Digital Image Processing Using Matlab Techniques and Applications

A mini project report submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

K. Radha Rani (S190145)

M. Chandra Rao (S190534)

M. Sirisha (S191059)

D. Sai Jagadeesh(S190023)

K. Yamuna (S190768)

N. Mounika (S190823)

Under the Guidance of

Ms. Y. Lakshmi Prasanna, M.Tech

Assistant Professor

Department of Electronics and Communication Engineering

Rajiv Gandhi University of Knowledge Technologies

CERTIFICATE

This is to certify that the work entitled "**Digital Image Processing Using Matlab Techniques and Applications**" is a bonafide record of authentic work carried out by S190145, S190534, S191059, S190023, S190768, S190823 under my supervision and guidance for the partial fulfilment of the requirements of the award of the degree of Bachelor of Technology in the department of **Electronics and Communication Engineering** at **RGUKT-SRIKAKULAM**.

The results embodied in this work have not been submitted to any other university or institute for the award of any degree or diploma. This thesis, in our opinion, is worthy of consideration for the award of the degree of Bachelor of Technology in accordance with the regulations of the institute.

----------------------------- ----------------------------

Project Guide	**Head of the Department**
Ms. Y. Lakshmi Prasanna, M.Tech	Mr. N. Ramesh Babu, M.Tech
Assistant Professor	Head of the Department
Department of ECE	Department of ECE

ACKNOWLEDGEMENT

The encomium goes to our project guide **Ms. Y. Lakshmi Prasanna.** We pay our deep sense of gratitude to our project supervisor for her outstanding guidance, motivation, and moral support whenever necessary during the period of project work.

I would like to express my appreciation to the entire team at the project. Their collaboration and willingness to share their knowledge and experiences have been invaluable to me.

Their constructive criticism and input have helped me refine my ideas and clarify my arguments. I'm truly fortunate to have such a supportive academic community.

ABSTARCT

Image processing plays a vital role in modern technology, enabling the extraction of valuable information from digital images for various applications. This project focuses on the implementation of image processing techniques using MATLAB to enhance feature extraction and analysis. The primary objectives include improving image quality, detecting and enhancing key features, and developing algorithms for efficient image-based decision-making.

The results of this project demonstrate the effectiveness of the proposed image processing techniques in improving the accuracy and efficiency of feature extraction and analysis.

The MATLAB-based implementations provide a user-friendly platform for researchers and practitioners to apply these methods to their specific domains.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Digital image processing stands as a cornerstone of modern technology, facilitating the extraction of valuable insights from digital images across various domains. This presentation is advanced image processing techniques implemented through MATLAB, aimed at augmenting feature extraction and analysis. Our primary focus lies in elevating image quality, pinpointing and enhancing critical features, and crafting algorithms conducive to efficient decision-making based on image data. Leveraging MATLAB as our platform of choice, our implementations offer a user-friendly avenue for researchers and practitioners alike to seamlessly integrate these techniques into their specific domains.

1.1 PURPOSE

- Image Enhancement: Improving the visual appearance of images to make features more detectable or visually pleasing. This includes techniques like contrast adjustment, noise reduction, and edge enhancement.
- Image Restoration: Correcting defects in images. This might involve removing blurs, correcting geometric distortions, or compensating for sensor noise.
- Image Compression: Reducing the amount of data required to represent an image. This is crucial for storage and transmission efficiency.
- Image Synthesis: Creating new images from existing data or models. This includes tasks like image generation, morphing, and texture synthesis.
- Visualization: Providing ways to visualize data and results. This is particularly useful in fields like medical imaging, remote sensing, and scientific research.
- Automation of Image Tasks: Automating repetitive or complex image processing tasks, making workflows more efficient and consistent.

MATLAB is particularly favored for these purposes due to its powerful built-in functions, extensive libraries, ease of use, and strong visualization capabilities making it a versatile tool for researchers, engineers, and practitioners in various field.

1.2 SCOPE

The scope of digital image processing in MATLAB is broad and encompasses various applications and fields, including:

- Medical Imaging:
  - Enhancement of medical images (MRI, CT scans, X-rays).
  - Automated diagnosis and analysis (tumor detection, blood flow analysis).
- Remote Sensing:
  - Satellite image analysis for environmental monitoring, agriculture, and urban planning.
  - Change detection in geographical areas over time.
  - Image classification for land cover mapping.
- Industrial Automation:
  - Quality control through defect detection in manufacturing.
  - Automated inspection systems.
  - Object recognition and tracking for robotics and assembly lines.

Key Features of MATLAB for Digital Image Processing:

- Toolboxes: MATLAB offers specialized toolboxes like the Image Processing Toolbox, which provides comprehensive algorithms and functions for processing images.
- Visualization: Robust tools for visualizing image data, which is essential for analysis and interpretation.
- Ease of Use: High-level language with built-in functions that simplify complex image processing tasks.
- Integration: Ability to integrate with other tools and languages, facilitating a more versatile development environment.
- Extensive Documentation and Community Support: Rich documentation and a large community help in troubleshooting and developing new applications.

Overall, the scope of digital image processing in MATLAB is extensive, providing powerful capabilities for a wide range of applications in various industries and research areas.

1.3 LANGUAGE USED

Key functions in the Image Processing Toolbox include image enhancement techniques, image restoration filters, Image segmentation algorithms, and feature extraction tools. These functions enable users to perform diverse image processing tasks with ease and precision.

Practical implementation of image processing techniques using functions from the Image Processing Toolbox. These examples showcase the step-by-step application of algorithms for tasks like image enhancement, restoration, and segmentation, aiding in understanding and utilization of MATLAB's image processing capabilities.

1.4 PROJECT REQUIREMENTS

Software Requirements: MATLAB Tool

CHAPTER-2

LITERATURE REVIEW

S. No	TITLE	JOUR-NAL	ADVANTAGES	DISADVANTAGES
1	Retinal vessels segmentation based on dilated multi-scale convolutional neural network	IEEE (2019)	High efficiency, accurate segmentation.	Large data requirement, overfitting.
2	Animal classification using facial images with score level fusion	IEEE (2018)	Ease of data collection, individual data collection.	Technical challenges, limited species applicability.
3	Complete blood check using image processing	IEEE (2021)	Speed, accuracy, cost effective is very high.	Complexity, interpretation challenges.
4	Sleep detection system using matlab image processing	IEEE (2023)	A non- invasive system for detecting the closing of eyes of a person.	Difficult to implement.

CHAPTER-3

PROBLEM DEFINITION

3.1 INTRODUCTION

Digital image processing is a critical technology with applications spanning various domains such as medical imaging, remote sensing, industrial automation, and multimedia. The primary goal of digital image processing is to improve the quality of images or extract meaningful information from them through computational techniques. MATLAB, with its robust computational environment and specialized toolboxes, provides a powerful platform for implementing and experimenting with various image processing algorithms.

