EAST WEST UNIVERSITY
Department of Computer Science and Engineering

Lab-report:06

Course Name: Digital Image Processing
Course Code: CSE438
Section No: 03

Submitted To:

Prof. Dr. Ahmed Wasif Reza

Department of Computer Science and Engineering

East West University

Submitted by:

Student's ID: 2020-3-60-012

Student's Name: Sadia Islam Prova

Date of submission: 30-4-24

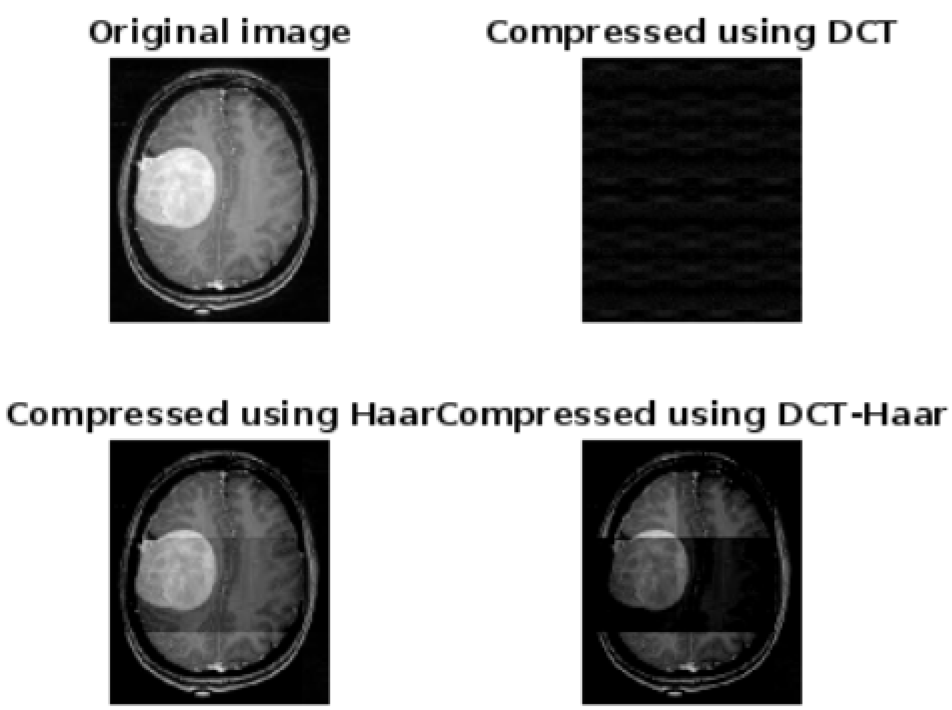
Problem 1: 1. Compress the above images using Discrete Cosine Transform (DCT), Haar Transform, and DCT-Haar, and find out which one is better in terms of compression ratio and PSNR for the given images.

Code:

```
img = imread("gray_image1.png");
gray_img = rgb2gray(img);
blocks_dct = im2col(gray_img, [8 8], 'distinct');
dct_coeffs = dct2(blocks_dct);
Q_dct = 10;
quantized_coeffs_dct = round(dct_coeffs ./ Q_dct);
inv_dct_coeffs = col2im(quantized_coeffs_dct, [8 8], size(gray_img), 
'distinct');
compressed_img_dct = uint8(idct2(inv_dct_coeffs));
[m, n] = size(gray_img);
if mod(m, 2) ~= 0
   gray_img(end+1, :) = 0;
end
if mod(n, 2) ~= 0
   gray_img(:, end+1) = 0;
end
[cA_haar, cH_haar, cV_haar, cD_haar] = haart2(gray_img);
Q_haar = 1.8;
quantized_cA_haar = round(cA_haar ./ Q_haar);
reconstructed_img_haar = ihaart2(quantized_cA_haar, cH_haar, cV_haar, 
cD_haar);
reconstructed_img_haar = uint8(reconstructed_img_haar(1:m, 1:n));
quantized_cA_dct_haar = round(dct2(cA_haar) ./ Q_dct);
reconstructed_img_dct_haar = ihaart2(quantized_cA_dct_haar, cH_haar, cV_haar, 
cD_haar);
reconstructed_img_dct_haar = uint8(reconstructed_img_dct_haar(1:m, 1:n));
figure;
```

```
subplot(2,2,1), imshow(gray_img); title('Original image');

subplot(2,2,2), imshow(compressed_img_dct); title('Compressed using DCT');

subplot(2,2,3), imshow(reconstructed_img_haar); title('Compressed using
Haar');

subplot(2,2,4), imshow(reconstructed_img_dct_haar); title('Compressed using
DCT-Haar');

psnr_dct = psnr(compressed_img_dct, gray_img);

psnr_haar = psnr(reconstructed_img_haar, gray_img);

psnr_dct_haar = psnr(reconstructed_img_dct_haar, gray_img);

disp('PSNR:');

disp(['DCT: ', num2str(psnr_dct), ' dB']);

disp(['Haar: ', num2str(psnr_haar), ' dB']);

disp(['DCT-Haar: ', num2str(psnr_dct_haar), ' dB']);
```

