# UNIVERSITY OF GONDAR

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE
DEPARTMENT OF INFORMATION TECHNOLOGY(MSC)

COURSE TITLE: MULTIMEDIA SYSTEM

PROJECT TITLE:-IMAGE PROCESSESSING USING MATLAB

BY: -



| NAME | ID |
|---|---|
| AlehegnAdane | 7655/07 |



Submitted to: Million Meshesha(PHD).

Gondar, Ethiopia

September, 2015

Contents

## Abstract

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Aspects of Image Processing are Image Enhancement, Image Restoration and Image Segmentation [1]. Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. Noise is any degradation in the image signal, caused by external disturbance. Noise can be Salt and pepper noise, Gaussian noise and Speckle noise[3]. Due to the existence of different noise in the multimedia data, the contents of the multimedia data could not view clearly to the users so that content-based multimedia data retrieval is getting a challenge. The objective of this project is to experiment noise removal and restoration technique using Matlab. I have used Gaussian noise, Salt and Pepper noise and Speckle noise as noise addition techniques and Gaussian filter and median filter as noise restoration/removal techniques. Median filter is often a better filter for reducing noise than the Gaussian and mean filter but it takes longer to compute. So, according to as my experiment procedure and result, I have recommended the median filter is best than others as noise removal technique.

## 1. Introduction

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Aspects of Image Processing are Image Enhancement, Image Restoration and Image Segmentation [1].

Image Enhancement: Processing an image so that the result is more suitable for a particular application. (sharpening or deblurring an out of focus image, highlighting edges, improving image contrast, or brightening an image, removing noise.

Image Restoration: This may be considered as reversing the damage done to an image by a known cause. (Removing of blur caused by linear motion, removal of optical distortions)

Image Segmentation: This involves subdividing an image into constituent parts, or isolating certain aspects of an image. (Finding lines, circles, or particular shapes in an image, in an aerial photograph, identifying cars, trees, buildings, or roads.

MATLAB stands for MATrix LABoratory. Hence, as the name suggests, here you play around with matrices. Hence, an image (or any other data like sound, etc.) can be converted to a matrix and then various operations can be performed on it to get the desired results and values. Image processing is quite a vast field to deal with. We can identify colors, intensity, edges, texture or pattern in an image [2].

Types of Digital Images are Binary, Grayscale and RGB.

Binary: Each pixel is just black or white. Since there are only two possible values for each pixel (0,1), we only need one bit per pixel.

Grayscale: Each pixel is a shade of gray, normally from 0 (black) to 255(white). This range means that each pixel can be represented by eight bits, or exactly one byte. Other grayscale ranges are used, but generally they are a power of 2.

True Color, or RGB: Each pixel has a particular color; that color is described by the amount of red, green and blue in it. If each of these components has a range 0–255, this gives a total of 256 3 different possible colors. Such an image is a "stack" of three matrices; representing the red, green and blue values for each pixel. This means that for every pixel there correspond 3 values.

Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information.[3]

• Noise is any degradation in the image signal, caused by external disturbance. Noise can be Salt and pepper noise, Gaussian noise and Speckle noise.

• Salt and pepper noise: It is caused by sharp, sudden disturbances in the image signal; it is randomly scattered white or black (or both) pixels. It can be modeled by random values added to an image.

• Gaussian noise: is an idealized form of white noise, which is caused by random fluctuations in the signal.

• Speckle noise: It is a major problem in some radar applications. It can be modeled by random values multiplied by pixel values.

## 2. Problem/challenge

The contents of the multimedia data should be view clearly to the users. One of the challenges in content-based Multimedia data retrieval is the existence of different types of noise in multimedia data. This unwanted noise should be properly handled using different image processing techniques.

## 3. Objective of the project

### 3.1 General objective

The general objective of this project is to experiment noise removal and restoration technique using Matlab.

### 3.2 Specific objectives

- ✓ To analyze different noise removal techniques,
- ✓ To add noise on the image,
- ✓ To remove noise on the image,
- ✓ To practice Matlab,
- ✓ To restore images
- ✓ To suggest the best noise removal technique
- ✓ To read different literatures

## 4. Requirement

- ✓ Scan historical printed documents,
- ✓ Matlab software
- ✓ Use noise addition techniques in matlab
- ✓ Use different noise removal and restoration techniques in matlab

## 5. Methods and algorithm used (experiment and result)

There are different noise removal and addition techniques in matlab.

**Matlab General Commands;**

- ✓ imread: Read an image
- ✓ figure: creates a figure on the screen.
- ✓ imshow(g): which displays the matrix g as an image.
- ✓ pixval on: turns on the pixel values in our figure.
- ✓ impixel(i,j): the command returns the value of the pixel (i,j)
- ✓ iminfo: Information about the image[4]

### 5.1 Noise addition and Removal techniques

Digital images are prone to a variety of types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise.
- Electronic transmission of image data can introduce noise.

To simulate the effects of some of the problems listed above, the toolbox provides the imnoise function, which you can use to *add* various types of noise to an image. The examples in this section use this function.

#### A. Gaussian additive noise

Additive white Gaussian noise affects each color component and pixel position independently. Each pixel in an image is disturbed by a Gaussian random variable with zero mean and variance plotting the amount of distortion of a pixel against the frequency plotting the amount of distortion of a pixel against the frequency with which it occurs produces a Gaussian distribution of noise. Using the imnoise() function, it is possible to add a specific type of noise to an image. The imnoise() function is a matlab function and for Gaussian noise it requires the mean and variance of the noise to be specified.