3.2 EXISTING SYSTEM

Digital image processing using MATLAB is widely adopted in various fields, and several existing systems and frameworks have been developed to address different image processing needs. Here are some notable examples of existing systems

- MATLAB Image Processing Toolbox: The Image Processing Toolbox is a collection of functions and tools in MATLAB designed for image processing, analysis, visualization, and algorithm development. It provides a comprehensive suite of standard algorithms and functions for tasks
- Computer Vision Toolbox: The Computer Vision Toolbox extends MATLAB's capabilities to include advanced computer vision and video processing functions.
- Medical Image Processing: Several specialized systems and frameworks for medical image processing have been developed using MATLAB, leveraging both the Image Processing Toolbox and custom algorithms.
- Brain Tumor Detection: Systems that preprocess MRI images, segment brain tumors, and extract features for analysis and diagnosis.

These existing systems demonstrate the versatility and power of MATLAB for digital image processing, catering to a wide range of applications from basic image enhancement to advanced computer vision and medical diagnostics.

Disadvantages:

While MATLAB is a powerful tool for digital image processing, there are some disadvantages associated with existing systems that use MATLAB for this purpose. These disadvantages can impact aspects such as performance, flexibility, and usability. Here are some common drawbacks of existing digital image processing systems using MATLAB

- Computational Intensity: MATLAB, while efficient for many tasks, can become computationally intensive when processing large images or datasets. This can lead to slower processing times, especially for complex algorithms or real-time applications.
- Memory Consumption: Handling large images or datasets can consume significant memory resources in MATLAB, particularly when performing operations that involve multiple images or high-resolution images.
- Limited Performance with Large Data: MATLAB's performance can degrade when dealing with large-scale image datasets, as it may not fully leverage parallel processing capabilities or GPU acceleration without additional toolboxes or optimizations.

3.3 PROPOSED SYSTEM

Proposed System for Digital Image Processing Using MATLAB

- Title: Enhanced Real-Time Image Processing and Analysis System Using MATLAB
- Objective: To develop a MATLAB-based system that performs real-time image processing and analysis for various applications such as object detection, medical image analysis, and industrial inspection.
- Scope: The system will focus on:

    1. Real-time acquisition and preprocessing of images.
    2. Enhancement and noise reduction.
    3. Segmentation and feature extraction.
    4. Real-time object detection and classification.
    5. Visualization and user interaction.

- System Architecture: The proposed system will be modular, with each module handling a specific task in the image processing pipeline.

This proposed system aims to provide a comprehensive, real-time image processing and analysis solution using MATLAB, suitable for a wide range of applications.

Advantages:

The proposed system for digital image processing using MATLAB offers several advantages over existing systems, addressing common limitations and leveraging MATLAB's strengths in computational processing, algorithm development, and tool availability. Here are the key advantages of the proposed system:

- Comprehensive Toolset: MATLAB provides a comprehensive set of tools and functions in its Image Processing Toolbox and Computer Vision Toolbox. These tools cover a wide range of image processing tasks such as filtering, segmentation, feature extraction, and object detection, facilitating efficient algorithm development and implementation.
- Integration with MATLAB Environment: The proposed system benefits from seamless integration within the MATLAB environment, allowing easy access to advanced mathematical functions, plotting capabilities, and debugging tools. This integration enhances productivity and facilitates rapid prototyping and testing of image processing algorithms.
- Real-Time Processing Capabilities: MATLAB supports real-time image processing through efficient algorithm implementations and potential integration with GPUs for parallel processing. This capability is crucial for applications requiring immediate feedback, such as surveillance systems, medical diagnostics, and industrial automation.

CHAPTER-4

SYSTEM ANALYSIS

4.1 METHODOLOGY

The methodology of digital image processing in MATLAB involves several systematic steps, ensuring efficient and accurate analysis of images. Below is a detailed step-by-step outline:

- Problem Definition:

Objective Setting: Clearly define the goals of the image processing task, such as enhancement, segmentation, feature extraction, or classification.

Requirement Analysis: Determine the specific requirements for image quality, resolution, and processing speed.

- Image Acquisition:

Image Sources: Acquire images from cameras, scanners, medical imaging devices, or online databases.

Image Formats: Ensure images are in compatible formats (e.g., JPEG, PNG, TIFF) for processing in MATLAB.

- Image Importation:

Loading Images: Use MATLAB functions such as imread to load images into the MATLAB workspace.

4.2 OPERATION

1. **IMAGE ENHANCEMENT:**
   Image enhancement techniques play a crucial role in improving the visual quality and interpretability of digital images. Techniques like histogram equalization and contrast stretching optimize the contrast and brightness levels, bringing out hidden details. Filtering operations, including low-pass, high-pass, and band-pass filters, manipulate the image's frequency content to enhance specific features.

Fig – 4.2.1: Image Enhancement

❖ **IMAGE ENHANCEMENT TECHNIQUES:**

I. **Histogram Equalization:** Histogram equalization is a technique used to improve the contrast of an image by redistributing pixel intensities evenly across the histogram, enhancing the overall visibility of details and features.

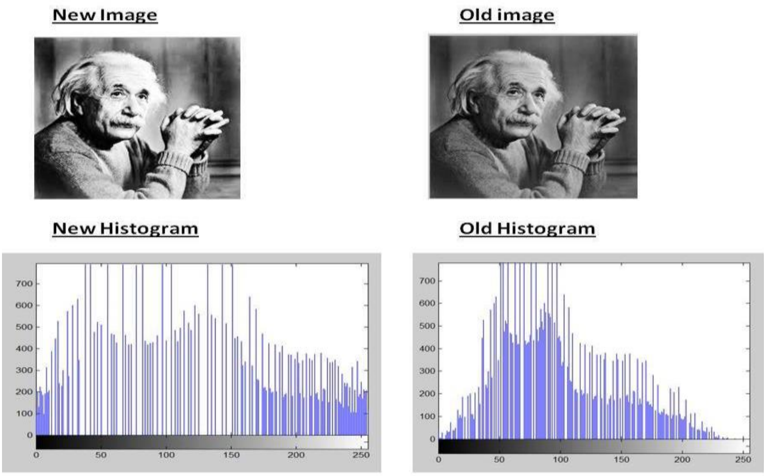


Fig – 4.2.2: Histogram Equalization

II. **Contrast Stretching:** Contrast stretching expands the range of pixel values in an Image, effectively Increasing the visual contrast between dark and light areas, leading to a more vibrant and detailed representation.

