used for comparison in the tests.
Some of the results are given in the figures. Denoising images with GWO and Weiner can be seen in Figure 5 and Figure 6.

a) Original image(Trailer)

b) Gaussian Noisy image μ=0.1, σ=0.02

c) Denoised image with Weiner

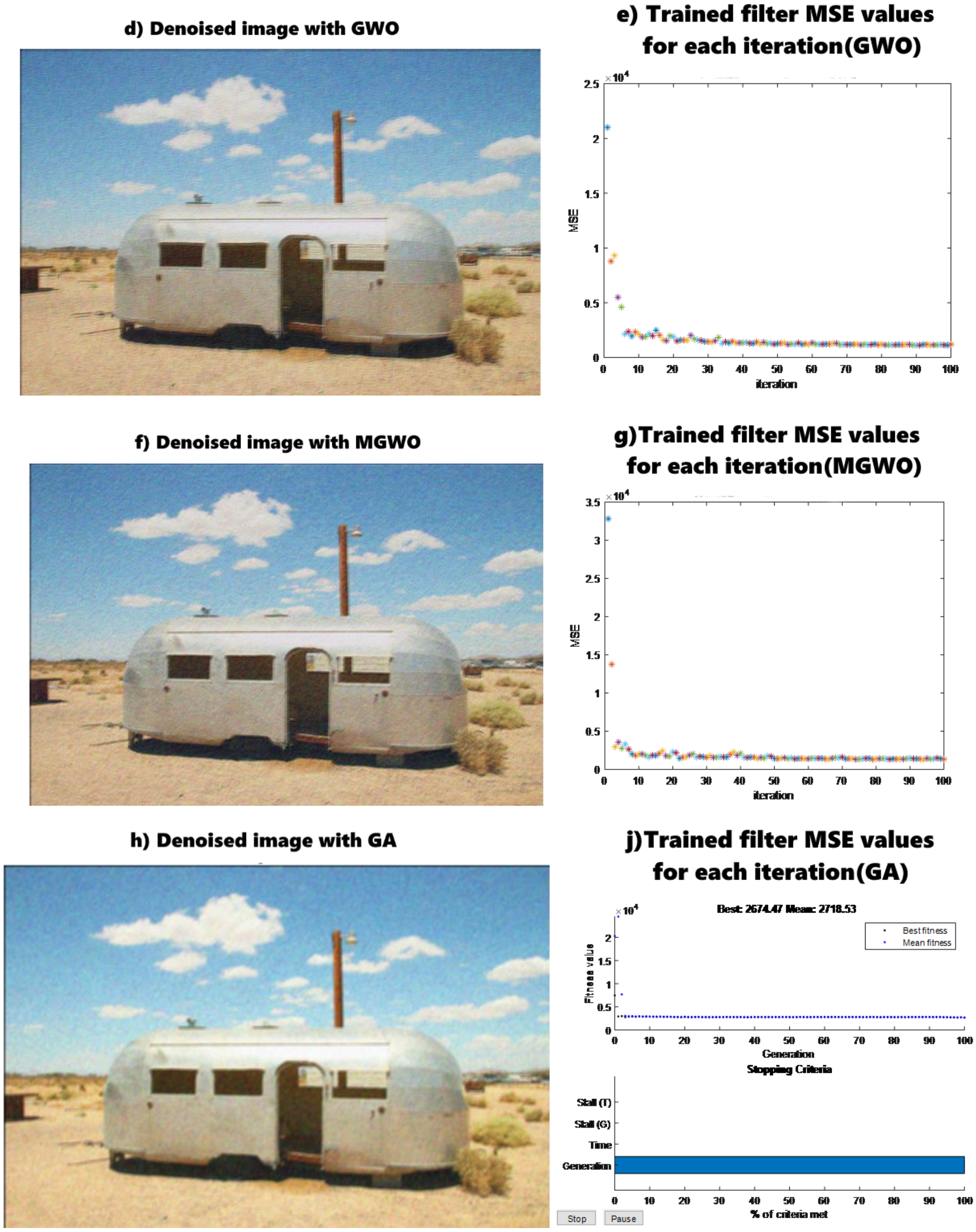
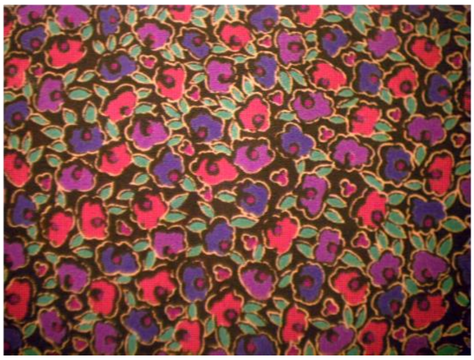
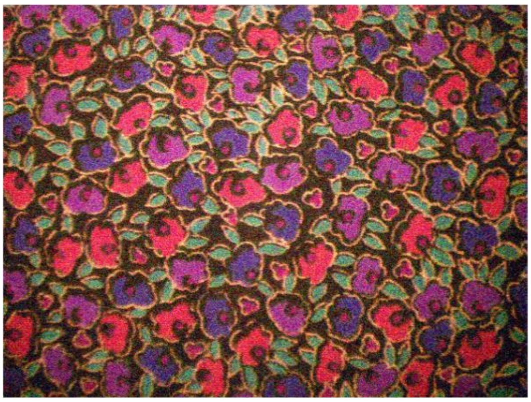