Example: - let us take sne image whose file name is “sampel” and its extension is “bmp”

- >> m=imread('sampel.bmp');
- >> z= imnoise(m,'gaussian',0.2,0.01);
- >> imshow(m); figure,imshow(z);
- >> u= imnoise(m,'gaussian',0.5,0.1);
- >> imshow(m); figure,imshow(u);
- Note that the mean and variance are in the range [0,1] as a percentage of the max gray

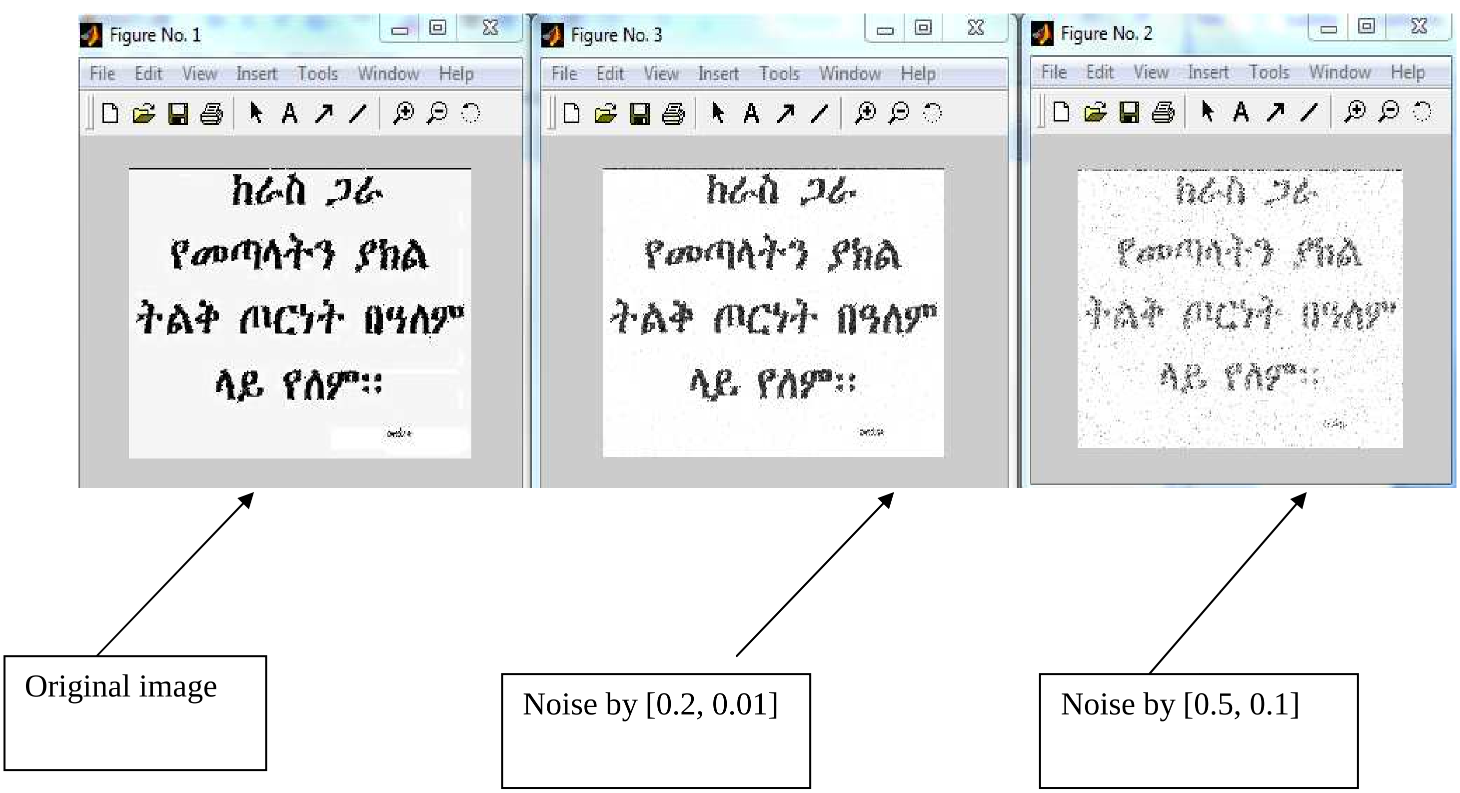



Figure 1:- Gaussian additive noise result

### B. Gaussian Filters

They are a class of linear smoothing filters with weights chosen according to a Gaussian function. It is a very good filter to remove noise drawn from a normal distribution.

**Properties of Gaussian filters**

- ✓ Amount of smoothing performed is the same in all directions, since Gaussian functions are isotropic.
- ✓ Gaussian filters smooth by replacing each image pixel with a weighted average of the neighboring pixels with weights monotonically decreasing with distance from central pixel. This property helps in retaining the edges.
- ✓ The width and hence the degree of smoothing of a Gaussian filter is parameterized by standard deviation.
- ✓ The larger the deviation, the wider the filter and the greater the smoothing and vice versa.

Let us remove the above image noise

```
>> m=imread('sampel.bmp');

>>u= imnoise(m,'gaussian',0.5,0.1);

>> filter = fspecial ('gaussian', [3 3], 0.1);

>> a =imfilter(u,  filter);
```

Filter by [3 3], 0.1)

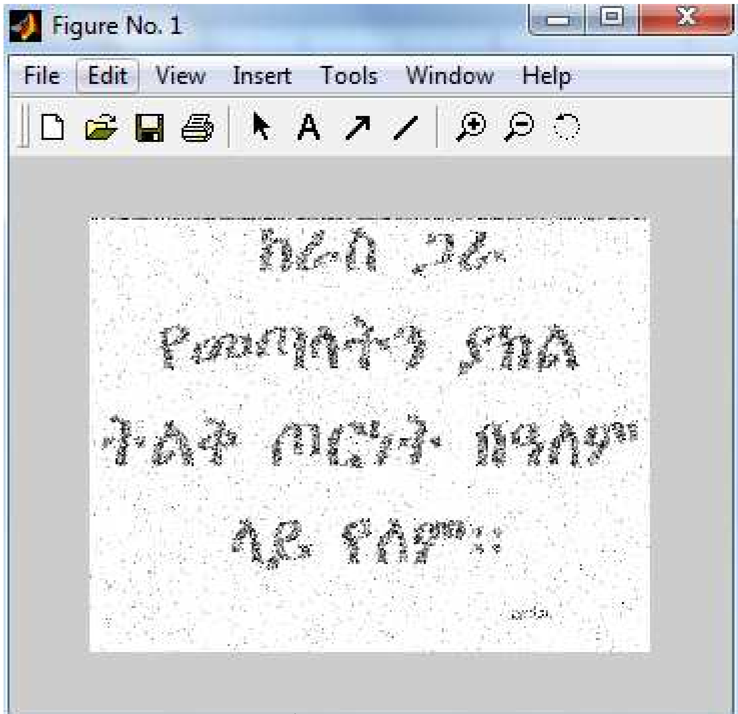
```
>> imshow(u); figure,imshow(a);

>> filter2 = fspecial('gaussian', [3 3], 0.5);

>> b = imfilter(u,  filter2);
```