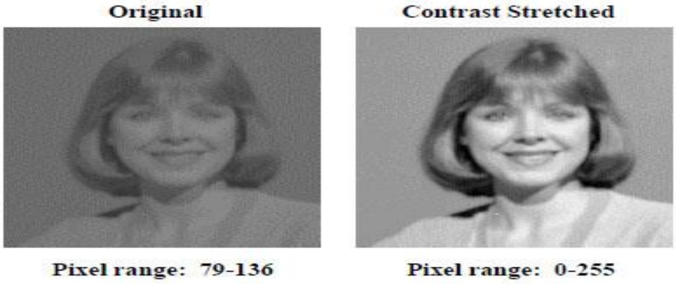


Fig – 4.2.3: Contrast Stretching

III. **Filtering (Low-pass, High-pass, Band-pass):**

Filtering techniques in image enhancement involve applying filters to modify the frequency content of an image. Low-pass filters smooth out variations, high-pass filters enhance edges and band-pass filters focus on specific frequency bands, each contributing to improving image quality.

Fig. 3: original image

Fig. 4: Result of Gaussian low pass filter

Fig. 5: Gaussian high pass filter

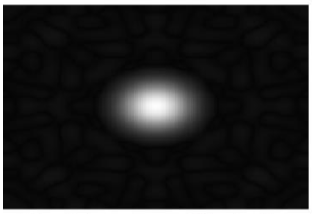


Fig. 6: Low pass filter in the frequency domain

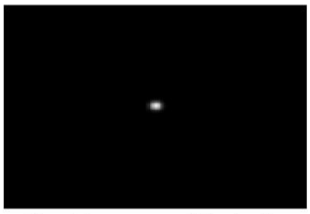


Fig. 7: Low pass filter in the spatial domain

Fig. 8: High pass filter in the frequency domain

Fig – 4.2.4: Filtering

IV. **Sharpening Filters:**

Sharpening filters enhance the details and edges in an image by emphasizing the high-frequency components, resulting in a crisper and more defined appearance of the objects within the image.

Fig – 4.2.5: Sharpening Filters

2. IMAGE RESTORATION:

Image restoration techniques focus on recovering the original quality of an image that has been degraded by noise or blurriness. Various noise reduction methods, such as filtering out Gaussian or salt-and- pepper noise, help in restoring image integrity.

Fig – 4.2.6: Image Restoration

❖ IMAGE RESTORATION TECHNIQUES:

I. **Noise Reduction Techniques (e.g., Gaussian, Salt and Pepper):** Noise reduction methods aim to remove unwanted disturbances in an image, such as Gaussian noise or salt-and- pepper noise, which can deteriorate image quality and affect visual interpretation. Techniques like smoothing filters and statistical methods help in restoring the original image clarity.

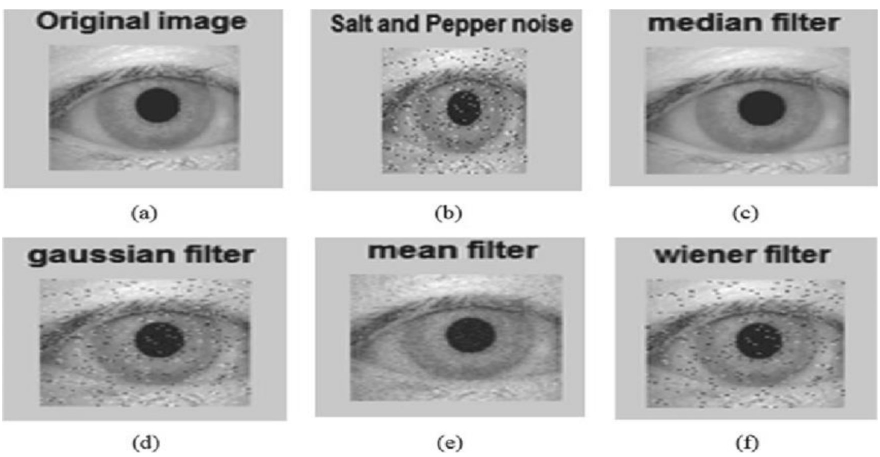


Fig – 4.2.7: Noise Reduction Technique

II. **Restoration Filters (e.g., Wiener Filter, Inverse Filter):**
Restoration filters like the Wiener filter and inverse filter are used to recover the degraded image caused by noise or blur. These filters utilize mathematical models to estimate and minimize noise effects, resulting in a cleaner and sharper image.

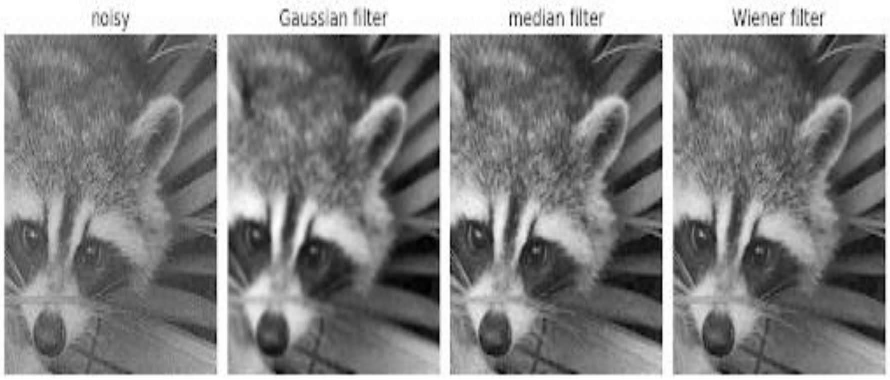
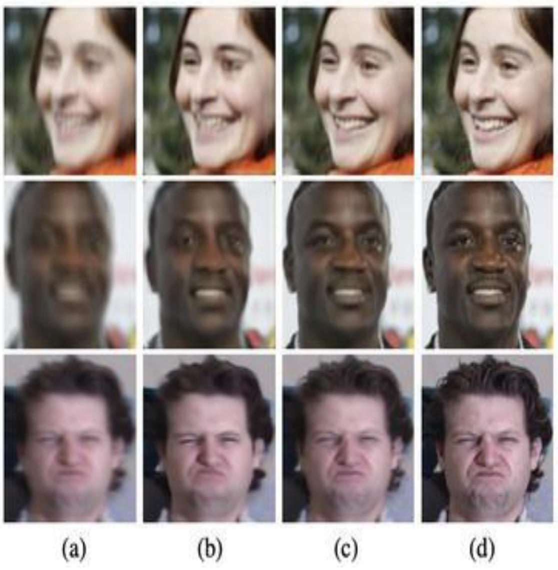


Fig – 4.2.8: Restoration filters

III. **Deblurring Techniques:**
Deblurring techniques address issues of image blurring caused by motion or optical distortions. By analyzing the degradation process, deblurring algorithms aim to reverse the effects and restore the image to its original sharpness and clarity.