Figure 5. a) *Original image(Trailer),* ***b)*** *Gaussian Noisy image μ=0.1, σ=0.02,* ***c)*** *Denoised image with Weiner,* ***d)*** *Denoised image with GWO,* ***e)*** *Trained filter MSE values for each iteration(GWO),* ***f)*** *Denoised image with MGWO,* ***g)*** *Trained filter MSE values for each iteration(MGWO),* ***h)*** *Denoised image with GA,* ***j)*** *Trained filter MSE values for each iteration(GA)*

a) Original image(Fabric)



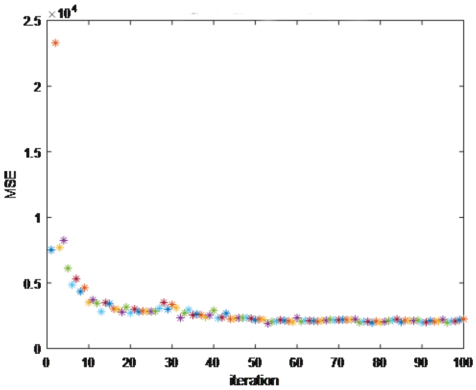
b) Gaussian Noisy image $\mu=0$, $\sigma=0.02$



c) Denoised image with Weiner

d) Denoised image with GWO

e) Trained filter MSE values for each iteration(GWO)



