Fundamentals of Robotics

Extra Tutorials (CA)

Aadi Singh Chauhan D12C_9

Code

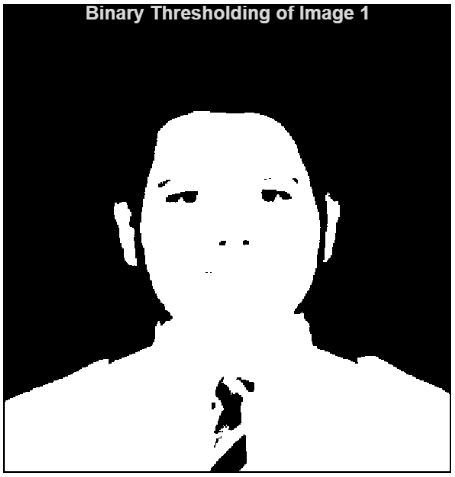
```
% ---------- Folder to save results
----------
outDir = fullfile(pwd,
'Output_Results');
if ~exist(outDir, 'dir')
   mkdir(outDir);
end
% ---------- Select input images
----------
[fn1, fp1] =
uigetfile({'.jpg;.png;.jpeg;.bmp'},
'Select Image 1 (Your Photo)');
I1 = imread(fullfile(fp1, fn1));
[fn2, fp2] =
uigetfile({'.jpg;.png;.jpeg;.bmp'},
'Select Image 2 (You +
Background)');
I2 = imread(fullfile(fp2, fn2));
% ---------- Helper function to save
image with title ----------
saveIm = @(img, name, titleStr)
saveWithTitle(img, fullfile(outDir,
name), titleStr);
%
====================================
========================
% 1. Simulation & Negative of Image
(Binary & Gray)
%
====================================
========================
I1_gray = im2gray(I1);
I1_neg_gray = imcomplement(I1_gray);
I1_binary = imbinarize(I1_gray);
saveIm(I1_gray, '01_I1_gray.png',
'Grayscale Image 1');
saveIm(I1_neg_gray,
'01_I1_negative_gray.png',
'Grayscale Negative of Image 1');
saveIm(I1_binary,
'01_I1_binary.png', 'Binary
Thresholding of Image 1');
%
====================================
========================
% 2. Relationships between Pixels
%
====================================
========================
% Pixel difference
I2_gray = im2gray(I2);
rel = imabsdiff(imresize(I1_gray,
size(I2_gray)), I2_gray);
saveIm(rel,
'02_pixel_relationship.png',
'Absolute Pixel Difference between
I1 & I2');
%
====================================
========================
% 3. Transformations of Image
%
====================================
========================
I1_rot = imrotate(I1, 30,
'bilinear', 'crop');
I1_trans = imtranslate(I1, [50 30]);
saveIm(I1_rot,
'03_I1_rotate_30deg.png', 'Rotation
by 30° of Image 1');
saveIm(I1_trans,
'03_I1_translate.png', 'Translation
of Image 1');
```

```
%
====================================
========================
% 4. Contrast Stretching & Histogram
Equalization
%
====================================
========================
I1_stretch = imadjust(I1_gray,
stretchlim(I1_gray), []);
I1_histeq = histeq(I1_gray);
saveIm(I1_stretch,
'04_I1_contrast_stretch.png',
'Contrast Stretched Image 1');
saveIm(I1_histeq,
'04_I1_hist_eq.png', 'Histogram
Equalized Image 1');
% Histogram Plot
fig = figure('Visible', 'off');
imhist(I1_gray); title('Histogram of
Image 1');
saveas(fig, fullfile(outDir,
'04_I1_histogram.png')); close(fig);
%
====================================
========================
% 5. Bit Plane Slicing
%
====================================
========================
for k = 1:8
   bit_plane = bitget(I1_gray, k) *
255;
   saveIm(uint8(bit_plane),
sprintf('05_I1_bitplane_%d.png', k),
sprintf('Bit Plane %d of Image 1',
k));
end
%
====================================
========================
% 6. FFT (1D & 2D)
%
====================================
========================
I2d = double(I2_gray);
F2 = fft2(I2d);
F2shift = fftshift(F2);
magF2 = log(1 + abs(F2shift));
saveIm(mat2gray(magF2),
'06_I2_fft2d_magnitude.png', '2D FFT
Magnitude Spectrum of Image 2');
% 1D FFT (using a middle row)
rowSignal =
double(I2_gray(round(end/2), :));
F1 = fft(rowSignal);
magF1 = abs(F1);
fig = figure('Visible', 'off');
plot(magF1); title('1D FFT Magnitude
of Middle Row');
saveas(fig, fullfile(outDir,
'06_I2_fft1d.png')); close(fig);
%
====================================
========================
% 7. Mean, Std. Deviation,
Correlation
%
====================================
========================
meanI1 = mean2(I1_gray);
stdI1 = std2(I1_gray);
% Correlation computation (robust)
targetRows = min(size(I1_gray,1),
size(I2_gray,1));
targetCols = min(size(I1_gray,2),
size(I2_gray,2));
I1r = imresize(I1_gray, [targetRows
targetCols], 'bilinear');
I2r = imresize(I2_gray, [targetRows
targetCols], 'bilinear');
v1 = double(I1r(:));
v2 = double(I2r(:));
if all(v1 == v1(1)) || all(v2 ==
v2(1))
   corr_val = NaN;
else
   C = corrcoef(v1, v2);
   corr_val = C(1,2);
end
fprintf('Image 1 Mean: %.2f, Std:
%.2f, Corr(I1,I2): %.4f\n', meanI1,
stdI1, corr_val);
%
====================================
========================
% 8. Smoothening Filters (Mean,
Median)
```