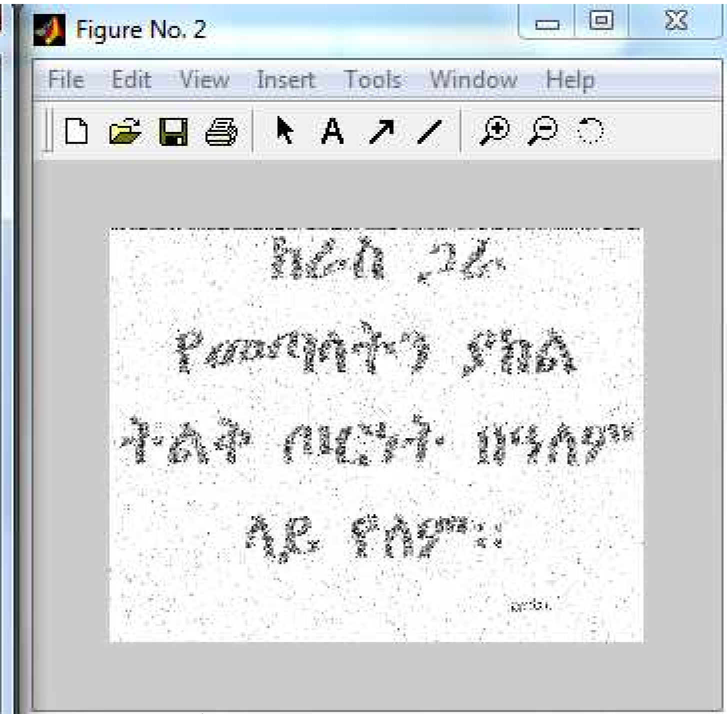
Filter by [3 3], 0. 5)

```
>> imshow(u); figure,imshow(b);

>> filter2 = fspecial('gaussian', [3 3], 1);

>> c = imfilter(u,  filter2);
```

Filter by [3 3], 1)

```
>> imshow(u); figure,imshow(c);
```




Noise image

Filter by [3 3], 0.1)

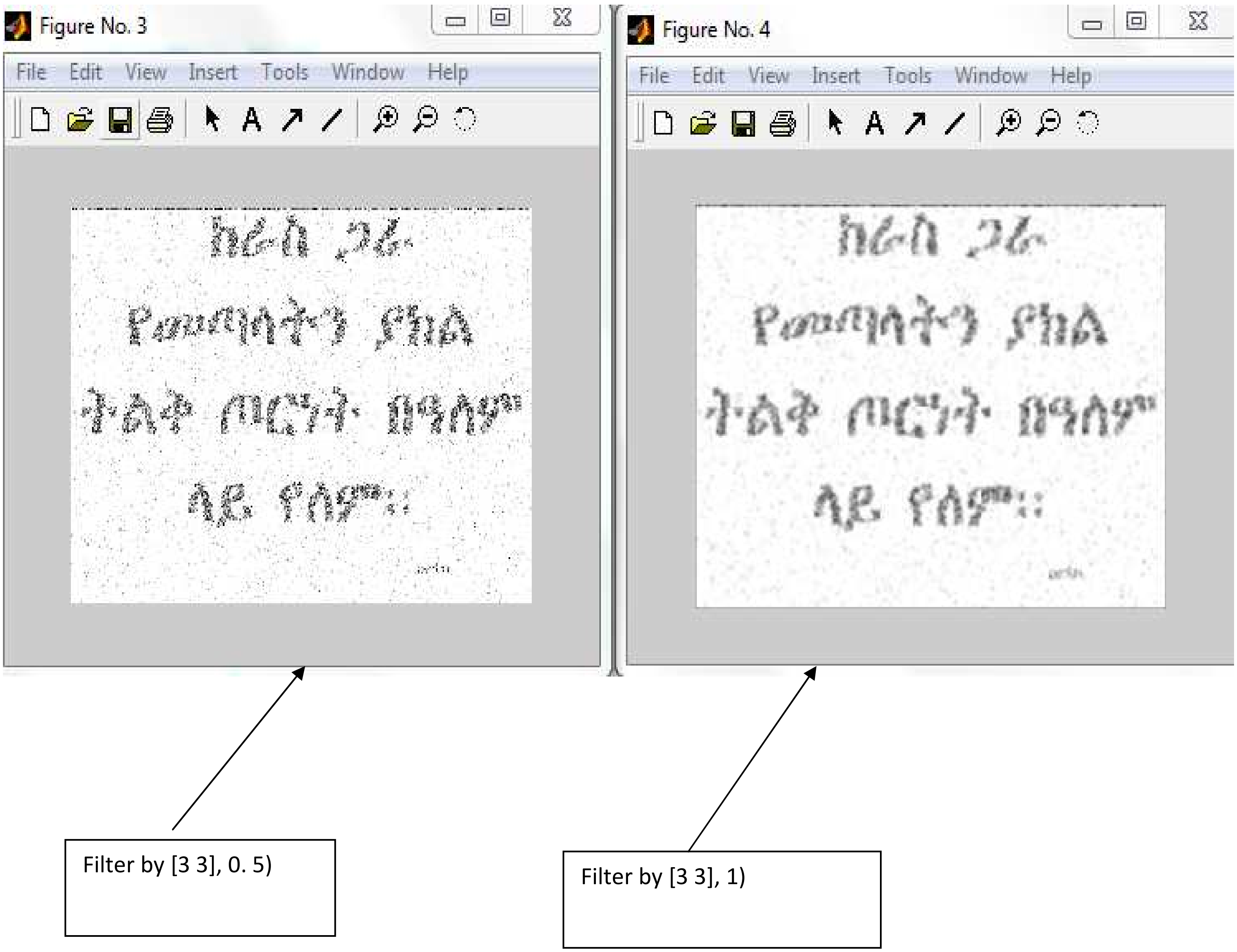



**Figure 2: Noise filters by Gaussian**

### C. Salt and Pepper noise

Each pixel in an image has probability p/2 (0<p<1) to be corrupted into either a white dot (salt) or a black be corrupted into either a white dot (salt) or a black dot (pepper)