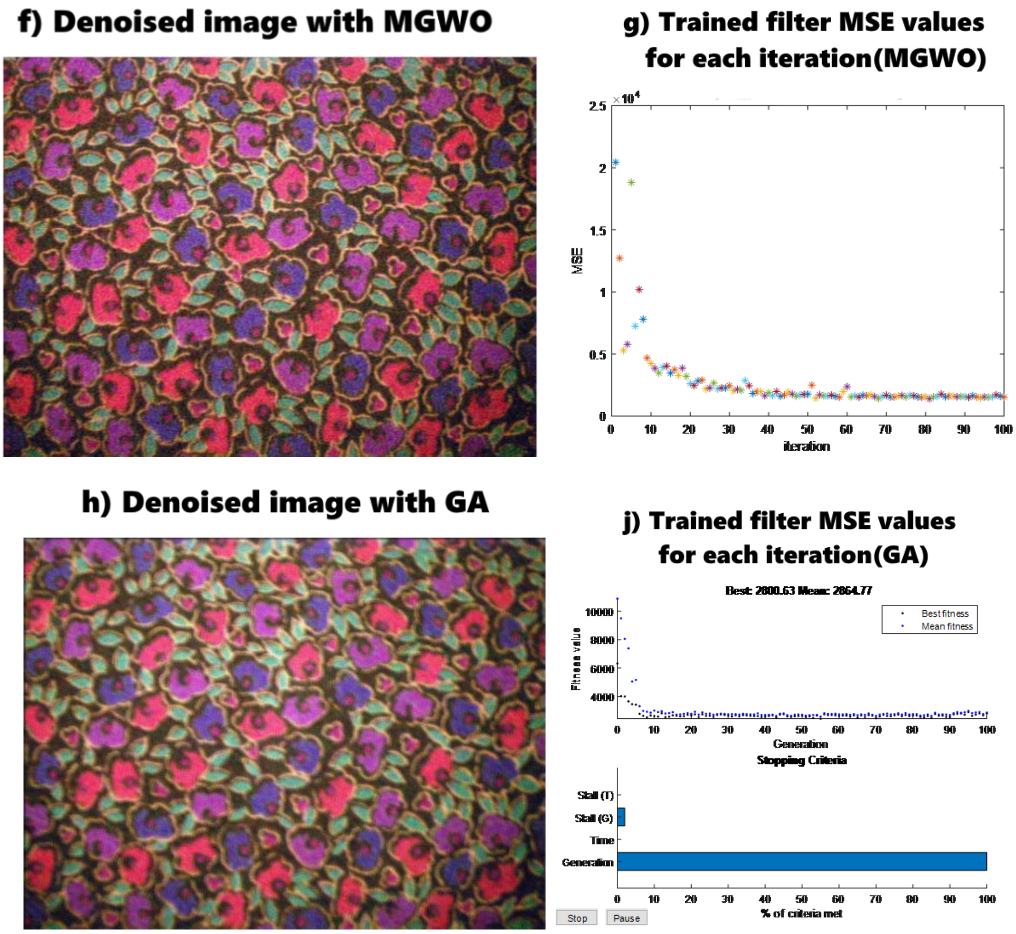


Figure 6. ***a)*** *Original image(Fabric),* ***b)*** *Gaussian Noisy image $\mu=0$, $\sigma=0.02$,* ***c)*** *Denoised image with Weiner,* ***d)*** *Denoised image with GWO,* ***e)*** *Trained filter MSE values for each iteration(GWO),* ***f)*** *Denoised image with MGWO,* ***g)*** *Trained filter MSE values for each iteration(MGWO),* ***h)*** *Denoised image with GA,* ***j)*** *Trained filter MSE values for each iteration(GA)*

The filtering results for each picture with some Gaussian noise for filter size N=3, 5 and 7 are given between the Table 2-13.

Table 2. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.01) (GWO)*

Image	Average Time(s)	GWO	Weiner	Average Time(s)	GWO	Weiner	Average Time(s)	GWO	Weiner
		N=3			N=5			N=7	
Fabric	134,3484	22,7897	25,8485	136,7641	21,8268	25,1783	139,3408	18,905	23,3692
Kobi	298,8725	25,8925	27,5211	310,0827	26,0859	30,0938	322,9824	25,1226	30,0788
Lighthouse	135,3147	24,2563	26,9584	135,8922	24,5908	27,8387	137,9614	23,01	27,4676
Onion	105,8386	24,8621	27,007	106,5224	24,8256	28,0234	106,6018	22,5069	27,2532
Pears	139,086	22,6484	27,2991	139,4954	20,8796	29,8569	141,5134	20,1853	30,3382
Gantrycrane	112,1146	23,0666	26,0325	112,9489	21,4982	25,9751	113,6594	21,1594	25,004
Yellowlily	282,4301	25,4763	27,4127	288,4157	25,317	29,2485	297,1742	23,4926	29,5293
Wagon	174,3174	23,1354	24,056	176,2602	20,1013	23,7345	180,1517	21,2028	23,2404
Trailer	165,8312	25,9632	27,0826	168,5925	26,0951	28,8726	171,9355	24,949	28,875
Strawberries	164,5891	24,3922	25,8506	167,1693	24,2702	25,3331	171,4862	21,9998	24,353

__Table 3.__ PSNR values after the removal of Gaussian Noise (M=0, Var=0.02) (GWO)

Image	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**
		N=3			**N=5**			**N=7**	
Fabric	134,0033	21,1163	23,6793	134,8141	20,26	23,9289	136,7654	18,0742	22,5535
Kobi	289,9725	24,548	24,6467	295,0976	24,2969	27,5801	305,9505	21,3766	28,1689
Lighthouse	136,8795	22,9722	24,1934	135,8335	22,8832	25,6435	138,2997	20,4745	25,6633
Onion	106,2051	23,25	24,2222	106,3686	22,9058	25,909	107,0079	22,0529	25,5759
Pears	138,2129	22,24	24,4312	139,1395	20,6046	27,4991	141,5461	20,2674	28,546
Gantrycrane	112,0751	21,2681	23,6268	112,8857	21,2397	24,228	113,585	19,7571	23,6222
Yellowlily	284,358	24,0018	24,4067	285,2345	23,9413	26,1841	293,3956	22,7801	26,6261
Wagon	174,2253	21,4967	22,3251	176,7826	21,5461	22,6439	179,88	20,9168	22,3588
Trailer	166,2067	24,1584	24,2934	169,2443	22,9496	26,6346	172,5073	23,3794	27,1294
Strawberries	164,9504	22,5999	23,4377	167,3169	20,7434	23,8186	170,7758	21,3649	23,2488

