Jessore University of Science & Technology

Department of Computer Science & Engineering

Course Title: Bio-Medical Signal and Image Processing Lab

Course Code: CSE-4104

Submitted to	Submitted by
Dr. A F M Shahab Uddin Lecturer Dept. of Computer Science and Engineering, Jashore University of Science and Technology.	Md. Shongkor Talukdar Roll: 180129 4th Year 1st Semester Session:2018-2019 Dept. of Computer Science and Engineering Jashore University of Science and Technology.

Date of Submission: 06.12.2023

Index

Experiment no: 1

Experiment name: Digital image read, write and show.
Objective:

Read an image from local storage or other source.Write the image to other file. Show the image read or write.

Requirement:

PC (minimum 4 GB RAM).
Windows 10(recommended).
Octave latest version with image processing package installed.

Description:

In this experiment, we will simply read an image. Then we will write it and show.

Environment Setup Process:

Open octave and create new file.
Save file with .m extension (M type file).
Install image package and load package by “pkg load image” command in octave
command prompt. So that we can use the library functions.

Code:

image read
image = imread('image.jpg');
imshow(image);
subplot(1,2,1);

image write

imwrite(image, 'newimage.jpg');
writtenImage = imread('newimage.jpg');
subplot(1,2,2);

imshow(writtenImage);
title('New Image');

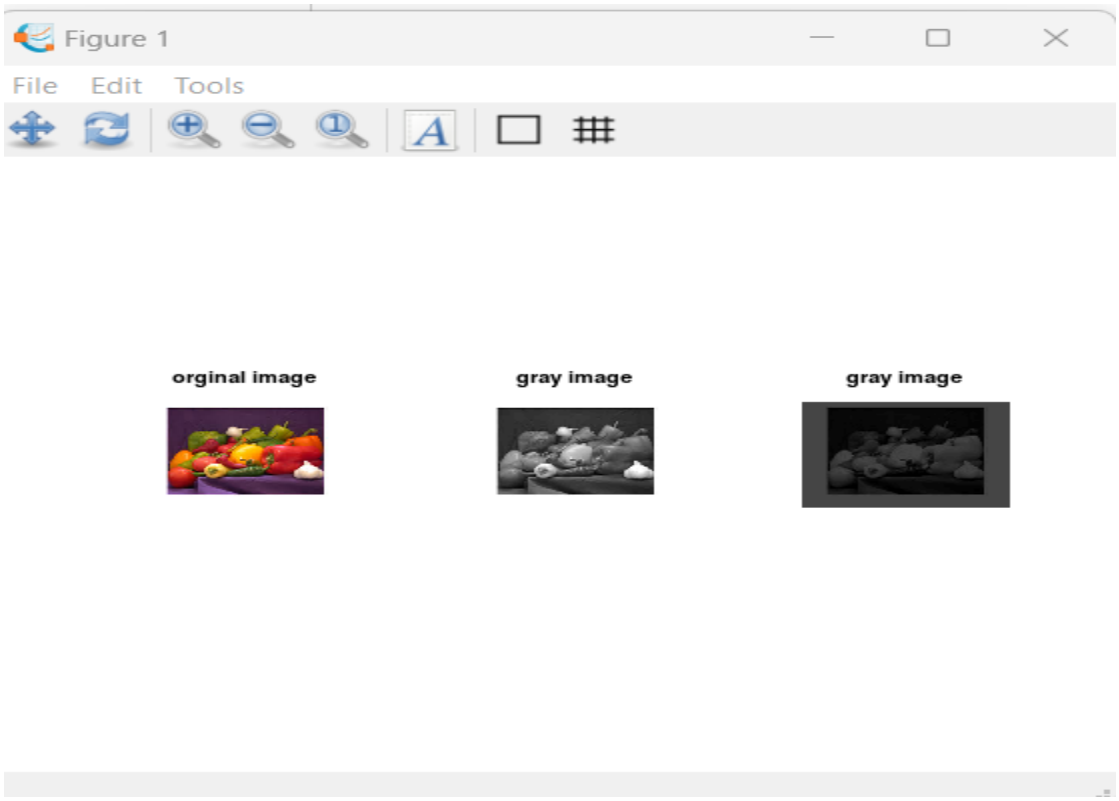

Fig 1: Read and write image

Here we read the original image named image. Then we write the first image in gray image.

Experiment no: 2

Experiment name: Noise removing using filter and average method.

Objective:

Make some noisy images.
Average image noises.
Remove noise from image.