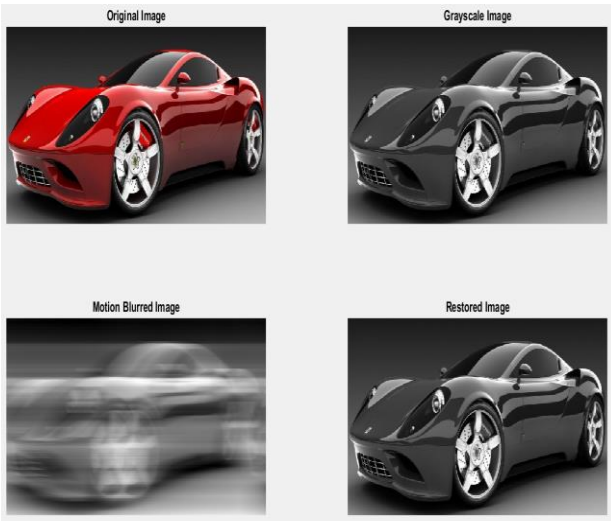


Fig – 4.2.9: Deblurring Techniques

3. COLOR IMAGE PROCESSING:

Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

Fig – 4.2.10: Color Image Processing

❖ COLOR IMAGE TECHNIQUES:

I. Color Enhancement: Adjusting the color balance, contrast, brightness, and saturation to improve the overall appearance of the image.

II. Color Filtering: Applying filters in specific color channels to enhance or suppress certain colors in the image.

III. Color Segmentation: Dividing an image into regions based on color similarity, which is useful for object detection and tracking.

IV. Color Histogram Equalization: Equalizing the color histograms to improve the overall contrast and brightness distribution of the image.

V. Color Edge Detection: Detecting edges based on color gradients to identify boundaries between different objects or regions in the image.

4.3 IMPLEMENTATION

The implementation of a digital image processing system in MATLAB involves several steps, from setting up the environment to executing specific image processing tasks. Here’s a step-by-step guide to implementing a comprehensive digital image processing system using MATLAB.

Step 1: Set Up the Environment

Ensure you have MATLAB installed with the necessary toolboxes:

- Image Processing Toolbox
- Computer Vision Toolbox (optional, for advanced tasks)

Step 2: Acquire an Image

Capture an image using a webcam or load an existing image from a file.

Step 3: Pre-process the Image

Convert the image to grayscale, reduce noise, and enhance contrast.

Step 4: Enhance the Image

Apply further enhancements like sharpening.

Step 5: Segment the Image

Segment the image to identify objects or regions of interest.

Step 6: Extract Features

Extract relevant features from the segmented image.

Step 7: Detect and Classify Objects

Use a pre-trained deep learning model for object detection and classification.

Step 8: Visualize the Results

Display the results of the processing steps.

Step 9: Implement Real-time Processing

Combine all the steps in a real-time processing loop.

Step 10: Optimize and Deploy

- Optimization: Optimize the code for performance, especially if real-time processing is required. This can include using GPU acceleration, reducing computational complexity, and optimizing algorithm parameters.

-Deployment: Package the application for deployment. This can involve creating a standalone executable using MATLAB Compiler.

This implementation outlines the key steps required to develop a robust digital image processing system using MATLAB.

CHAPTER-5

CODE

5.1 Code for basic image operations:

```
clc;
clear all;
close all;
a=imread('C:\Users\MALAPALLI CHANDRARAO\dipimage.jpg');
subplot(2,3,1);
imshow(a);
b=rgb2gray(a);
subplot(2,3,2);
imshow(b);
c=im2bw(a);
subplot(2,3,3);
imshow(c);
d=imadjust(b);
subplot(2,3,4);
imshow(d);
e=a;
e=rgb2gray(e);
subplot(2,3,5);
imhist(e);
```

5.2 DEBLURRING OF IMAGE

```
clc;
close all;
y = imread('cameraman.tif');
y = im2double(y);
figure, imshow(y);
PSF = fspecial('disk', 8);
yblur = conv2(y, PSF);
figure, imshow(yblur);
%wiener deconvolution
x = deconvwnr(yblur, PSF, 0.005);
figure, imshow(x)
%using blind deconv
psf2 = fspecial('disk', 5.4);
x = deconvblind (yblur, PSF, 18);
figure, imshow(x)
```

5.3 IMAGE REDUCTION TECHNIQUE

```
%Read image
Image=imread('coins.png');
subplot(2,2,1),imshow(Image)
title('Original Image');
%%Add salt and pepper noise
%Take input from user
Noise_level=input('How much noise you want to add? (eg input 0.1 for
10%):');
Noisy_image = imnoise(Image, 'salt & pepper', Noise_level);
subplot(2,2,2),imshow(Noisy_image)
title('Corrupted with salt and pepper noise');
%3-by-3 averaging filter
Output_avg = filter20(fspecial('average',3), Noisy_image)/255;
subplot(2,2,3),imshow(Output_avg)
title('Image filtered by Averaging filter');
%3-by-3 Median filter
Output_med = medfilt2(Noisy_image, [3 3]);
subplot(2,2,4),imshow(Output_med)
title('Image filtered by Median filter');
```

5.4 RESTORATION OF IMAGE BY USING WIENER FILTER

```
close all
clear
clc
A=imread('C:\Users\MALAPALLI CHANDRARAO\dipimage.jpg');
A=double(rgb2gray(A));
[M,N]=size(A);
h=ones(5,5)/25;
Freq_a=fft2(A);
Freq_h=fft2(h,M,N);
B=real(ifft2(Freq_h.*Freq_a))+25*randn(M,N);
Freq_b=fft2(B);
pow_b=abs(Freq_b).^2/(M*N);
sigma=50;
gamma=1;
alpha=1;
sFreq_h=Freq_h.*(abs(Freq_h)>0)+1/gamma*(abs(Freq_h)==0);
iFreq_h=1./sFreq_h;
iFreq_h=iFreq_h.*(abs(Freq_h)*gamma>1)+gamma*abs(sFreq_h).*iFreq_h
.*(abs(sFreq_h)*gamma<=1);
pow_b=pow_b.*(pow_b>sigma^2)+sigma^2*(pow_b<=sigma^2);
Freq_g=iFreq_h.*(pow_b-sigma^2)./(pow_b-(1-alpha)*sigma^2);
Res_Freq_a=Freq_g.*Freq_b;
Res_a=real(ifft2(Res_Freq_a));
imshow(uint8(A));title('original image')
figure,imshow(uint8(B)),title('degraded image')
figure,imshow(uint8(Res_a)),title('restored image after wiener filter')
```