__Table 4.__ PSNR values after the removal of Gaussian Noise (M=0.1, Var=0.02) (GWO)

Image	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**
		N=3			**N=5**			**N=7**	
Fabric	133,6148	17,6692	18,5691	134,8184	16,963	18,6334	136,3902	15,3649	18,1664
Kobi	291,8498	**21,7013**	19,4668	298,2074	**21,129**	20,1074	307,028	**22,6996**	20,2171
Lighthouse	135,2092	19,671	19,7071	136,1868	19,2719	20,18	138,3968	18,8522	20,1958
Onion	105,6204	**19,1249**	18,8219	106,4527	19,3907	19,3532	107,0532	18,9647	19,4769
Pears	138,4479	**18,9895**	18,9198	138,5476	19,175	19,5851	139,3987	18,3398	19,755
Gantrycrane	112,1355	18,2828	18,7725	113,1537	18,3268	19,0309	113,9882	18,4821	18,8867
Yellowlily	277,9273	17,9588	18,5557	283,1165	17,9153	19,0708	291,2346	17,2095	19,1808
Wagon	174,2757	19,2671	18,667	174,4376	**19,2315**	18,7978	177,3628	**18,8392**	18,6954
Trailer	166,6848	**19,6589**	19,2105	168,5054	**21,6918**	19,7719	172,5877	**20,9479**	19,8896
Strawberries	165,1459	18,3858	18,6325	168,487	18,3677	18,8042	173,0544	17,3426	18,6236

__Table 5.__ PSNR values after the removal of Gaussian Noise (M=0, Var=0.04) (GWO)

Image	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**	**Average Time(s)**	**GWO**	**Weiner**
		N=3			**N=5**			**N=7**	
Fabric	133,69	19,6806	21,2806	134,8594	19,402	22,3234	136,9146	17,8912	21,4806
Kobi	293,1977	**22,6167**	21,8782	298,4864	22,7732	24,8535	310,3481	22,273	25,8169
Lighthouse	133,3527	21,1538	21,4046	134,1711	21,1107	23,1195	135,8882	20,691	23,429
Onion	105,7577	21,2133	21,4218	106,4979	20,4916	23,3452	106,9988	20,3228	23,6075
Pears	136,2395	20,5472	21,7475	137,7498	19,6898	25,0086	139,6628	19,6516	26,4309
Gantrycrane	112,2422	19,5637	21,1332	113,1742	19,1263	22,2679	114,7772	17,5506	22,0245
Yellowlily	281,2751	21,147	21,3783	291,4211	20,8967	23,0441	300,8309	19,7165	23,5452
Wagon	172,1187	19,4874	20,2994	174,6267	20,1943	21,2371	179,1735	19,5832	21,2369
Trailer	168,8631	**22,0243**	21,5876	168,5988	22,7112	24,1533	172,1153	22,6116	24,9774
Strawberries	166,4015	20,8136	20,889	169,3823	19,8732	21,8981	172,7733	19,8393	21,7585

Table 6. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.01) (MGWO)*

Image	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner
		N=3			N=5			N=7	
Fabric	133,3652	22,6776	25,8779	134,4606	21,6243	25,1912	136,4701	19,4578	23,362
Kobi	288,3694	26,2327	27,5357	293,6421	26,244	30,0932	322,5773	25,1201	30,0738
Lighthouse	139,8766	24,7474	26,9555	142,8566	25,6874	27,4555	143,6178	23,8591	27,4668
Onion	107,0948	24,75	27,0587	107,2631	24,2276	27,9538	107,464	23,1005	27,2461
Pears	143,6635	21,9508	27,2781	147,5703	20,9467	29,8432	149,3868	19,5866	30,3448
Gantrycrane	113,7784	22,7629	26,0213	114,512	20,0646	25,9625	115,3446	20,1667	24,9561
Yellowlily	333,843	25,7258	27,4097	318,723	24,463	29,2498	336,0282	24,6897	29,524
Wagon	177,3223	23,157	24,0509	174,0743	23,0607	23,7366	185,0308	21,8754	23,2344
Trailer	171,6666	25,5616	27,0794	169,0148	26,2144	28,8791	173,2772	25,1057	28,8947
Strawberries	177,9264	24,3379	25,8475	182,3927	24,1923	25,3363	191,8655	22,552	24,3491