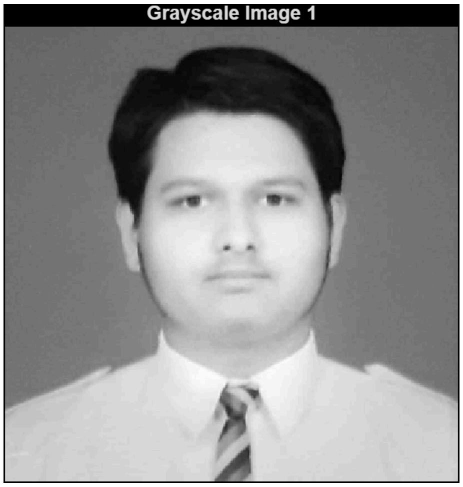
```
%
====================================
========================
meanf = imfilter(I2_gray,
fspecial('average', [5 5]));
medf = medfilt2(I2_gray, [5 5]);
saveIm(meanf,
'08_I2_mean_filter.png', 'Mean
Filtered Image (5x5)');
saveIm(medf,
'08_I2_median_filter.png', 'Median
Filtered Image (5x5)');
%
====================================
========================
% 9. Sharpening & Edge Detection
(Gradient)
%
====================================
========================
sharp = imsharpen(I2_gray);
edges_sobel = edge(I2_gray,
'sobel');
saveIm(sharp, '09_I2_sharpen.png',
'Sharpened Image 2');
saveIm(edges_sobel,
'09_I2_edges_sobel.png', 'Sobel Edge
Detection of Image 2');
%
====================================
========================
% 10. Image Restoration (Wiener
Filter)
%
====================================
========================
I2_noisy = imnoise(I2_gray,
'gaussian', 0, 0.005);
I2_restored = wiener2(I2_noisy, [5
5]);
saveIm(I2_noisy, '10_I2_noisy.png',
'Gaussian Noisy Image');
saveIm(I2_restored,
'10_I2_restored.png', 'Restored
Image using Wiener Filter');
%
====================================
========================
% 11. Intensity Slicing
%
====================================
========================
I1_slice = I1_gray;
I1_slice(I1_slice > 80 & I1_slice <
150) = 255;
saveIm(I1_slice,
'11_I1_intensity_slice.png',
'Intensity Slicing Enhancement');
%
====================================
========================
% 12. Canny Edge Detection
%
====================================
========================
edges_canny = edge(I1_gray,
'canny');
saveIm(edges_canny,
'12_I1_canny_edges.png', 'Canny Edge
Detection');
disp('✅ All processed images saved
in:');
disp(outDir);
%
====================================
========================
% --- Helper Function: Save Image
with Title Below ---
%
====================================
========================
function saveWithTitle(img,
filePath, titleStr)
   fig = figure('Visible','off');
   imshow(img, []);
   title(titleStr, 'FontSize', 12,
'FontWeight', 'bold', 'Interpreter',
'none');
   set(gca, 'LooseInset', get(gca,
'TightInset'));
   exportgraphics(gca, filePath,
'Resolution', 150);
   close(fig);
end
```