Example:-let take image whose file name is “sample” and it file extension is “bmp”

```
q=imread('sampel.bmp');

>> j = imnoise(q,'salt & pepper', 0.02);

>> imshow(q); figure,imshow(j);

>> r= imnoise(q,'salt & pepper', 0.3);
```

```
>> imshow(q); figure,imshow(j);

>> imshow(q); figure,imshow(r);
```

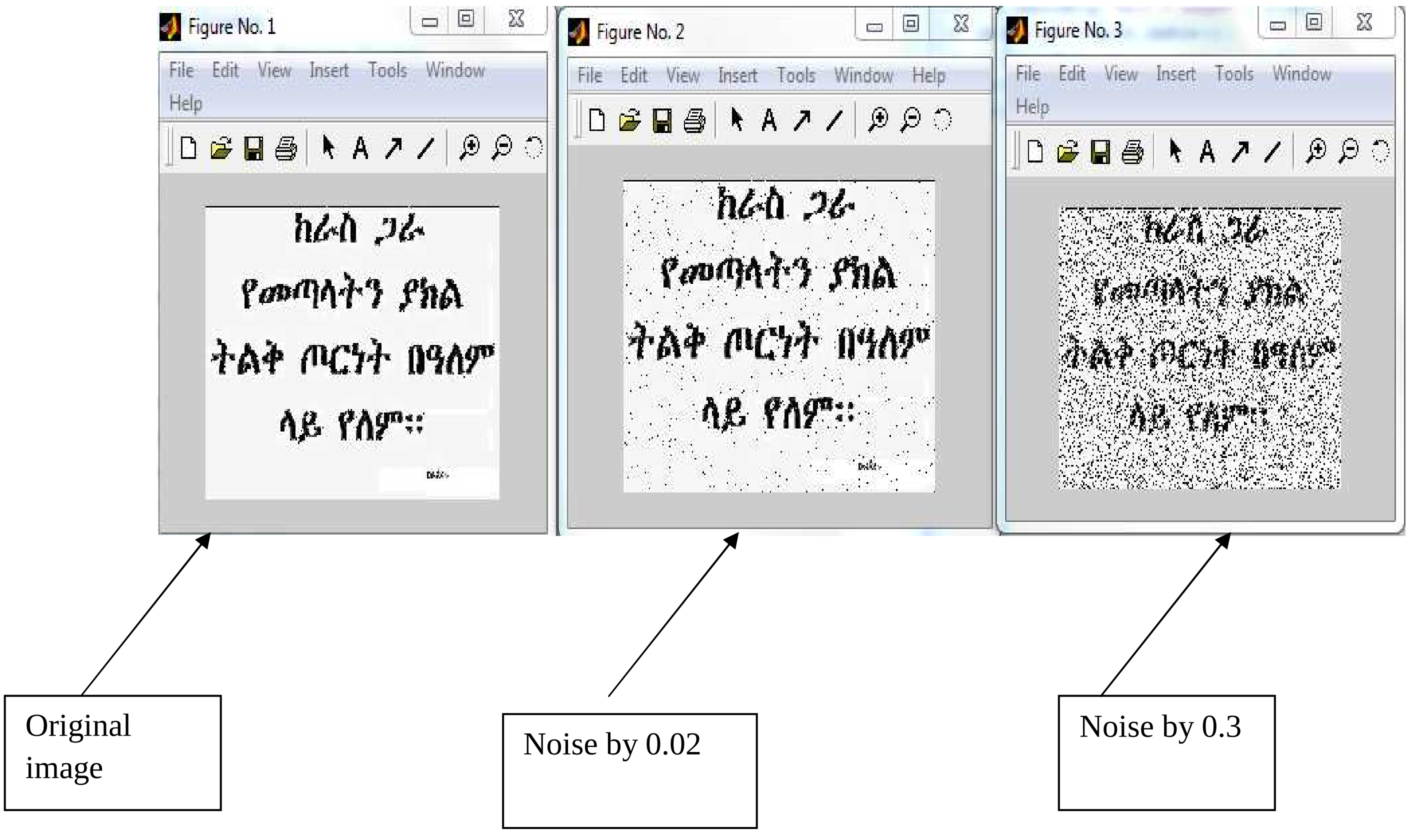



Figure 3:- Salt and Pepper noise addition result

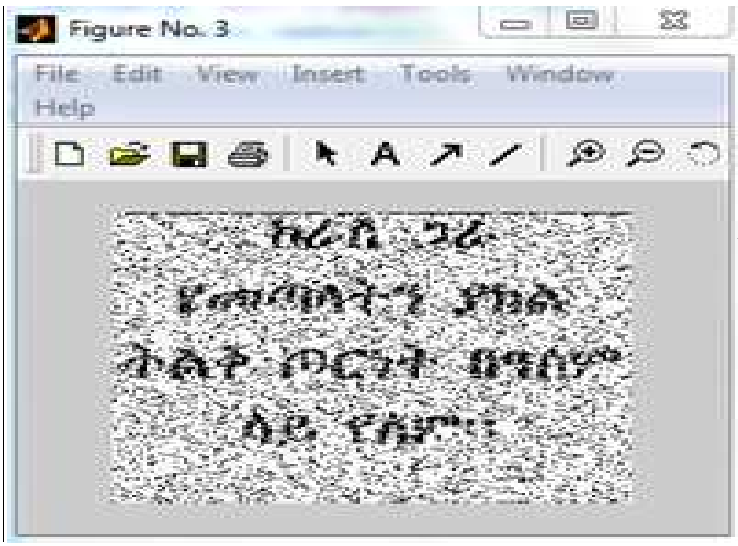
### D. Median Filter

Median filter is often a better filter for reducing noise than the mean filter but it takes longer to compute than the mean filter, but it takes longer to compute. The median is calculated by first sorting all the pixel values from the surrounding neighborhood and then values from the surrounding neighborhood and then replacing the pixel being considered with the middle pixel value pixel value. So let us remove such noise using Median filter as follows;