Table 7. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.02) (MGWO)*

Image	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner
		N=3			N=5			N=7	
Fabric	133,45	20,3007	23,696	134,6831	21,1478	23,9249	136,597	19,3403	22,5441
Kobi	295,4934	24,3013	24,6551	334,0922	24,4338	27,5731	376,6728	24,3988	28,1636
Lighthouse	140,5744	22,8819	24,1728	142,5427	22,0226	25,6766	144,0474	22,0154	25,6406
Onion	107,0126	22,6813	24,2554	106,9324	23,1963	25,7296	107,7687	21,3641	25,6027
Pears	144,8482	21,8551	24,4463	146,7596	21,2223	27,5098	149,0278	20,6226	28,5555
Gantrycrane	114,004	21,2233	23,6167	114,6656	21,3793	24,2452	116,2253	19,7314	23,6059
Yellowlily	350,425	23,9279	24,4156	361,9068	23,1757	26,1778	368,8394	22,3431	26,6175
Wagon	177,8133	21,2786	22,3173	175,8125	21,5711	22,646	178,861	20,718	22,3642
Trailer	167,1213	23,9916	24,2887	169,4697	24,2387	26,6225	173,2695	24,5011	27,1351
Strawberries	185,0589	22,6067	23,4377	188,0555	21,3388	23,8174	195,7107	20,7909	23,255

Table 8. *PSNR values after the removal of Gaussian Noise (M=0.1, Var=0.02) (MGWO)*

Image	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner
		N=3			N=5			N=7	
Fabric	133,2294	17,3245	18,5583	134,5593	17,1341	18,6386	136,448	17,4353	18,1771
Kobi	346,6606	**21,8546**	19,4585	364,4716	**22,9807**	20,1146	375,2674	**23,2187**	20,2248
Lighthouse	141,0673	19,0155	19,7162	144,0296	19,4059	20,1787	143,1814	20,1272	20,1919
Onion	107,1561	18,7679	18,8256	107,2315	19,263	19,3507	107,8479	19,1781	19,4238
Pears	144,9998	**19,0738**	18,9481	146,9665	19,5018	19,5947	149,3479	19,2801	19,7651
Gantrycrane	113,4972	18,268	18,7959	120,0271	18,9685	19,0164	124,0573	17,8454	18,8682
Yellowlily	346,5991	18,2467	18,5554	354,6791	17,9728	19,0699	330,526	17,5983	19,1945
Wagon	173,068	**18,8002**	18,661	175,3768	**19,4151**	18,8029	179,5636	18,1616	18,6961
Trailer	166,8805	**20,3935**	19,2181	178,0588	**22,0685**	19,7794	194,273	**22,0921**	19,8665
Strawberries	188,1755	18,4277	18,6363	186,4083	**18,9737**	18,7938	191,2795	**18,8412**	18,6212

Table 9. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.04) (MGWO)*

Image	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner	Average Time(s)	MGWO	Weiner
		N=3			N=5			N=7	
Fabric	133,763	19,7531	21,2637	135,0499	19,9751	22,3087	136,937	18,4578	21,4928
Kobi	343,7036	**22,6018**	21,8776	357,6147	23,164	24,8571	362,3382	22,2236	25,8095
Lighthouse	137,9777	20,7075	21,3775	143,1658	21,3829	23,1336	144,8573	19,2344	23,423
Onion	106,7463	21,2299	21,4491	107,4155	20,3476	23,4280	107,3432	20,5665	23,423
Pears	145,0422	21,047	21,7544	146,534	20,4281	25,0326	148,2611	20,0933	26,3948
Gantrycrane	125,0575	19,4684	21,0967	117,9371	19,5383	22,2393	119,9955	18,4053	22,0245
Yellowlily	314,4412	21,1556	21,3703	315,3832	21,2302	23,0558	330,7387	20,6115	23,5309
Wagon	173,3476	19,4402	20,3103	181,2612	20,1373	21,2407	182,1381	19,4515	21,2262
Trailer	185,0538	21,2512	21,5839	188,3792	22,8752	24,1556	185,3137	23,3709	24,9662
Strawberries	185,5828	20,8149	20,8902	188,2174	19,9501	21,8918	191,9452	20,6462	21,7458

Table 10. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.01) (GA)*

Image	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner
		N=3			N=5			N=7	
Fabric	41,4988	21,4671	25,8714	42,8663	17,6124	25,1866	42,9729	17,2521	23,3811
Kobi	261,3749	24,8927	27,5251	261,8328	24,4208	30,0871	287,8099	22,2683	30,0853
Lighthouse	47,0023	23,08	26,9857	48,7433	21,0321	27,849	49,7034	19,9094	27,4467
Onion	9,4753	23,638	27,0099	9,1977	21,5099	28,0011	10,3619	19,9583	27,2224
Pears	45,7399	21,3022	27,2736	40,1252	20,0464	29,8482	50,1232	19,593	30,3355
Gantrycrane	21,8319	20,9673	26,0305	22,2179	18,4006	25,9485	15,9433	16,3008	24,9619
Yellowlily	283,8835	26,0409	27,4203	299,4907	24,777	29,2569	322,1625	23,0057	29,519
Wagon	104,6458	22,7733	24,0528	97,8151	20,0146	23,7497	126,7426	18,5334	23,2392
Trailer	68,0519	25,0693	27,0802	68,9768	24,6449	28,8666	72,7132	22,7121	28,8871
Strawberries	66,7125	23,452	25,8352	68,4861	20,2545	25,3404	71,4942	19,4331	24,3452

