LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.04**

**A.1 Aim:**

To write a program to enhance the quality of an image (on your own photograph) by noise removal.

**A.2 Prerequisite:**

1 MATLAB programming syntax (Refer the MATLAB manual).

2. Knowledge of filtering techniques.

2. Availability of Soft copy of your Photograph for experiment.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Understand the fundamentals of Noise and its effects in digital images
2. Design HPF, LPF and median filters.
3. Remove noise from given image using following Filtering methods.
4. Low Pass Filtering
5. High Pass Filtering
6. Median Filtering.
7. Identify applications of filtering techniques studied.

**A.4 Theory:**

**Filtering**

**Noise:**

Images are corrupted by random variations in intensity values called noise due to non-perfect camera acquisition or environmental conditions.

**Common Types of Noise**

* **Salt and pepper noise:** random occurrences of both black and white intensity values
* **Impulse noise:** random occurrences of white intensity values
* **Gaussian noise:** impulse noise but its intensity values are drawn from a Gaussian distribution. Noise Intensity Value is given by following equation:

…… Equation (1)

k: random intensity value

**Effects of noise on digital images:**

Presence of particular type of noise in an image may deteriorate image quality to certain level. The level of deterioration depends on the density of noise

**Noise Filtering:**

**Basic Idea**: replace each pixel intensity value with a new value taken over a neighborhood of fixed size

**The size of the filter controls degree of smoothing**

* large filter » large neighborhood » intensive smoothing

**Low Pass Filter:**

Low pass filtering removes the high frequency contents from the image. It is used to remove noise present in the image. Noise, is normally a high frequency signal and low pass filtering eliminates the noise.

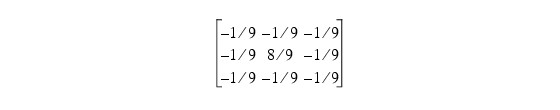
A low pass filter is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.

Using a low pass filter tends to retain the low frequency information within an image while reducing the high frequency information. An example is an array of ones divided by the number of elements within the kernel, such as the following 3 by 3 kernel: 

**High Pass Filter:**

High pass filtering eliminates the low frequency regions while retaining or enhancing the high frequency components. An image, which is high passed, would have no background and would have enhanced edges.

A high pass filter is the basis for most sharpening methods. An image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness. A high pass filter tends to retain the high frequency information within an image while reducing the low frequency information. The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels. The kernel array usually contains a single positive value at its center, which is completely surrounded by negative values. The following array is an example of a 3 by 3 kernel for a high pass filter:



**Median Filter**

Replace each pixel value with the median of the gray values in the region of the pixel:

1. Take a 3 x 3 (or 5 x 5 etc.) region centered around pixel (i,j)

2. Sort the intensity values of the pixels in the region into ascending order

3. Select the middle value as the new value of pixel (i,j)

Computation of Median Values:



**Figure 1. Computation of Median values**

**A.5 Procedure/Algorithm:**

**A.5.1 TASK 1:**

**Image Filtering**

1. Read the input image.

2. Add noise to the input image. (Make use of Use imnoise( ) ).

3. Design your own HPF, LPF and Median filters

4. Apply LPF, HPF and Median Filter to the image.

4. Display the original and the output image.

5. Observe the output and complete PART B of lab manual.

6. Add salt and pepper noise to the input image.

7. Apply LPF, HPF and Median Filter to the image obtained in step 6.

8. Display the original and the output images.

9. Observe the output and complete PART B of lab manual.

10. Save and close the file and name it as **EX4\_Task1\_your Roll no.m**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

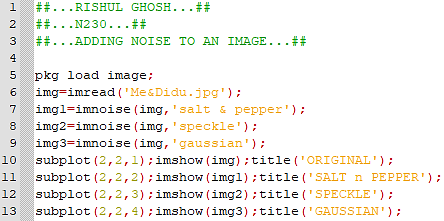
(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