5.5 CODE FOR COLOR IMAGE PROCESSING

```
close all;
clear all;
clc;
warning off;
A=imread('C:\Users\MALAPALLI CHANDRARAO\dipimage.jpg');
subplot(2,4,1)
imshow(A);
title('original image');
```

```
B=imadd(A,100);
subplot(2,4,2);
imshow(B);
title('intensity added image');

C=imcomplement(A);
subplot(2,4,3)
imshow(C);
title('complement image’);
%subplot(2,4,4)
%imshow(D);
%title('Histogram equilized image');

A=im2double(A);
red_ch1=A(:,:,1);
green_ch1=A(:,:,2);
blue_ch1=A(:,:,3);
Amp_Factor=1.33;
red_ch1=(red_ch1*Amp_Factor);
rgbImage=cat(3,red_ch1,green_ch1,blue_ch1);
subplot(2,4,5)
imshow(rgbImage);
title('color increased image');

w=fspecial('average',[5 5]);
red=A(:,:,1);
green=A(:,:,2);
blue=A(:,:,3);
R=imfilter(red,w);
G=imfilter(green,w);
B=imfilter(blue,w);
RGB=cat(3,R,G,B);
subplot(2,4,6);
imshow(RGB);
title('each rgb LPF image');

M=rgb2hsv(A);
Hue=M(:,:,1);
Saturation=M(:,:,2);
Value=M(:,:,3);
```

```
DD=imfilter(Value,w);
HSV_cat=cat(3,Hue,Saturation,DD);
RGB_cat=hsv2rgb(HSV_cat);
subplot(2,4,7)
imshow(RGB_cat);
title('only inetensity of HSI LPF image');
diff_image=imsubtract(RGB,RGB_cat);
subplot(2,4,8);
imshow(diff_image);
title('difference image');
```

5.6 CODE FOR IMAGE ENHANCEMENT

Enhancement of Contrast And Histogram Equalization

```
i=imread('pout.tif’);

figure;

subplot(1,2,1);

imshow(i);

subplot(1,2,2);

imhist(i);

imh=imadjust(i,[0.3,0.6],[0.0,1.0]);

imh1=histeq(i);

figure;subplot(2,2,1);

imshow(imh);

title('stretched’);

subplot(2,2,2);

imhist(imh);

subplot(2,2,3);

imshow(imh1);

title('hist eq’);

subplot(2,2,4);

imhist(imh1);
```

5.7 SHARPENING OF IMAGE

```
a=imread('moon.tif');
w=fspecial('laplacian',0);
w2=[0 -1 0; -1 4 -1; 0 -1 0];
w3=[-1 -1 -1; -1 8 -1; -1 -1 -1];
d=imfilter(a,w2,'replicate');
d2=imfilter(a,w3,'replicate');
figure,imshow(d)
figure,imshow(d2)
b=a+d;
c=a+d2;
figure,imshow(b)
figure,imshow(c)
figure,imshow(a)
w4=[0 -1 0; -1 5 -1; 0 -1 0];
r=imfilter(a,w4,'replicate');
figure,imshow(r)
```