Table 11. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.02) (GA)*

Image	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner
		N=3			N=5			N=7	
Fabric	41,7937	19,8628	23,6755	41,4299	18,3043	23,9356	43,8131	17,0635	22,5511
Kobi	240,5879	23,5142	24,6563	253,5413	23,8201	27,5652	271,4847	22,4243	28,1646
Lighthouse	41,5229	22,4101	24,1849	46,1619	21,2402	25,6808	44,6024	20,1455	25,6775
Onion	9,3217	22,141	24,2179	11,3989	21,061	25,8049	11,7957	19,1099	25,5983
Pears	47,9712	20,9853	24,4465	48,5342	20,3401	27,4976	55,5888	19,812	28,5856
Gantrycrane	21,0333	20,3751	23,6024	15,7716	17,2587	24,2391	16,0895	15,28	23,599
Yellowlily	305,4893	23,373	24,4123	307,9756	22,9819	26,1868	326,069	21,9797	26,625
Wagon	127,2575	20,9927	22,3277	77,2082	19,6099	22,652	78,2904	18,4278	22,366
Trailer	55,9851	23,5533	24,2892	69,3534	23,8387	26,6478	73,5194	22,9182	27,1478
Strawberries	66,7316	22,206	23,4273	68,8344	20,706	23,8321	70,8182	18,8052	23,2586

Table 12. *PSNR values after the removal of Gaussian Noise (M=0.1, Var=0.02) (GA)*

Image	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner
		N=3			N=5			N=7	
Fabric	41,527	16,2775	18,5418	42,3557	15,762	18,6318	47,0664	15,1826	18,1754
Kobi	252,8218	**20,1933**	19,473	253,7768	19,5916	20,1008	245,1399	19,5916	20,1008
Lighthouse	45,9429	19,168	19,703	43,4029	18,9023	20,178	40,3719	18,0881	20,2030
Onion	9,1657	18,813	18,8205	9,137	18,8955	19,402	9,473	17,2161	19,448
Pears	57,936	17,9701	18,917	67,6669	17,5462	19,5972	61,6646	16,8875	19,7775
Gantrycrane	15,2802	16,6257	18,7942	21,825	15,7551	19,0341	22,9933	14,7169	18,8548
Yellowlily	284,7644	17,8891	18,5502	300,1836	17,7304	19,0693	308,9944	17,1731	19,1888
Wagon	73,889	**19,2232**	18,6674	75,4003	18,6499	18,7947	78,378	16,8814	18,7003
Trailer	67,572	**20,2523**	19,2193	69,3749	**21,1818**	19,7755	72,118	**20,1628**	19,8731
Strawberries	67,0096	18,2843	18,6351	69,2872	17,5227	18,8007	71,6449	16,5254	18,6127

Table 13. *PSNR values after the removal of Gaussian Noise (M=0, Var=0.04) (GA)*

Image	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner	Average Time(s)	GA	Weiner
		N=3			N=5			N=7	
Fabric	45,633	19,6431	21,2819	47,5375	17,911	22,3492	52,6115	16,0481	21,4589
Kobi	233,8753	21,3802	21,8714	266,5404	22,5575	24,8577	291,5235	21,6254	25,8223
Lighthouse	43,3984	20,967	21,4038	46,1647	20,4508	23,1113	47,0753	19,4435	23,4361
Onion	11,3878	18,8422	21,4998	9,0669	20,602	23,2443	11,6438	19,4913	23,5637
Pears	58,2026	21,5982	21,7264	61,7265	20,6919	25,0071	41,3623	20,0131	26,4396
Gantrycrane	22,0066	19,4617	21,1299	15,3294	17,0795	22,3126	16,0595	15,7672	22,0571
Yellowlily	272,4333	20,164	21,3724	299,5036	20,9121	23,0406	312,6632	19,7259	23,5433
Wagon	74,1945	19,7784	20,3144	76,0623	19,112	21,2304	79,7701	18,3017	21,2455
Trailer	67,959	21,5114	21,5812	69,4109	22,4146	24,1510	73,176	21,3379	24,97
Strawberries	67,8407	19,6057	20,8827	69,1379	19,4629	21,907	71,6522	19,0708	21,7321