Output:



Problem 2: Apply Gaussian noise to Figure 1, and then use the following to restore the image:

i. Geometric Mean filter

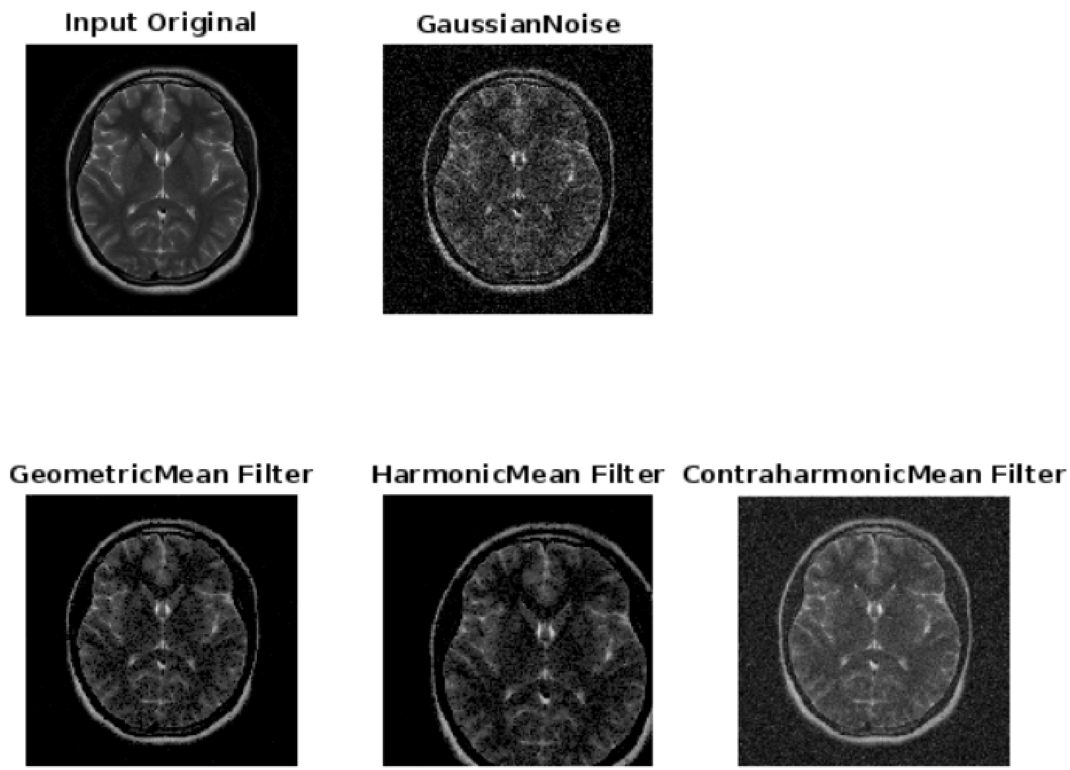
ii. Harmonic Mean filter

iii. Contra-harmonic Mean filter

Code:

```
img = imread("gray img.png");

img = rgb2gray(img);

GI = imnoise(img, 'gaussian');

Gg = im2double(GI);

Kr = 3;

Kc = 3;

Ord = 2;

GM = exp(imfilter(log(Gg), ones(Kr,Kc), 'replicate')).^(1/(Kr*Kc));

HM = (Kr*Kc) ./ imfilter(1./(Gg + eps), ones(Kr,Kc), 'replicate');

CM = imfilter(Gg.^(Ord+1), ones(Kr,Kc), 'replicate') ./ (imfilter(Gg.^(Ord),
ones(Kr,Kc), 'replicate') + eps);

subplot(2, 3, 1), imshow(img), title('Input Original');

subplot(2, 3, 2), imshow(GI), title('GaussianNoise');

subplot(2, 3, 4), imshow(GM), title('GeometricMean Filter');

subplot(2, 3, 5), imshow(HM), title('HarmonicMean Filter');

subplot(2, 3, 6), imshow(CM), title('ContraharmonicMean Filter');

GM_normalized = mat2gray(GM);

HM_normalized = mat2gray(HM);

CM_normalized = mat2gray(CM);

figure;

subplot(1, 3, 1), imshow(GM_normalized), title('Geometric Mean Filter
(Normalized)');

subplot(1, 3, 2), imshow(HM_normalized), title('Harmonic Mean Filter
(Normalized)');

subplot(1, 3, 3), imshow(CM_normalized), title('Contraharmonic Mean Filter
(Normalized)');
```