CHAPTER-6

OUTPUT

6.1 BASIC OUTPUT

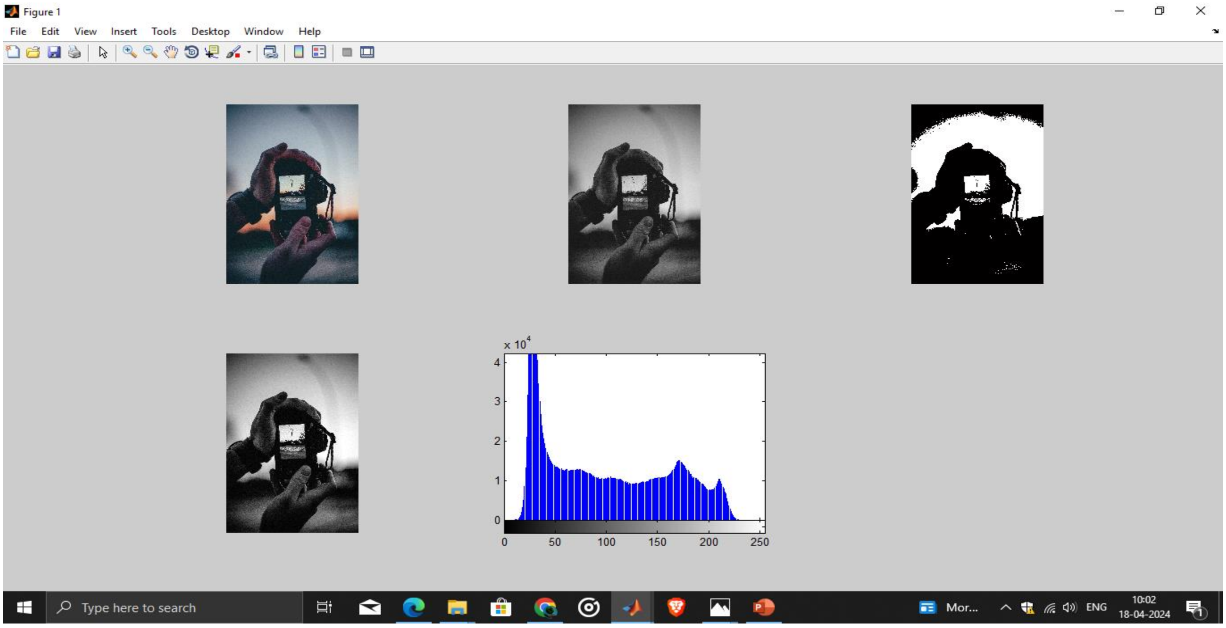


Fig – 6.1: Basic Output

6.2 BASIC OPERATIONS USING GUI

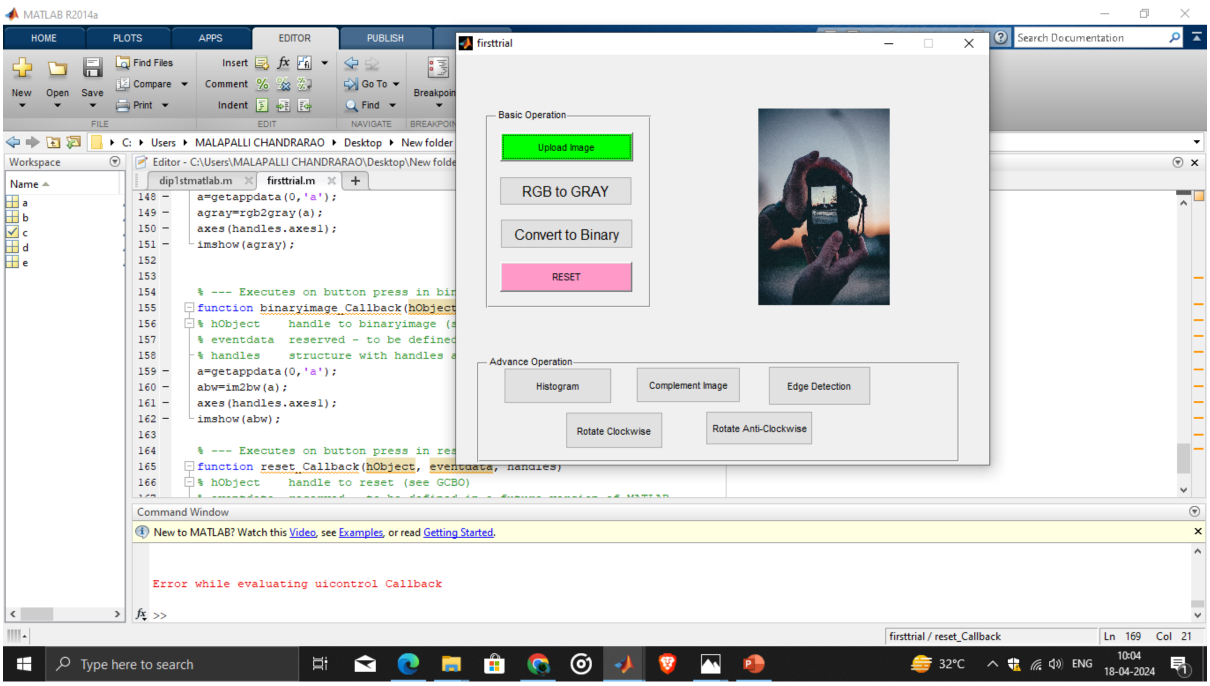


Fig – 6.2: Basic operations using GUI

6.3 COLOR IMAGE

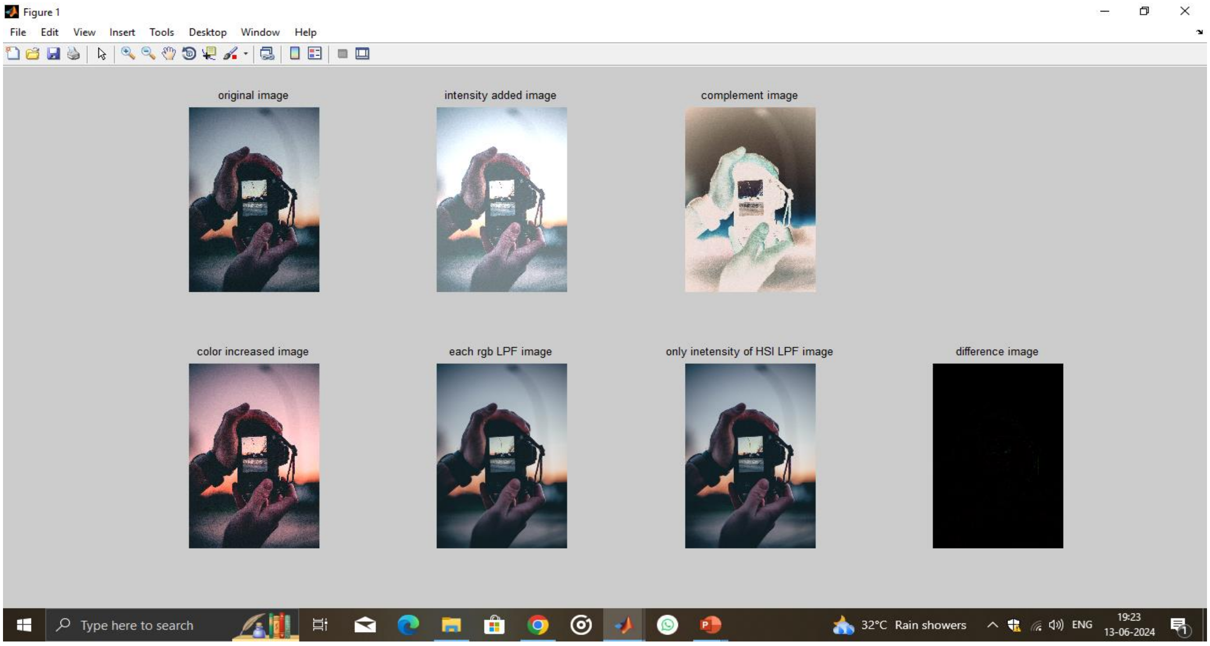


Fig – 6.3: Color Image

6.4 DEBLURRING

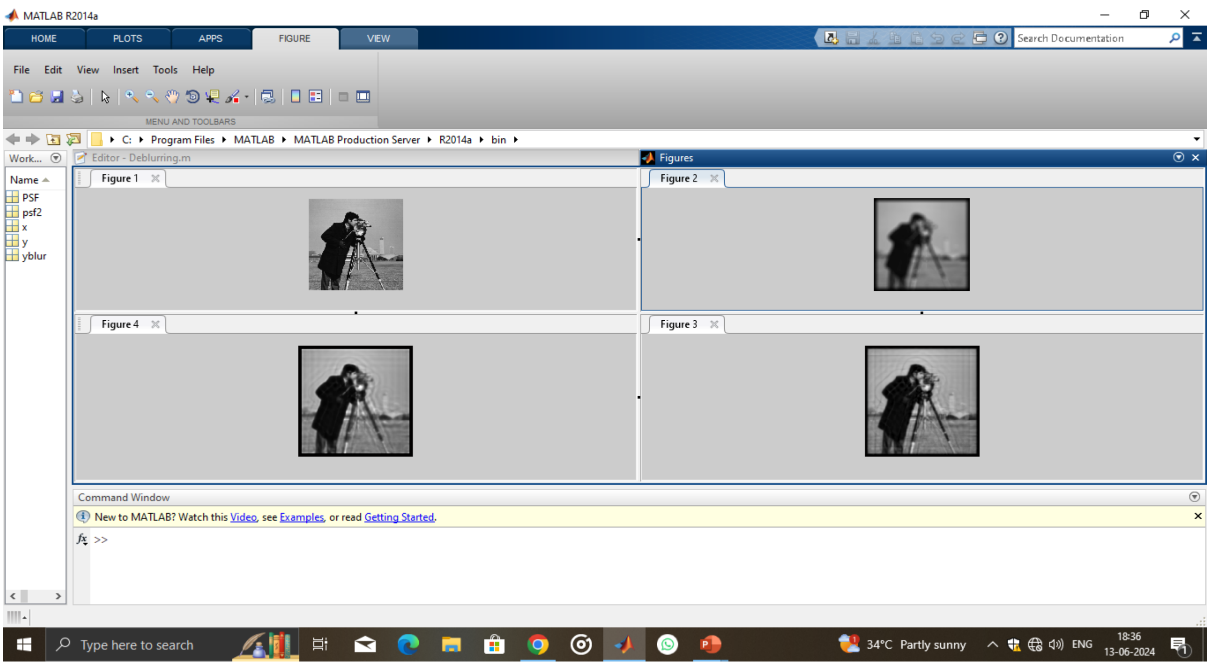


Fig – 6.4: Deblurring

6.5 WIENER FILTER

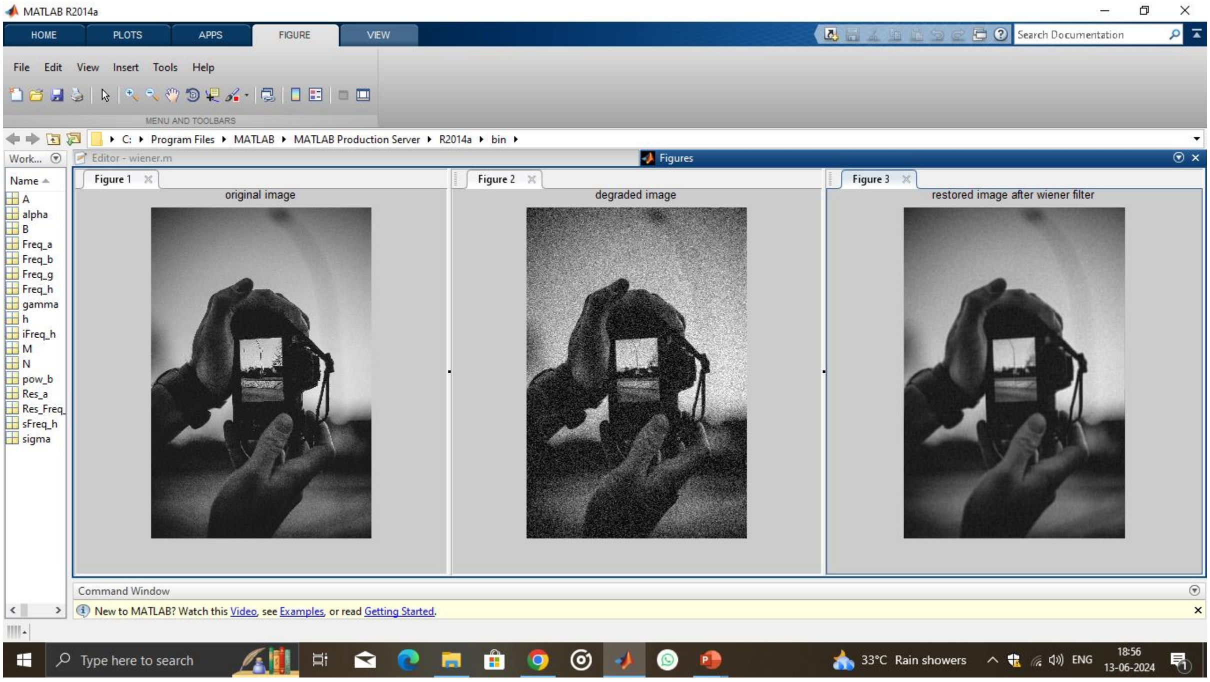


Fig – 6.5: Wiener Filter

6.6 REDUCTION TECHNIQUES

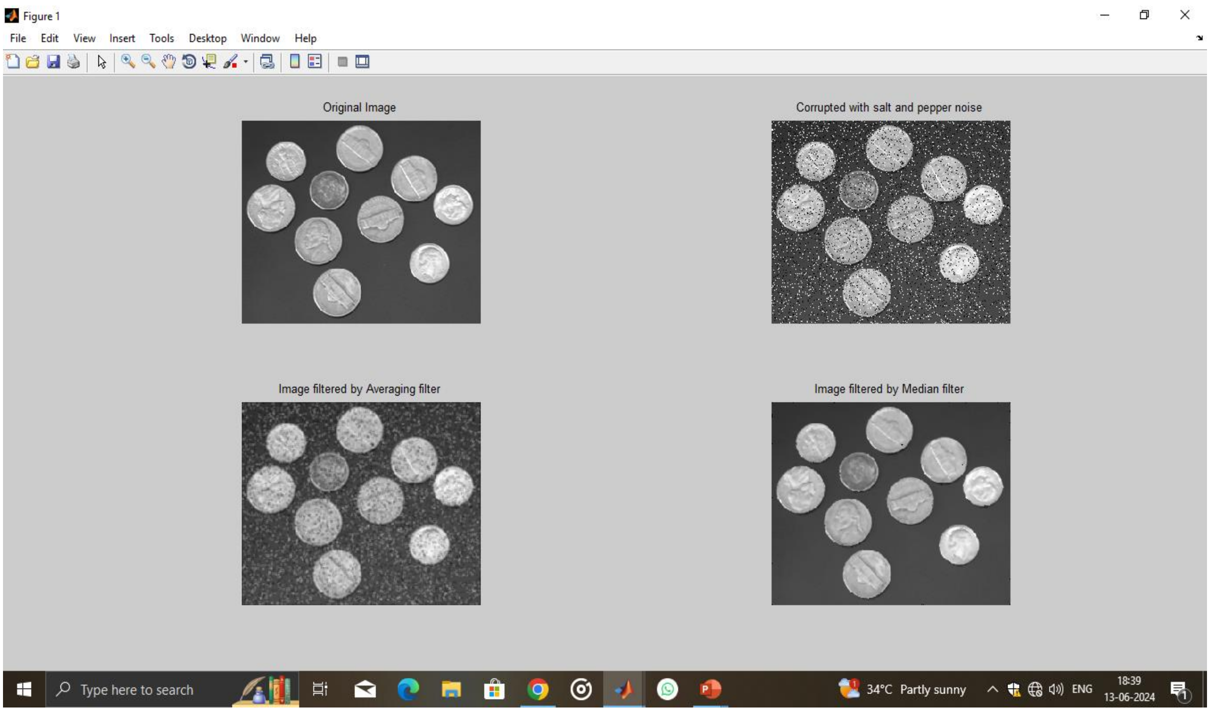


Fig – 6.6: Reduction Techniques

6.7 ENHANCED CONTRAST & HISTOGRAM

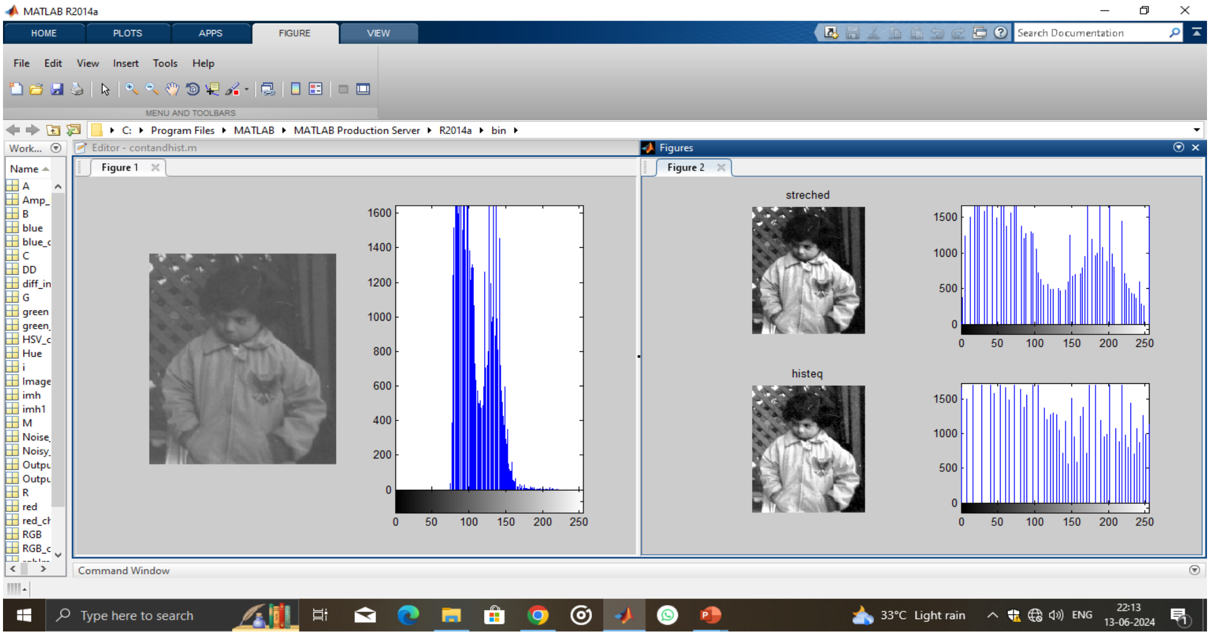


Fig – 6.7: Enhanced contrast & Histogram

6.8 SHARPENING IMAGE

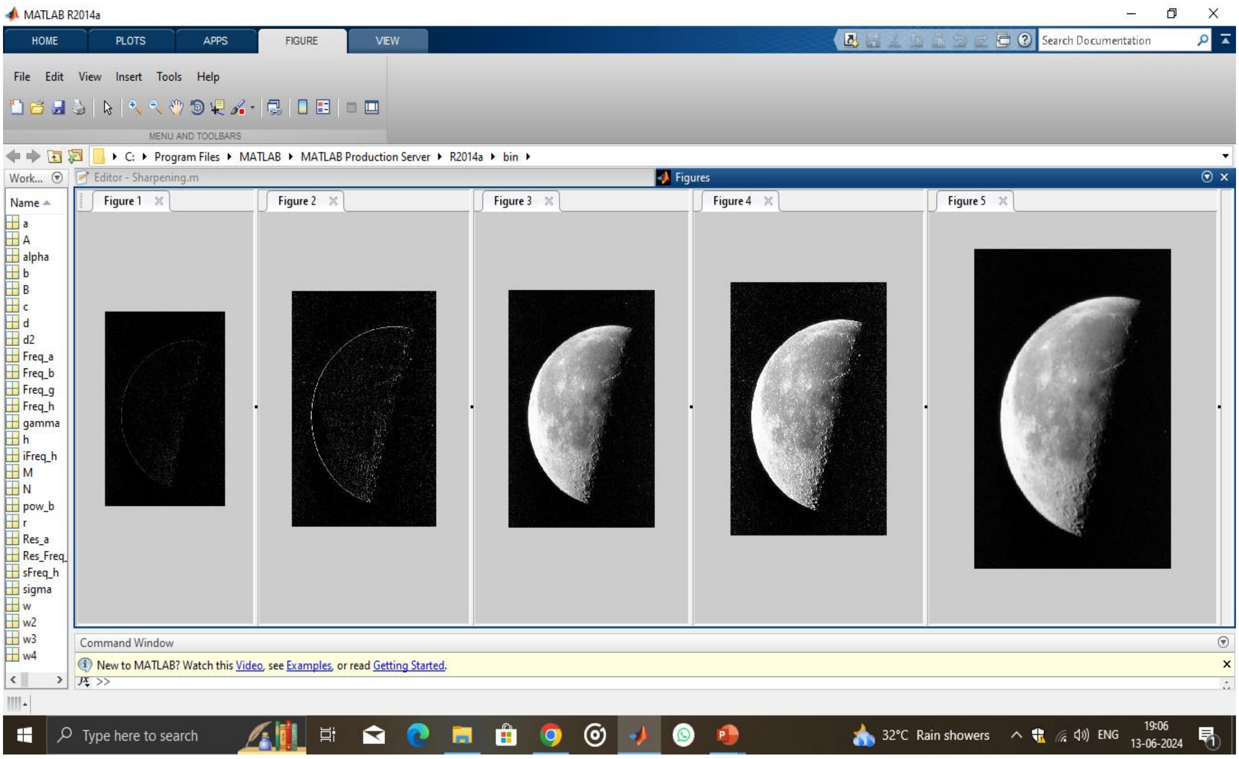


Fig – 6.8: Sharpening Image

CHAPTER-7

FUTURE SCOPE