VI. CONCLUSION

In this study, image denoising with Nature-Inspired Algorithms has been implemented. With this aim ten images have been selected and Gaussian noise with some mean and standart deviation has been added to the images. These noisy images have been used for denoising process. Denosing process has been realized with Weiner filtering, and GWO, MGWO and GA. Filter coefficents have been trained with GWO,MGWO and GA for N=3,5 and 7 square filter size. Results have been compared. According to the PSNR values, Weiner and MGWO have most succesfull results. After Weiner and MGWO, GWO is the third and GA has the worst results. As it can be seen from the tables, MGWO has better results for some images from Weiner. But MGWO needs some adaptation. If we adapt MGWO for image denoising for different type of images, results can be better.

VII. References

[1] D. K. Priya, B. B. Sam, S. Lavanya and A. P. Sajin, “A survey on medical image denoising using optimisation technique and classification,” *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1-6, 2017.

[2] D. Chowdhury, S. Gupta, D. Roy, D. Sarkar, C. C. Chattopadhyay and S. K. Das, “A quantum study on digital image noises and their in-depth clusterization,” *2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)*, pp. 1-7, 2017.

[3] D. Lee, S. Choi, H. J. Kim, “Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators*, vol 884, pp 97-104, 2018.

[4] M. Diwakar, M. Kumar, “A review on CT image noise and its denoising,*” Biomedical Signal Processing and Control*, vol. 42, pp. 73-88, 2018.

[5] B. Gupta and S. Singh Negi, “Image denoising with linear and non-linear filters: a review”, *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 2, 2013.

[6] P. K. Patidar, B. Singh and G. Bagaria, “Image filtering using linear and non linear filter for gaussian noise,” *International Journal of Computer Applications*, vol. 93, no. 8, 2014.

[7] V. R. Tripathi, “Image Denoising Using Non Linear Filters,” *International Journal of Simulation Systems, Science and Technology,*vol. 1, no. 1, 2011.

[8] F. Liu, J. Liu, “Anisotropic diffusion for image denoising based on diffusion tensors,” *Journal of Visual Communication and Image Representation*, vol 23, pp. 516-521, 2012.

[9] H. Kim and S. Kim, “Impulse-mowing anisotropic diffusion filter for image denoising,” *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2923-2927, 2014.

[10] L. Chato, S. Latifi and P. Kachroo, “Total variation denoising method to improve the detection process in IR images,” *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 441-447, 2017.

[11] Y. Li, S. Ding, Z. Li, X. Li and B. Tan, “Dictionary learning in the analysis sparse representation with optimization on Stiefel manifold,” *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1270-1274, 2017.

[12] J. Maggu, R. Hussein, A. Majumdar and R. Ward, “Impulse denoising via transform learning,” *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1250-1254, 2017.

[13] A. B. Said, R. Hadjidj, K. E. Melkemi and S. Foufou, “Multispectral image denoising with optimized vector non-local mean filter,” *Digital Signal Processing*, vol. 58, pp. 115-126, 2016.

[14] S. Bhattacharyya, P. Pal and S. Bhowmick, “Binary image denoising using a quantum multilayer self organizing neural network,” *Applied Soft Computing*, vol 24, pp. 717-729, 2014.

[15] C. Ferles, Y. Papanikolaou, K. J. Naidoo, “Denoising Autoencoder Self-Organizing Map (DASOM),” *Neural Networks*, vol. 105, pp. 112-131, 2018.

[16] A. Benou, R. Veksler, A. Friedman, T. Riklin Raviv, “Ensemble of expert deep neural networks for spatio-temporal denoising of contrast-enhanced MRI sequences,” *Medical Image Analysis*, vol. 42,2017,pp. 145-159, 2017.

[17] D. Lee, S. Choi, H. J. Kim, “Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators*, vol. 884, pp. 97-104, 2018.

[18] X. Kuang, X. Sui, Y. Liu, Q. Chen and G. GU, “Single infrared image optical noise removal using a deep convolutional neural network,” *in IEEE Photonics Journal*, vol. 10, no. 2, pp. 1-15, 2018.

[19] M. F. Fahmy and O. M. Fahmy, “A new image denoising technique using orthogonal complex wavelets,” *2018 35th National Radio Science Conference (NRSC)*, pp. 223-230, 2018.