Take a photo in which you are the sole subject and perform the following:

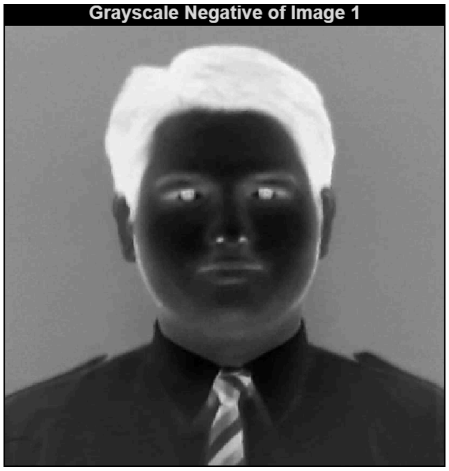
1. Simulation and Display of an Image, Negative of an Image
(Binary & Gray Scale)



Binary Thresholding of Image 1

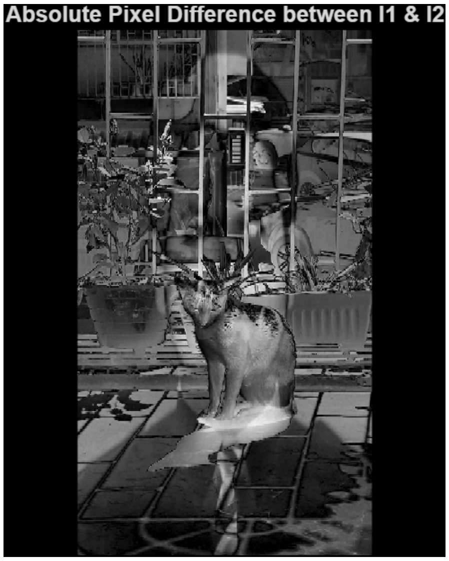


Grayscale Image 1



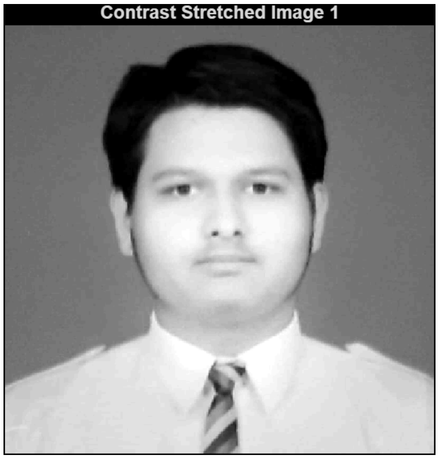
Grayscale Negative of Image 1

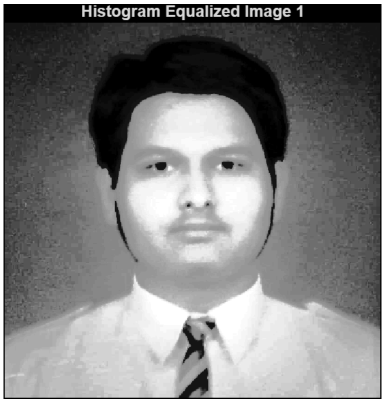
2. Implementation of Relationships between Pixels