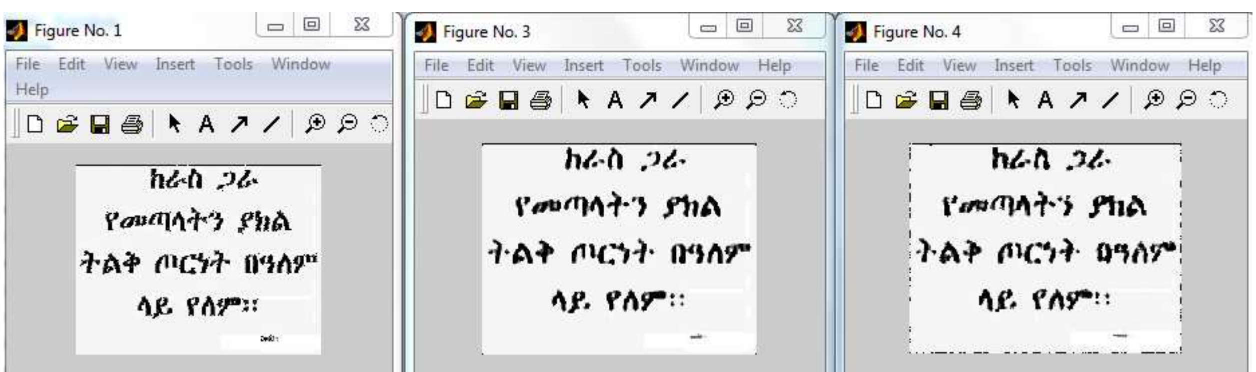
```
>> q=imread('sampel.bmp');

>> j = imnoise(q,'salt & pepper', 0.02);

>> imshow(q); figure,imshow(j);

>> r= imnoise(q,'salt & pepper', 0.3);
```

```
>> imshow(q); figure,imshow(j);

>> imshow(q); figure,imshow(r);

>> h = medfilt2(j,[5 5]);

>> w= medfilt2(r,[5 5]);

>> imshow(h); figure,imshow(w);
```

Median filter code




Noised image




Original image

0.02 noises-filter by [5 5]

0.3 noise-filter by [5 5]

**Figure 4:-median filter result**

From this it is possible to conclude that as more noise added, removing that noise is difficult

*Example 2:- noise image*

```
s=imread('image.jpg');
>> c=rgb2gray(s);
>> h = medfilt2(c,[5 5]);
>> imshow(c); figure,imshow(h);
>> imshow(s); figure,imshow(h);
>> h = medfilt2(c,[2 2]);
>> imshow(s); figure,imshow(h);
>> h = medfilt2(c,[5 5]);
>> imshow(s); figure,imshow(h);
```