[20] M. Kimlyk and S. Umnyashkin, “Image denoising using discrete wavelet transform and edge information,” *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1823-1825, 2018.

[21] T. Williams and R. Li, “An efficient hybrid fourier-wavelet neighborhood coefficient image denoising approach,” *In SoutheastCon*, pp. 1-4, 2016.

[22] J. Fan, H. Yi, L. Xu and T. Zhao, “A histogram-based denoising algorithm in a joint-fourier transform correlator for image recognition,” *2012 Symposium on Photonics and Optoelectronics*, pp. 1-3, 2012.

[23] S. K. Panigrahi, S. Gupta and P. K. Sahu, “Curvelet-based multiscale denoising using non-local means and guided image filter,” *in IET Image Processing*, vol. 12, no. 6, pp. 909-918, 6 2018.

[24] S. Zhen-gang and L. Qin-zi, “Pulmonary CT image denoising algorithm based on curvelet transform criterion,” *2017 7th IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE)*, pp. 520-524, 2017.

[25] Q. Zhao, B. Ye, X. Wang and D. Zhou, “Mixed image denoising method of non-local means and adaptive bayesian threshold estimation in NSCT domain,” *2010 3rd International Conference on Computer Science and Information Technology*, pp. 636-639, 2010.

[26] S. Fei and R. Zhao, “Adaptive wavelet shrinkage for image denoising based on sure rule,” *8th international Conference on Signal Processing*,vol. 1, 2006.

[27] Y. Q. Mohsin, G. Ongie and M. Jacob, “Iterative Shrinkage Algorithm for Patch-Smoothness

Regularized Medical Image Recovery," *in IEEE Transactions on Medical Imaging*, vol. 34, no. 12, pp. 2417-2428, 2015.

[28] S. Mirjalili, S. M. Mirjalilia and A. Lewis, "Grey Wolf Optimizer, " *In Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[29] T. I. Singh, R. Laishram and S. Roy, "Image segmentation using spatial fuzzy C means clustering and grey wolf optimizer," *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1-5, 2016.

[30] S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons, " *Applied Intelligence*, vol. 43, no. 1, pp. 150-161, 2015.

[31] D. Jitkongchuen and P. Phaidang, "Grey wolf algorithm with borda count for feature selection in classification," *2018 3rd International Conference on Control and Robotics Engineering (ICCRE)*, pp. 238-242, 2018.

[32] E. Daniel, "Optimum Wavelet Based Homomorphic Medical Image Fusion Using Hybrid Genetic – Grey Wolf Optimization Algorithm," *in IEEE Sensors Journal*, 2018.

[33] Y. L. Karnavas and I. D. Chasiotis, "PMDC coreless micro-motor parameters estimation through Grey Wolf Optimizer," *2016 XXII International Conference on Electrical Machines (ICEM)*, pp. 865-870, 2016.

[34] A. Buades, B. Coll, J. M. Morel., "A review of image denoising algorithms, with a new one," *SIAM Journal on Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, pp.490-530, 2005.

[35] C. Lu, S. Xiao, X. Li, L. Gao, "An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production," *In Advances in Engineering Software*, vol 99, pp. 161-176, 2016.

[36] T. Jayabarathi, T. Raghunathan, B. R. Adarsh and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *In Energy*, vol. 111, pp. 630-641, 2016.

[37] V. K. Kamboj, "A novel hybrid PSO–GWO approach for unit commitment problem. " *Neural Computing and Applications*, vol. 27, no. 6, pp. 1643-1655, 2016.

[38] V. Soni,G. Parmar,M. Kumar and S. Panda, "Hybrıd grey wolf optımızatıon-pattern search (hgwo-ps) optımızed 2dof-pıd controllers for load frequency control (lfc) ın ınterconnected thermal power plants," *ICTACT Journal on Soft Computing*, 2016.

[39] A. A. Heidari, P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *In Applied Soft Computing*, vol. 60, pp. 115-134, 2017.

[40] N. Mittal, U. Singh, and B. S. Sohi, "Modified Grey Wolf Optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, pp. 1-16, 2016.

[41] E. P. Ijjina and K. M. Chalavadi, “Human action recognition using genetic algorithms and convolutional neural networks,” *Pattern Recognition*, vol. 59, pp. 199-212, 2016.

[42] P. Erdoğmuş and S. Ekiz. “Nonlinear regression using particle swarm optimization and genetic algorithm.” *International Journal of Computer Applications*, vol. 153, no. 6, pp. 28-36, 2016.