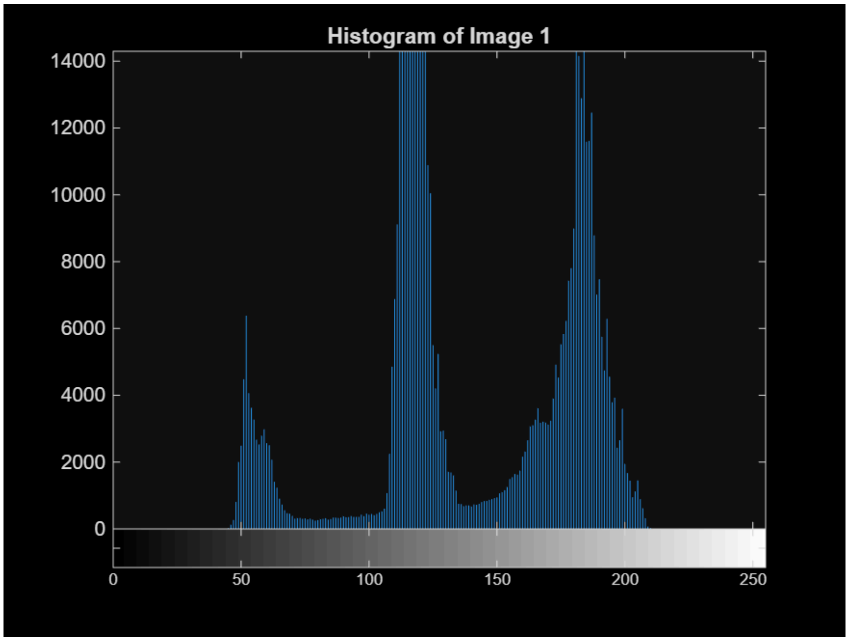


3. Implementation of Transformations of an Image

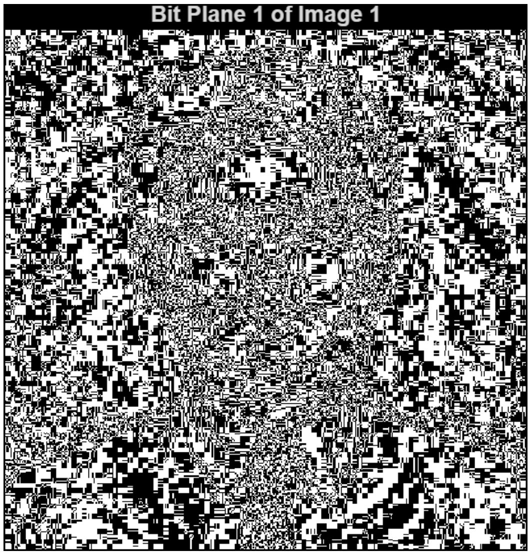
4. Contrast stretching of a low contrast image, Histogram, and Histogram Equalization