Requirement:
PC (minimum 4 GB RAM).
Windows 10(recommended).
Octave latest version with image processing package installed.
Description:
In this experiment, we will remove noise. At first, we will read an image. Then we will
generate some noisy images from the input image. After that we will make the average
of the noisy images. At last we will de noise the noisy image.

Environment Setup Process:
Open octave and create new file.
Save file with .m extension (M type file).
Install image package and load package by “pkg load image” command in octave
command prompt. So that we can use the library functions.

Code:

```
mean = 0;
sd=0;
noise1 = 0;
%Read image
im = imread('1.jpg');
%Resize image
im = imresize(im,[700,600]);
subplot(3,3,1);
imshow(im);
title('Original Image');
i=3;
for n=0:5
   sd=sd+0.05;
   noiseimg =
im2double(im)+sqrt(sd)*randn(size(im))+mean;
   i=i+1;
   subplot(3,3,i);
   imshow(noiseimg);
   title(['Noisy Image ' num2str(n+1)]);
   noise1 = noise1+noiseimg;
end

avgNoise = noise1/6;
subplot(3,3,2);
```

```
imshow(avgNoise);
title('Average Noisy Image');
% Remove noise using Gaussian filtering
% denoisedImage = imgaussfilt(avgNoise, 1);
% subplot(3,3,2);
% imshow(denoisedImage);
% title('denoisedImage');
avgNoise = uint8(avgNoise);
deNoise = im-avgNoise;
figure
imshow(deNoise);
title('De Noised Image');
```

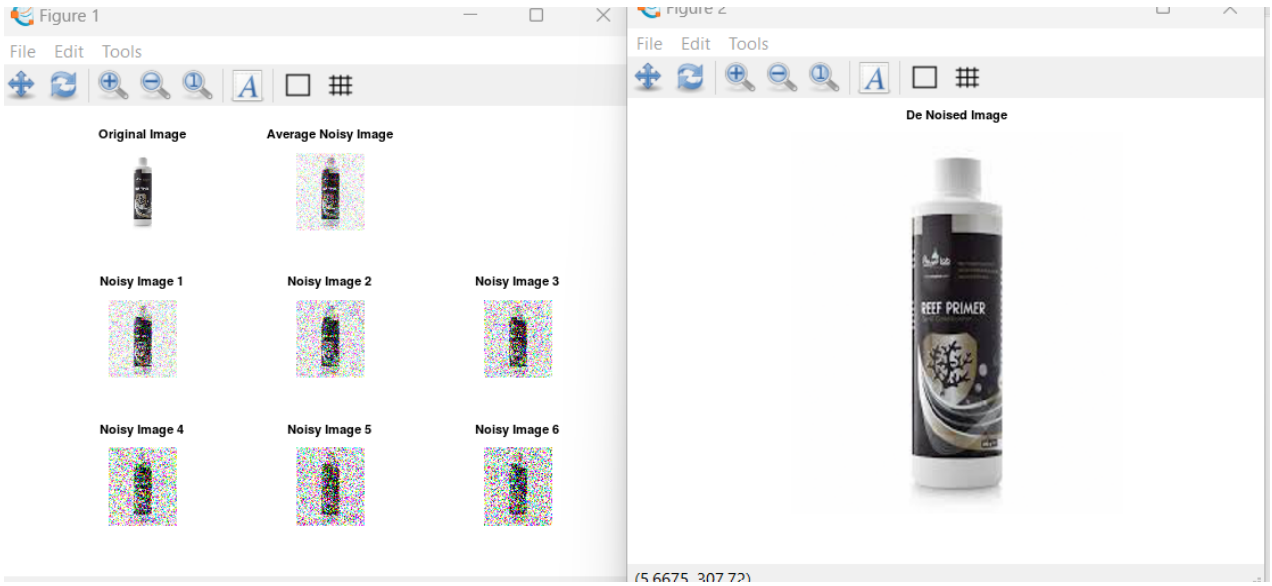

Fig- with noise and denoise

Experiment no: 3

Experiment name: Convert color image to negative image.
Objective:

Convert color image to negative image.

Requirement:

PC (minimum 4 GB RAM).
Windows 10(recommended).
Octave latest version with image processing package installed.

Description:

In this experiment, we will simply convert a color image to negative image.

Environment Setup Process:

Open octave and create new file.
Save file with .m extension (M type file).
Install image package and load package by “pkg load image” command in octave
command prompt. So that we can use the library functions.