Output:



Problem 3: Apply Gaussian noise to Figure 1, and then use the following order statistic filters to restore the image:

i. Median filter

ii. Maximum filter

iii. Minimum filter

iv. Midpoint filter

v. Alpha-trimmed filter

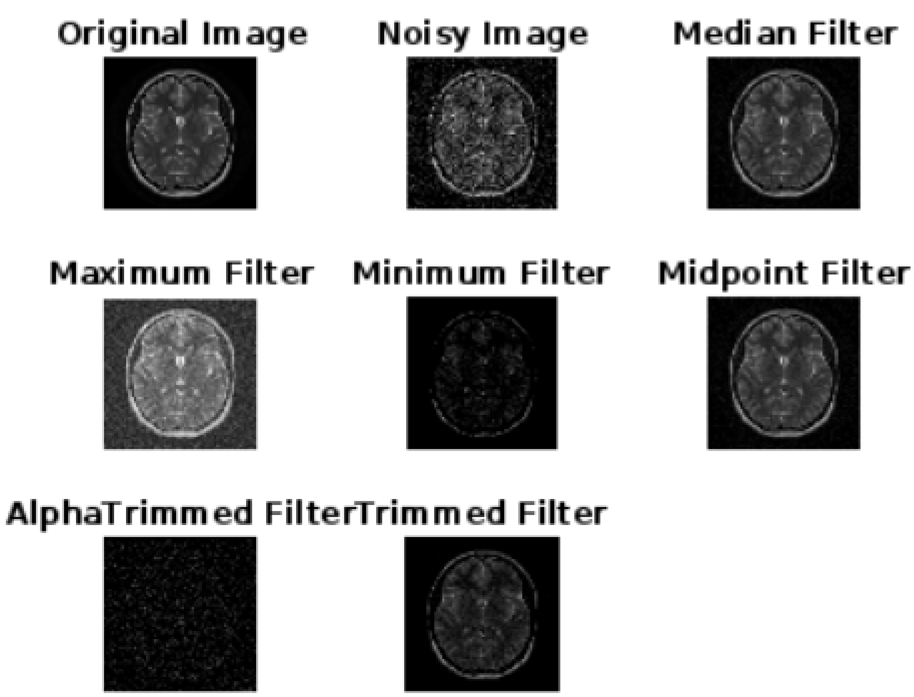
vi. Trimmed filter

Code:

```
img = imread("gray img.png");
img = rgb2gray(img);
img_noise = imnoise(img, 'gaussian', 0, 0.01);
med_filt = medfilt2(img_noise, [5, 5]);
```

```
max_filt = ordfilt2(img_noise, 25, ones(5,5));
min_filt = ordfilt2(img_noise, 1, ones(5,5));
mid_filt = ordfilt2(img_noise, 13, ones(5,5));
alpha_trim_filt = imgaussfilt(img_noise, 2);
alpha_trim_filt = img_noise - alpha_trim_filt;
trimmed_filt = ordfilt2(img_noise, 5, ones(5,5));
subplot(3, 3, 1), imshow(img), title('Original Image');
subplot(3, 3, 2), imshow(img_noise), title('Noisy Image');
subplot(3, 3, 3), imshow(med_filt), title('Median Filter');
subplot(3, 3, 4), imshow(max_filt), title('Maximum Filter');
subplot(3, 3, 5), imshow(min_filt), title('Minimum Filter');
subplot(3, 3, 6), imshow(mid_filt), title('Midpoint Filter');
subplot(3, 3, 7), imshow(alpha_trim_filt), title('AlphaTrimmed Filter');
subplot(3, 3, 8), imshow(trimmed_filt), title('Trimmed Filter');
```

Output:



Problem 4: By observing and comparing each of the outputs, determine which filter restores the image closest to its original state. Mention the reasoning behind your observation.

Solution: Here the median filter restores the image closest to its original state among other filters. Because the median filter replaces each pixel's intensity with the median value of its neighborhood, effectively reducing the impact of outlier noise while preserving the image's overall structure and details.