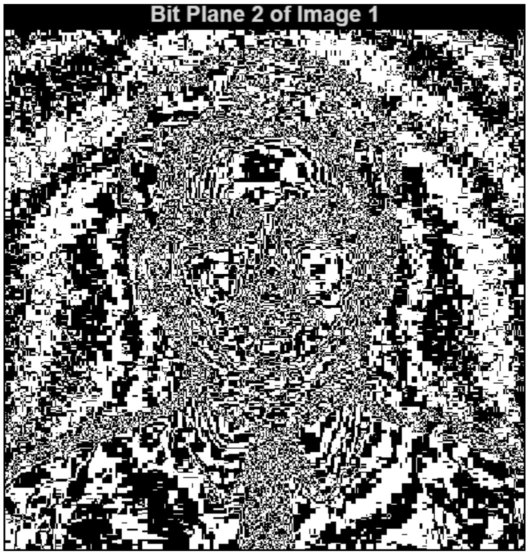


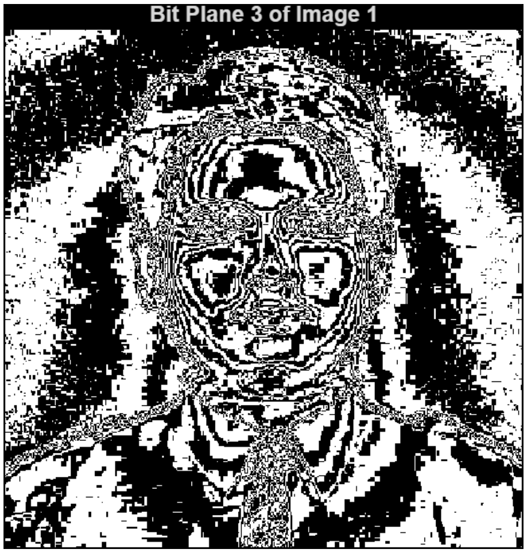


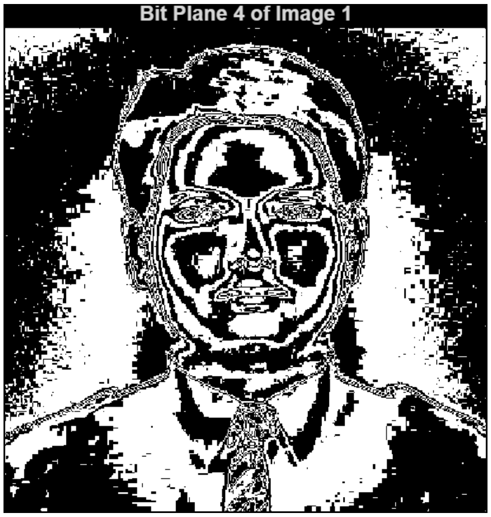


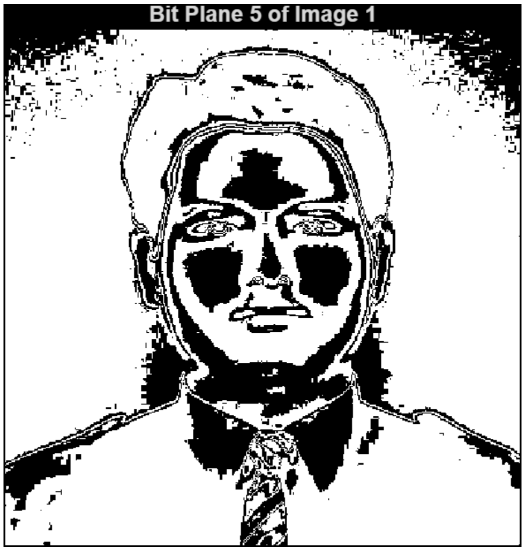
5. Display of bit planes of an Image

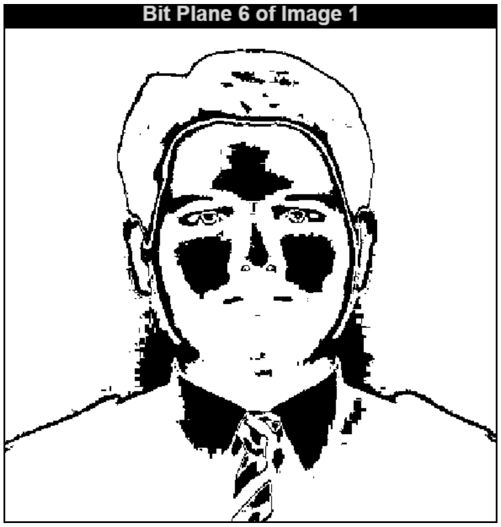

Bit Plane 1 of Image 1

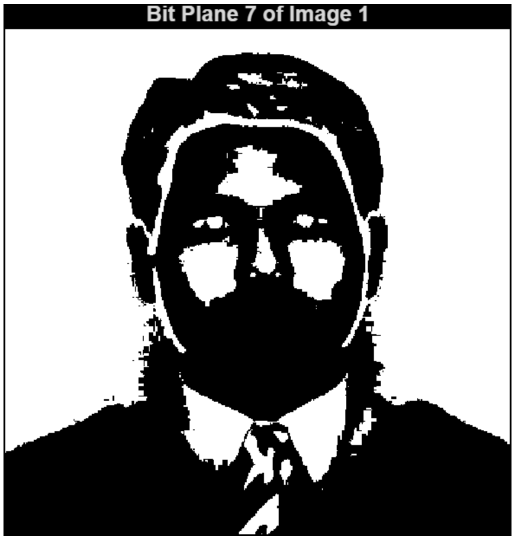

Bit Plane 2 of Image 1


Bit Plane 3 of Image 1