Code:

```
pkg load image

% Load the image
A = imread('1.jpg');

% Display the original image and its negative
subplot(2, 1, 1);
imshow(A);
title('Original Image');

L = 256;
B = (L - 1) - A;
subplot(2, 1, 2);
imshow(B);
title('Negative Image');

% Create a new figure
figure;

% Convert the image to grayscale
C = rgb2gray(A);

% Display the grayscale version of the original image and
its negative
```

```
subplot(2, 1, 1);
imshow(C);
title('Grayscale Original Image');

L = 256;
D = (L - 1) - C;
subplot(2, 1, 2);
imshow(D);
title('Negative of Grayscale Image');
```

Fig 3: Negative image

Experiment no: 4
Experiment name: RGB image to Gray scale image conversion by average method.

Objective:

Convert RGB image to gray scale image.
Print RGB and gray scale image.

Requirement:

PC (minimum 4 GB RAM).
Windows 10(recommended).
Octave latest version with image processing package installed.

Description:

In this experiment, we will convert a RGB image to Gray scale image by using average
Method.

Average method= (R+ G+B)/3

By using this method, we will try to make gray scale image where red, green and blue
colors are divided equally.

Environment Setup Process:
Open octave and create new file.
Save file with .m extension (M type file).
Install image package and load package by “pkg load image” command in octave
command prompt. So that we can use the library functions.

Code:

```
clc
clear all
a= imread('2.jpg');
subplot(1,3,1)
imshow(a);
title('orginal image')
b= rgb2gray(a)
subplot(1,3,2)
imshow(b);
title('gray image')
c = rgb2ind(a)
subplot(1,3,3)
imshow(c);
title('gray image')
```

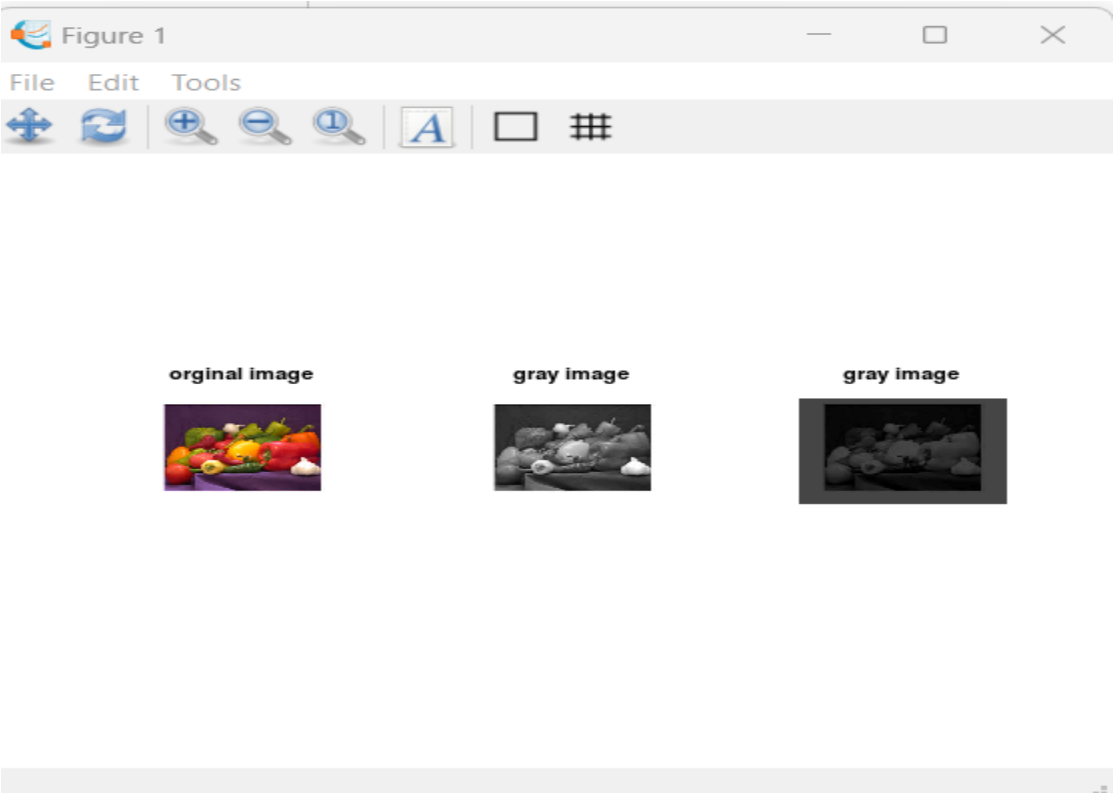

Fig 4: RGB to Gray scale conversion

Here, we convert a RGB image to Gray scale image. But in average method the color quality of gray scale image is very poor as they are equally distributed.