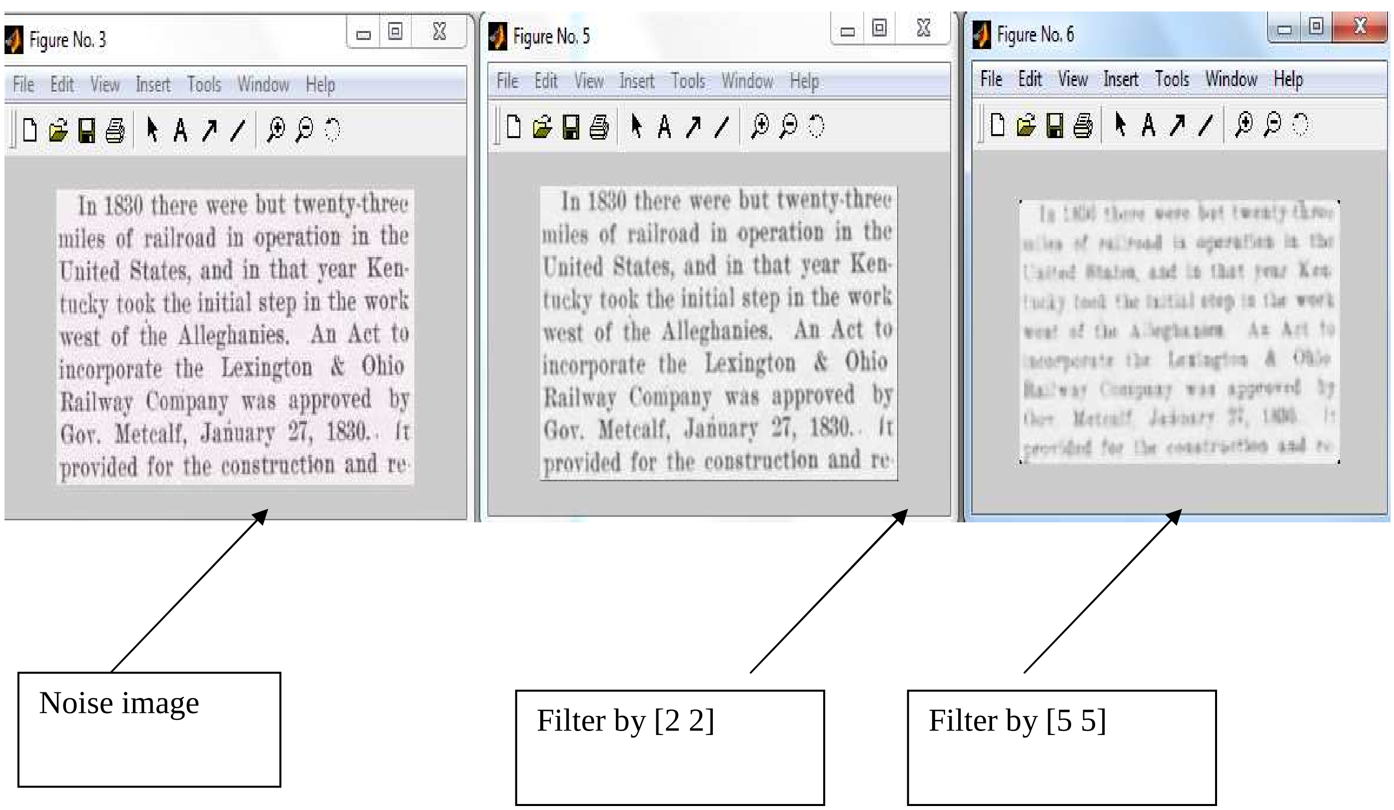



*Figure 5: noise removal by median filter*

# 6. Conclusion

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Aspects of Image Processing are Image Enhancement, Image Restoration and Image Segmentation. One of the challenges in content-based Multimedia data retrieval is the existence of different types of noise in multimedia data. The general objective of this project is to experiment noise removal and restoration technique using Matlab. I have used Gaussian additive noise and Salt and Pepper noise as noise addition techniques and Gaussian filter and median filter as noise restoration/removal techniques.

Median filter is often a better filter for reducing noise than the mean filter but it takes longer to compute than the mean filter, but it takes longer to compute. The median is calculated by first sorting all the pixel values from the surrounding neighborhood and then values from the surrounding neighborhood and then replacing the pixel being considered with the middle pixel value pixel value. So, according to as my experiment procedure and result, I have recommended the median filter is best than others as noise removal technique.