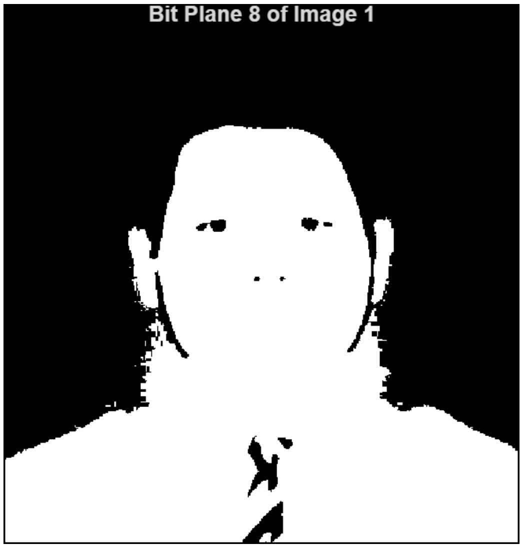

Bit Plane 4 of Image 1


Bit Plane 5 of Image 1


Bit Plane 6 of Image 1

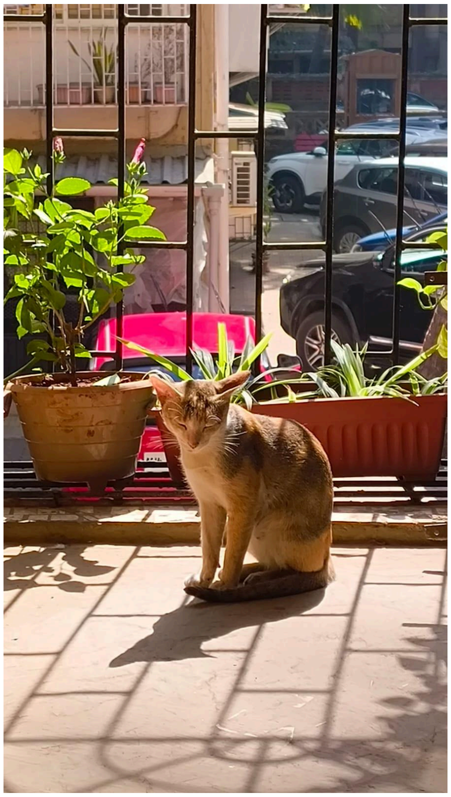


Bit Plane 7 of Image 1

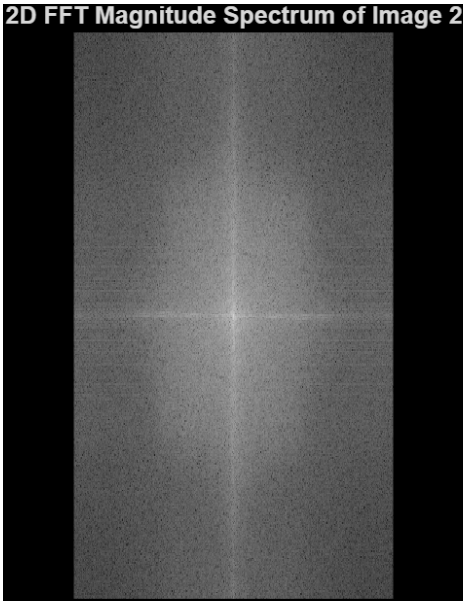


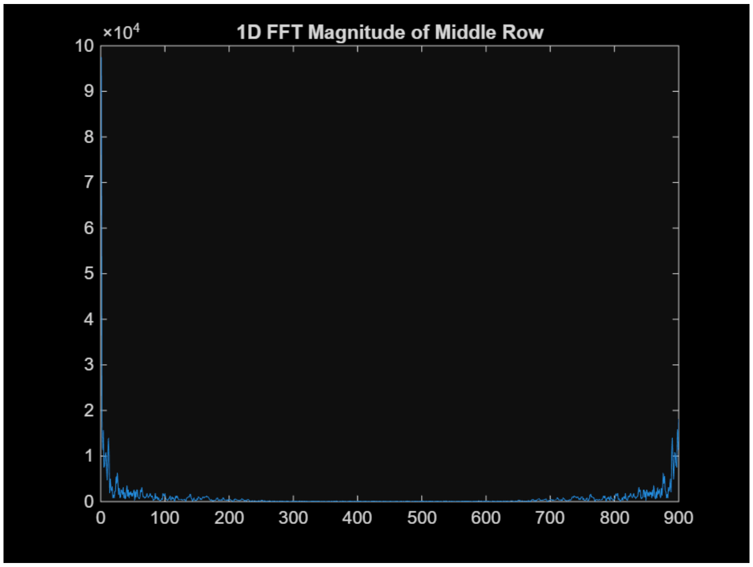
Bit Plane 8 of Image 1

Take a photo featuring yourself along with background elements and perform the Following:



6. Display of FFT(1-D & 2-D) of an image


2D FFT Magnitude Spectrum of Image 2


1D FFT Magnitude of Middle Row

7. Computation of Mean, Standard Deviation, Correlation coefficient of the given Image

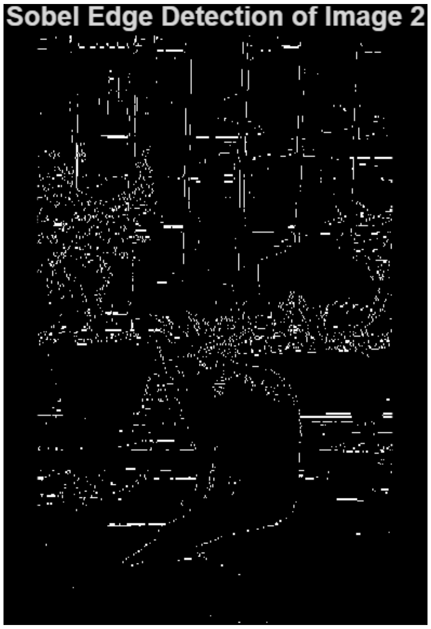
```
Image 1 Mean: 67.51, Std: 44.52, Corr(I1,I2): 0.1222
```

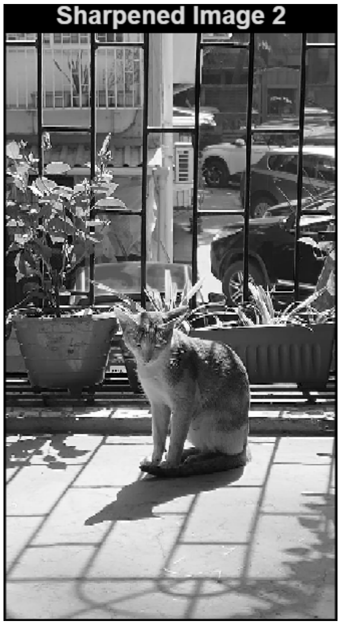
8. Implementation of Image Smoothening Filters (Mean and Median filtering of an Image)

Mean Filtered Image (5x5)

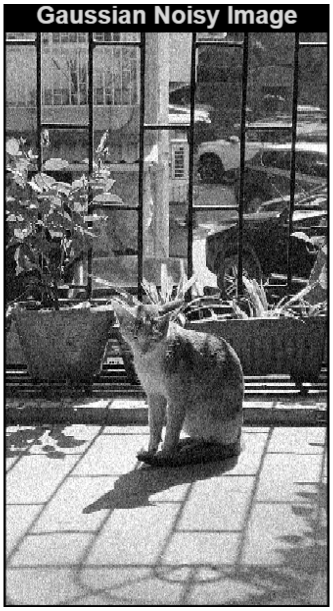
Median Filtered Image (5x5)

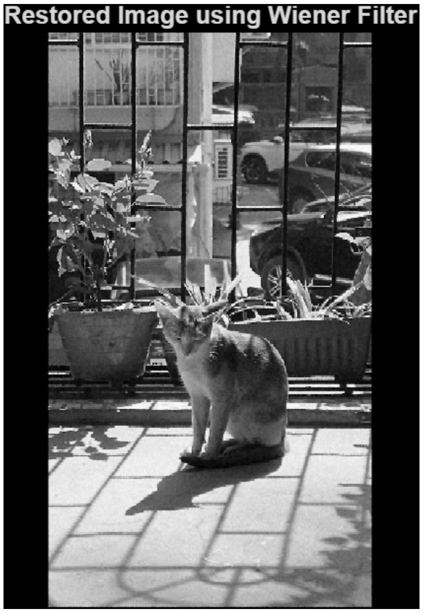
9. Implementation of image sharpening filters and Edge Detection using Gradient Filters



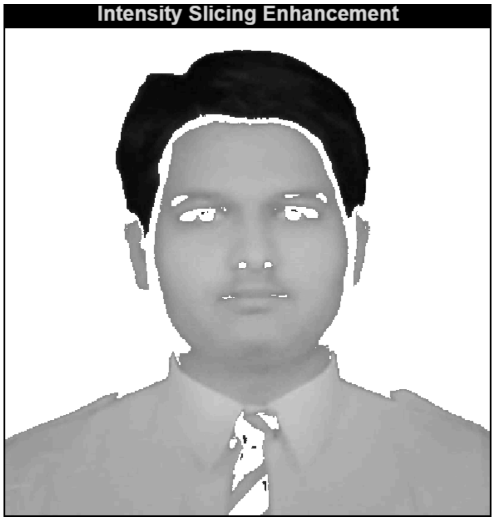


10. Implementation of image restoring techniques





11. Implementation of Image Intensity slicing technique for image enhancement



12. Canny edge detection Algorithm