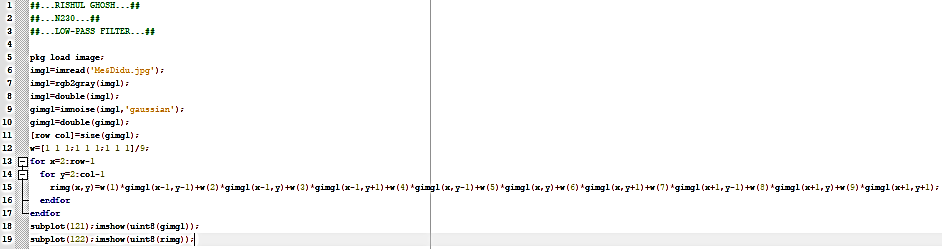
|  |  |
| --- | --- |
| **Roll No. :** N230 | **Name:** Rishul Ghosh |
| **Class :** MBA Tech CS 3rd Year Div. B | **Batch :** B |
| **Date of Experiment:** 18-8-21 | **Date of Submission:** |
| **Grade :** |  |

**B.1 Software Code written by student:**

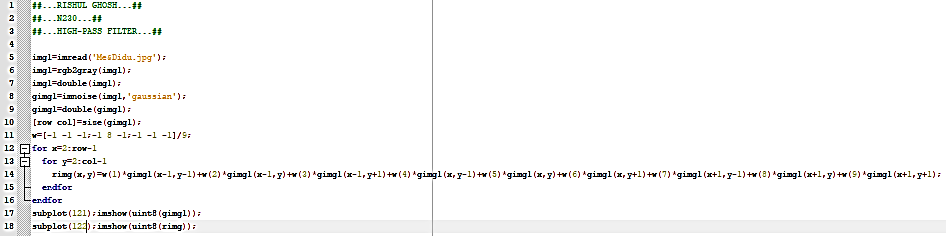
**TASK-1: ADDITION OF DIFFERENT TYPES OF NOISE TO IMAGE: -**



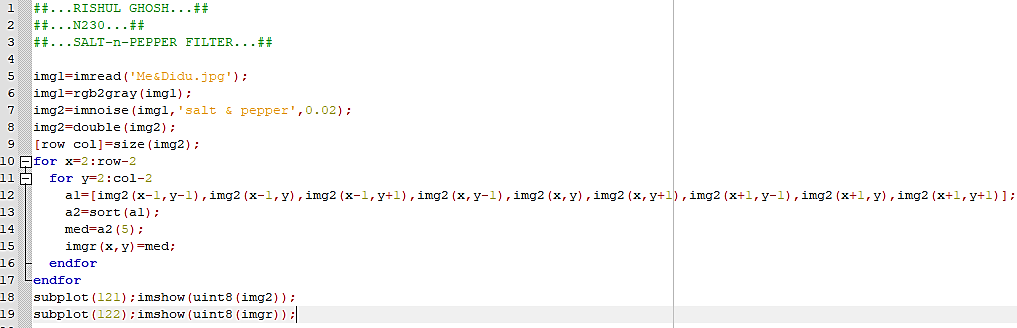