## 7. Sample Code

```
%IMAGE PROCESSING PROJECT

 m=imread('sampel.bmp');title('original image')

figure,imshow(m);

 imnoise(m,'gaussian',0.2,0.01);

 figure,imshow(m);

 u= imnoise(m,'gaussian',0.5,0.1);

 figure,imshow(u);

 m=imread('sampel.bmp');title('alehegn')

 u= imnoise(m,'gaussian',0.5,0.1);

 filter = fspecial ('gaussian', [3 3], 0.1);

 a =imfilter(u,  filter);

figure,imshow(a);

filter2 = fspecial('gaussian', [3 3], 0.5);

b = imfilter(u,  filter2);

imshow(u); figure,imshow(b);

filter2 = fspecial('gaussian', [3 3], 1);

c = imfilter(u,  filter2);

figure,imshow(c)

q=imread('sampel.bmp');title('original image')
```

```
j = imnoise(q,'salt & pepper', 0.02);

imshow(q); figure,imshow(j);

r= imnoise(q,'salt & pepper', 0.3);

imshow(q); figure,imshow(j);

 figure,imshow(r);

q=imread('sampel.bmp');title('original image')

j = imnoise(q,'salt & pepper', 0.02);

imshow(q); figure,imshow(j);

r= imnoise(q,'salt & pepper', 0.3);

figure,imshow(j);

figure,imshow(r);

h = medfilt2(j,[5 5]);

w= medfilt2(r,[5 5]);

imshow(h); figure,imshow(w);

s=imread('sample2.jpg');title('original image')

c=rgb2gray(s);

h = medfilt2(c,[5 5]);

imshow(c); figure,imshow(h);

imshow(s); figure,imshow(h);

h = medfilt2(c,[2 2]);

imshow(s); figure,imshow(h);
```

```
h = medfilt2(c,[5 5]);

figure,imshow(h);

a=imread('sample4.png');subplot(1,1,1);Imshow(a);title('original image');

%% draw the cumulative  histogram of  orginal image.

   figure;

histogram = imhist(a);%image histogram.

   c= cumsum(histogram);%cumulative  histogram.

   subplot(1,1,1);bar(c,'BarWidth',8); title('draw bar');%draw bar..

   %% image equalization and cumulative histogram

   figure;

   j = histeq(a);

   subplot(1,1,1);imshow(j); title('image equalization and cumulative histogram ')

   figure;

histogramEq = imhist(j);%image histogram.

   c = cumsum(histogramEq);%cumulative  histogram.

   bar(c,'BarWidth',8)%draw bar.

%blurred image

i = fspecial('gaussian',7,10);

Blurred = imfilter(a,i,'symmetric','conv');

figure;subplot(1,1,1);imshow(Blurred);title('Blurred Image');

UNDERi = ones(size(i)-4);
```

```
[i1, P1] = deconvblind(Blurred,UNDERi);

figure;subplot(1,1,1);imshow(i1);title('Deblurred Image');

b=rgb2gray(a);subplot(1,1,1);imshow(b);title('gray image');

 t = wiener2(b,[3,3]); subplot(1,1,1); imshow(t); title('wiener adaptive filtering');

l = medfilt2(b,[5  5]);figure, subplot(1,1,1);imshow(l); title('median  filtering');

z = imadjust(b,[0.3 0.7]);figure, subplot(1,1,1);imshow(z);title('imadjust picture');

%quantization

rgb= imread('sample4.png'),[x,map]=rgb2ind(rgb,0.5),subplot(1,1,1);imshow(rgb);title('quantization')

rgb = imread('sample4.png'), [x,map]=rgb2ind(rgb,0.1),subplot(1,1,1);imshow(rgb);title('quantization')

%add noise before binarization

black=3;

white=253;

%Adjust the values in 'black' and 'white' to increase the noise.

NoiseImg = b;

Rmatrix = randint(size(NoiseImg,1),size(NoiseImg,2),[0,255]);

NoiseImg(Rmatrix <= black) = 0;

NoiseImg(Rmatrix >=white) = 255;

figure,subplot(1,1,1),imshow(NoiseImg),title('Add "Salt and Pepper" Noise');

t=wiener2(NoiseImg,[1 1]);figure, subplot(1,1,1); imshow(t); title('wiener adaptive filtering');

rimg = medfilt2(NoiseImg);subplot(1,1,1),imshow(rimg),title('median  filtering');

x = edge(NoiseImg,'prewitt');figure, subplot(1,1,1);imshow(x);title('prewitt Edge detected Image');
```

```
bw2 = edge(NoiseImg,'canny');figure, subplot(1,1,1);imshow(bw2);title('canny Edge detected Image');

bw = edge(NoiseImg,'sobel',0.1); subplot(1,1,1);imshow(bw);title('sobel Edge detected Image');

h= fspecial('laplacian');

blurred = imfilter(NoiseImg,h);subplot(1,1,1);imshow(blurred); title('laplacian Edge detected Image');

%Read Background Image

background=imread('sample4.png');

%Read Current Frame

CurrentFrame=imread('sample4.png');

%Display Background and Foreground

subplot(2,2,1);imshow(background);title('backGround');

subplot(2,1,2);imshow(CurrentFrame);title('Current Frame');

r = imread('sample3.jpg');subplot(1,1,1);imshow(r), imhist(r,255) ;title('histogram');

r = imread('sample3.jpg');subplot(1,1,1);imshow(r), imhist(r,100) ;title('histogram');

% Image with Middle level Noise

n=imread('sample3.jpg');

%% draw the cumulative  histogram of  orginal image.

   figure;

histogram = imhist(n);%image histogram.

   c= cumsum(histogram);%cumulative  histogram.

   subplot(1,1,1);bar(c,'BarWidth',16); title('draw bar')%draw bar.

   %% image equalization and cumulative histogram
```

```
   figure;
   j= histeq(n);
   subplot(1,1,1);imshow(j); title('image equalization and cumulative histogram ')
   figure;
histogramEq = imhist(j);%image histogram.
   c = cumsum(histogramEq);%cumulative  histogram.
  m= rgb2gray(n);Imshow(m);title('gray image');
j= wiener2(m,[3 3]);figure, imshow(j); title('wiener adaptive filtering');
l = medfilt2(m,[5  5]);figure, imshow(l); title('median  filtering');
z = imadjust(l,[0.3 0.7],[]);
figure, imshow(z);title('imadjust picture');
%blurred image
j= fspecial('gaussian',7,10);
Blurred = imfilter(n,j,'symmetric','conv');
figure;imshow(Blurred);title('Blurred Image');
UNDERJ = ones(size(j)-4);
[j1, P1] = deconvblind(Blurred,UNDERJ);
figure;imshow(j1);title('Deblurred Image');
%add noise before binarization
black=5;
white=251;
```

```
%Adjust the values in 'black' and 'white' to increase the noise.

img = m;

Rmatrix = randint(size(img,1),size(img,2),[0,255]);

img(Rmatrix <= black) = 0;

img(Rmatrix >=white) = 255;

figure,subplot(1,1,1),imshow(img),title('Add "Salt and Pepper" Noise');

x = edge(img,'prewitt');figure, subplot(1,1,1);imshow(x);title('prewitt Edge detected Image');

bw2 = edge(img,'canny');figure, subplot(1,1,1);imshow(bw2);title('canny Edge detected Image');

bw = edge(img,'sobel',0.1); subplot(1,1,1);imshow(bw);title('sobel Edge detected Image');

h= fspecial('laplacian');

blurred = imfilter(img,h);subplot(1,1,1);imshow(blurred); title('laplacian Edge detected Image')

t=wiener2(img,[1 1]);figure, subplot(1,1,1); imshow(t); title('wiener adaptive filtering');

rimg = medfilt2(img);subplot(1,1,1),imshow(rimg),title('median  filtering');

%the frequency of the image

[h,w] = freqz(rimg,1,8,'whole');

colormap(jet(100))

plot(w/pi-3,fftshift(abs(h)))

[h,w] = freqz(rimg,1,16,'whole');

colormap(jet(100))

plot(w/pi-2,fftshift(abs(h)))

[h,w] = freqz(rimg,1,24,'whole');
```

```
colormap(jet(100))

plot(w/pi-1,fftshift(abs(h)))

r = imread('sample6.jpg');imshow(r), imhist(r,100) ;title('histogram');

% Image with High level Noise