Image processing has come a long way but the future holds even more exciting prospects. With the advent of artificial intelligence, augmented reality, and advancements in various industries, image processing will play a vital role in transforming the way we perceive and interact with the world.

CHAPTER-8

CONCLUSION

The image processing techniques in MATLAB unveils a dynamic landscape of tools and methods that empower users to manipulate and analyze digital images for various purposes. From enhancing image quality to extracting meaningful features, each technique offers a unique perspective on visual data processing, showcasing the immense potential and versatility of image processing in unlocking insights and generating impactful results.

CHAPTER-9

REFERENCES

- S. K. Dewangan, "Importance & Applications of Digital Image Processing," International Journal of Computer Science & Engineering Technology (IJCSET), vol. 7, no. 7, pp. 316-320, 2016.
- D. Zhang, F. Wang, R. Burgos, and D. Boroyevich, "Common mode circulating current control of interleaved three-phase two-level voltage-source converters with discontinuous space-vector modulation," 2009 IEEE Energy Conversion Congress and Exposition, 2009, pp. 2801-2807, doi: 10.1109/ECCE.2009.5316313.
- R. S. Kavita, R. Bala, and S. Siwach, "Review paper on overview of image processing and image segmentation, " International journal of Research in Computer applications and Robotics, vol. 1, no. 7, 2013.
- H. Zhang, Z. Zhang, and Z. Pei, "Design and Implementation of Image Processing System Based on MATLAB," in International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2015), 2015, pp. 1356-1359, doi: 10.2991/lemcs-15.2015.270.
- R. Buksh, S. Routh, P. Mitra, S. Banik, A. Mallik, and S. Gupta, "MATLAB based Image Editing and Color Detection," International Journal of Scientific and Research Publications, vol. 4, no. 1, pp. 1-6, 2014.