**TASK-2: LOW-PASS FILTER: -**



**TASK-3: HIGH-PASS FILTER: -**

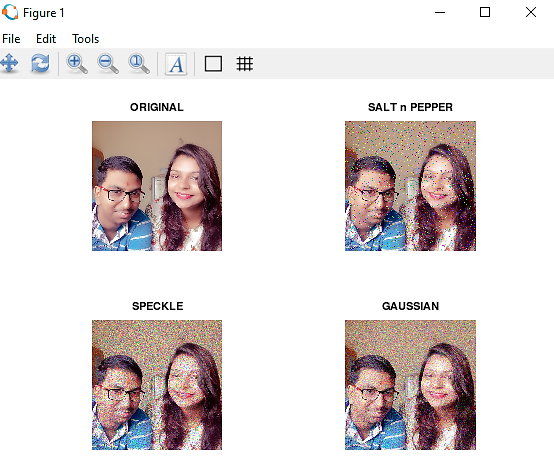


**TASK-4: FILTERING A SALT-n-PEPPER IMAGE: -**



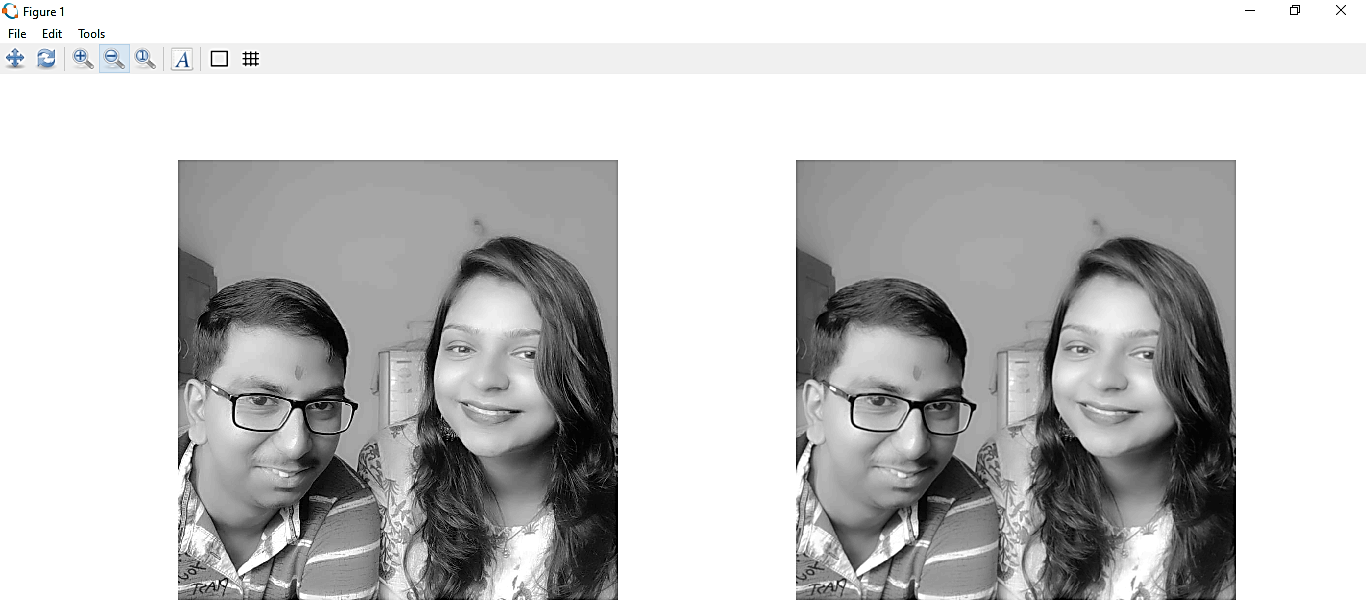
**B.2 Input and Output:**

**Input Images:**

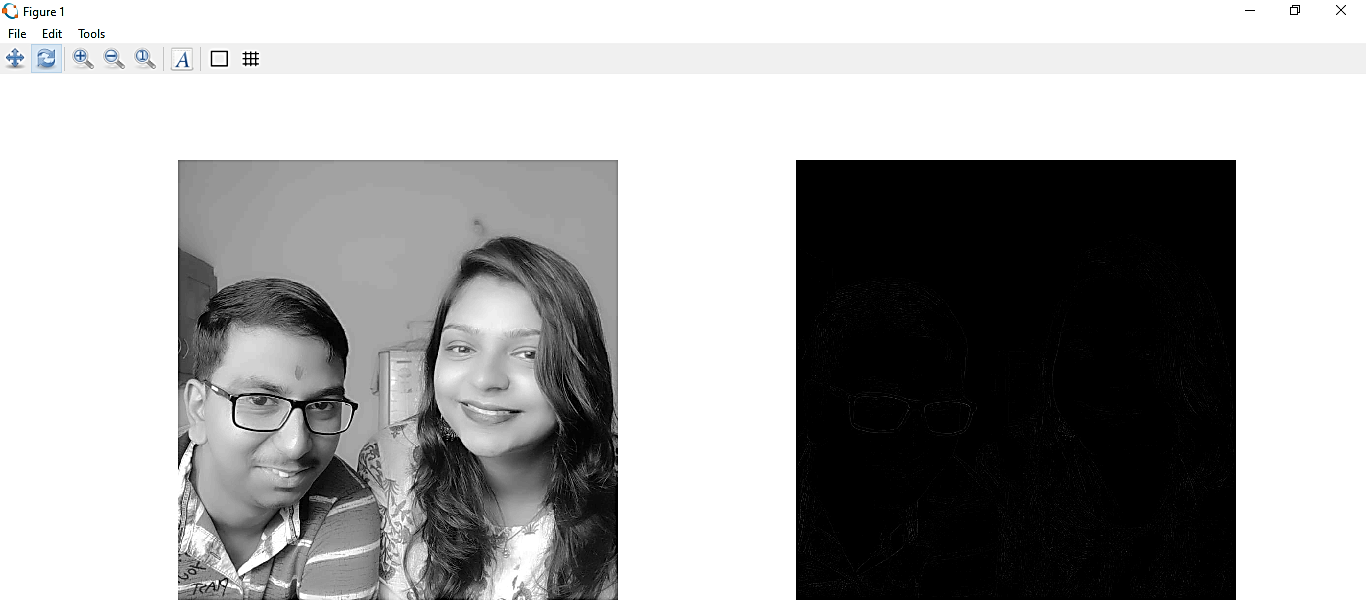


**Output Images:**

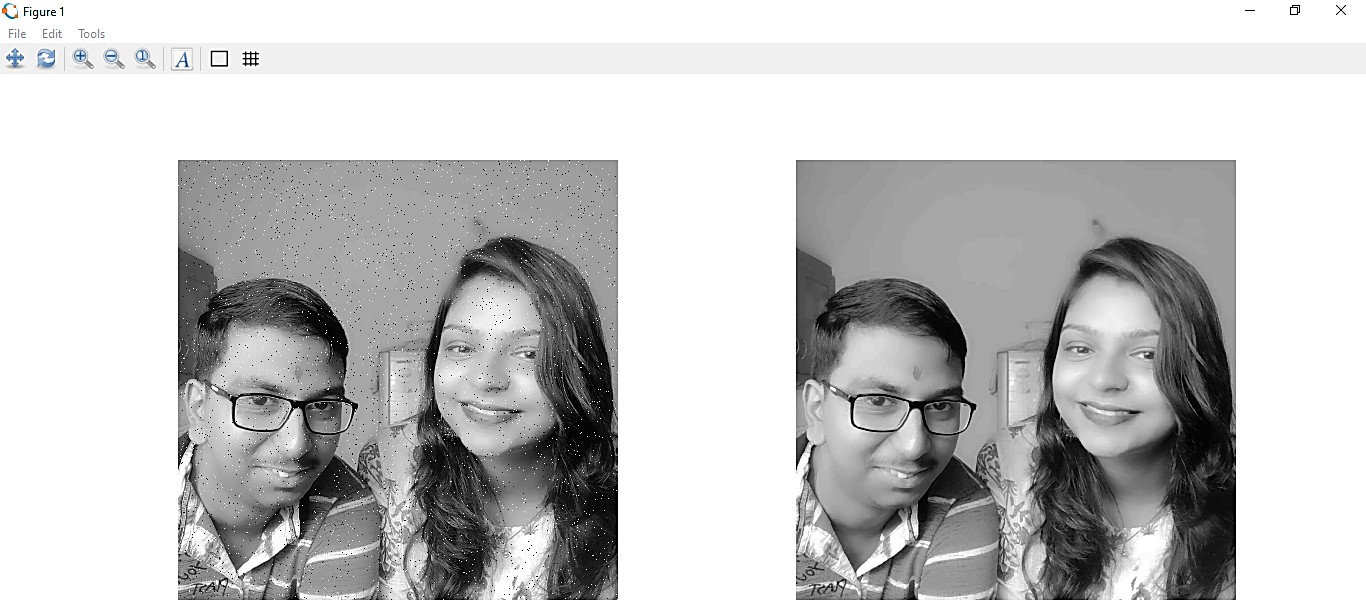
**TASK-2: LOW-PASS FILTER: -**



**TASK-3: HIGH-PASS FILTER: -**



**TASK-4: SALT-n-PEPPER FILTERING: -**



**B.3 Observations and learning:**

From the above experiment, we observed and learned various types of filters that are used to filter an image, during image processing. We also saw the various types of noises that an image can have. We implemented the same and observed the output into Octave software.

**B.4 Conclusion:**

Thus, the aim to write a program to enhance the quality of an image by noise removal, is completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*