n=imread('sample1.jpg');subplot(1,1,1);Imshow(n);title('original image');

%% draw the cumulative  histogram of  orginal image.

   figure;

histogram = imhist(n);%image histogram.

c = cumsum(histogram);%cumulative  histogram.

subplot(1,1,1);bar(c,'BarWidth',8); title('draw bar');%draw bar..

%% image equalization and cumulative histogram

figure;

j = histeq(n);

subplot(1,1,1);imshow(j); title('image equalization and cumulative histogram ')

figure;

histogramEq = imhist(j);%image histogram.

c = cumsum(histogramEq);%cumulative  histogram.

bar(c,'BarWidth',8)%draw bar.

s = rgb2gray(n);subplot(1,1,1);Imshow(s);title('gray image');

l = wiener2(s,[3 3]);figure, imshow(l); title('wiener adaptive filtering');

 l= medfilt2(s,[5  5]);figure, subplot(1,1,1);imshow(l); title('median  filtering');
```

```
z = imadjust(l,[0.3 0.7]);figure,subplot(1,1,1); imshow(z);title('imadjust picture');

%quantization

rgb = imread('sample1.jpg'),[x,map]=rgb2ind(rgb,0.5),imshow(rgb);title('quantization')

rgb = imread('sample1.jpg'), [x,map]=rgb2ind(rgb,0.1),imshow(rgb);

%add noise before binarization

black=3;

white=253;

%Adjust the values in 'black' and 'white' to increase the noise.

Noise = s;

Rmatrix = randint(size(Noise,1),size(Noise,2),[0,255]);

Noise(Rmatrix <= black) = 0;

Noise(Rmatrix >=white) = 255;

figure,subplot(1,1,1),imshow(Noise),title('Add "Salt and Pepper" Noise');

t=wiener2(Noise,[1 1]);figure, subplot(1,1,1); imshow(t); title('wiener adaptive filtering');

rimg = medfilt2(Noise);subplot(1,1,1),imshow(rimg),title('median  filtering');

Z = imadjust(rimg,[0.3 0.7],[]);figure,subplot(1,1,1); imshow(z);title('imadjust picture');

r = imread('sample3.jpg');imshow(r), imhist(r,255) ;title('histogram');

r = imread('sample3.jpg');imshow(r), imhist(r,64) ;title('histogram');

x = edge(Noise,'prewitt');figure, subplot(1,1,1);imshow(x);title('prewitt Edge detected Image');

bw2 = edge(Noise,'canny');figure, subplot(1,1,1);imshow(bw2);title('canny Edge detected Image');

bw = edge(Noise,'sobel',0.1); subplot(1,1,1);imshow(bw);title('sobel Edge detected Image');
```

```
 h= fspecial('laplacian');
blurred = imfilter(Noise,h);subplot(1,1,1);imshow(blurred); title('laplacian Edge detected Image');
% Image with Very High level Noise
b= imread('sample4.jpg');subplot(1,1,1);imshow(b); ;title('original image');
%% the cumulative  histogram of  orginal image.
   figure;
histogram = imhist(b);%image histogram.
   h = cumsum(histogram);%cumulative  histogram.
   subplot(1,1,1);bar(h,'BarWidth',32); title('draw bar')%draw bar..
   %% image equalization and cumulative histogram
   figure;
   i = histeq(b);
   subplot(1,1,1);imshow(i); title('image equalization and cumulative histogram ')
   figure;
histogramEq = imhist(i);%image histogram.
   h = cumsum(histogramEq);%cumulative  histogram.
   bar(h,'BarWidth',16);%draw bar.
f = rgb2gray(b);subplot(1,1,1);Imshow(f);title('gray image');
j = wiener2(f,[1 1]);figure,subplot(1,1,1); imshow(j); title('wiener adaptive filtering');
l = medfilt2(f,[5  5]);figure,subplot(1,1,1); imshow(l); title('median  filtering');
z = imadjust(f,[0.3 0.7],[]);figure,subplot(1,1,1); imshow(z);title('imadjust picture');
```

```
%blurred image

j= fspecial('gaussian',7,10);

Blurred = imfilter(n,j,'symmetric','conv');

figure;imshow(Blurred);title('Blurred Image');

UNDERJ = ones(size(j)-4);

[j1, P1] = deconvblind(Blurred,UNDERJ);

figure;imshow(j1);title('Deblurred Image');

%quantization

%add noise before binarization

black=3;

white=253;

%Adjust the values in 'black' and 'white' to increase the noise.

y = f;

Rmatrix = randint(size(y,1),size(y,2),[0,255]);

y(Rmatrix <= black) = 0;

y(Rmatrix >=white) = 255;

figure,subplot(1,1,1),imshow(y),title('Add "Salt and Pepper" Noise');

t=wiener2(y,[1 1]);figure, subplot(1,1,1); imshow(t); title('wiener adaptive filtering');

rimg = medfilt2(y);subplot(1,1,1),imshow(rimg),title('median  filtering');

Z= imadjust(y,[0.3 0.7]);figure,subplot(1,1,1); imshow(Z);title('imadjust picture');

x = edge(y,'prewitt');figure, subplot(1,1,1);imshow(x);title('prewitt Edge detected Image');
```

```
bw2 = edge(y,'canny');figure, subplot(1,1,1);imshow(bw2);title('canny Edge detected Image');

bw = edge(y,'sobel',0.1); subplot(1,1,1);imshow(bw);title('sobel Edge detected Image');

 h= fspecial('laplacian');

blurred = imfilter(y,h);subplot(1,1,1);imshow(blurred); title('laplacian Edge detected Image')

%the frequency of the image

[h,w] = freqz(y,1,24,'whole');

colormap(jet(256))

plot(w/pi-3,fftshift(abs(h)))

[h,w] = freqz(rimg,1,8,'whole');

colormap(jet(255))

plot(w/pi-2,fftshift(abs(h)))

[h,w] = freqz(rimg,1,16,'whole');

colormap(jet(100))

plot(w/pi-1,fftshift(abs(h)))

r = imread('sample3.jpg');imshow(r), imhist(r,255) ;title('histogram');

r = imread('sample3.jpg');imshow(r), imhist(r,64) ;title('histogram');

%Binarized image filtering

%1. low image filtering

rgb = imread('sample4.jpg');figure,subplot(1,1,1);imshow(rgb);title('Original Image');

b=rgb2gray(rgb);figure,subplot(1,1,1);Imshow(b);title('gray image');

binI = dither(b);figure, subplot(1,1,1);imshow(binI);title('binary Image');
```

```
t = wiener2(binI,[3 3]);figure, imshow(t); title('wiener adaptive filtering');

l = medfilt2(binI,[5  5]);figure, imshow(l); title('median  filtering');

z= imadjust(b,[0.3 0.7]);figure,subplot(1,1,1); imshow(z);title('imadjust picture');

%% the cumulative  histogram of  binarized image.

   figure;

histogram = imhist(binI);%image histogram.

   h = cumsum(histogram);%cumulative  histogram.

   subplot(1,1,1);bar(h,'BarWidth',1); title('draw bar');%draw bar.

%add noise after binarization

g = imnoise(binI,'salt & pepper',0.02); subplot(1,1,1);imshow(g);title(' salt and pepper binarized noisy image');

m = imnoise(binI,'speckle',0.04); subplot(1,1,1);imshow(m); title('binarized noisey image with speckle')

t = imnoise(binI,'gaussian',0.06); subplot(1,1,1);imshow(t); title('binarized noisey image with gaussian')

k = wiener2(h,[3 3]);figure, subplot(1,1,1);imshow(k); title('wiener adaptive filtering');

l = medfilt2(h,[3 3]);figure,subplot(1,1,1); imshow(l);title('median filtered Image ');

v=imread('end.jpg');

imshow(v);
```

## 8. Reference


[1]. Haris Papasaika-Hanusch Institute of Geodesy and Photogrammetry,”Digital Image Processing Using Matlab”

[3]. The Ohio State University, “A MATLAB Tutorial” January 29, 2015 12:18 a.m

[4].Stormy Attaway, MA Matlab:” Practical Introduction to Programming and Problem Solving”.2009

[4]. Joseph E.Gonzalez “ the language of technical computing:Matlab Tutorial” 2007