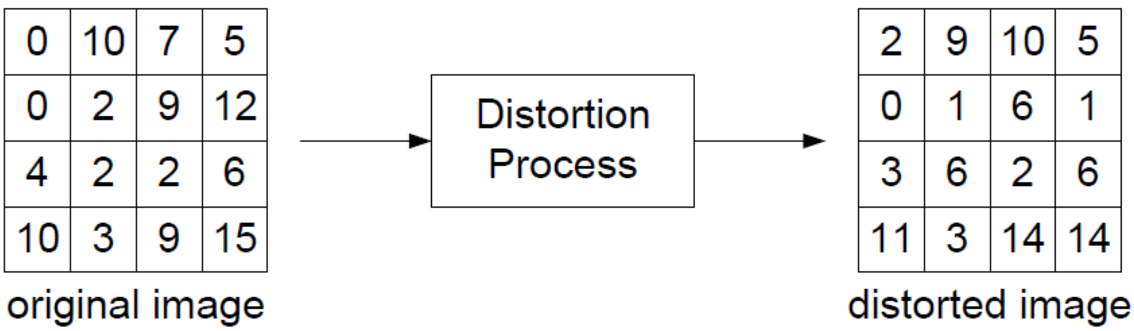
ECE 613, Winter 2022, Homework 1

Parth Nitinbhai Patel, WatID: 20942013

<u>Q1 Solution:</u>



Let suppose, the original image is denoted by x and the distorted image is denoted by y. N represents the total number of pixels. x_i denotes the i^{th} pixel in the original image and similarly, y_i denotes the i^{th} pixel in the distorted image. So, the value of i ranges from 1 – 16, because as we can see that each image represents 16 pixels in total. Hence, N = 16.

Part (a) Mean Absolute Error (MAE):

$$MAE = \sum_{i=1}^{N} \frac{|y_i - x_i|}{N} = \frac{1}{N} \sum_{i=1}^{N} |y_i - x_i|$$

$$= \frac{|2-0| + |9-10| + |10-7| + |5-5| + |0-0| + |1-2| + |6-9| + |1-12| + |3-4| + |6-2| + |2-2| + |6-6| + |11-10| + |3-3| + |14-9| + |14-15|}{16}$$

$$= \frac{2+1+3+0+0+1+3+11+1+4+0+0+1+0+5+1}{16}$$

$$= \frac{33}{16}$$

$$= 2.0625$$

Part (b) Mean Squared Error (MSE):

$$MSE = \sum_{i=1}^{N} \frac{(y_i - x_i)^2}{N} = \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i)^2$$

$$= \frac{\begin{array}{c}(2-0)^2 + (9-10)^2 + (10-7)^2 + (5-5)^2 + (0-0)^2 + (1-2)^2 + \\ (6-9)^2 + (1-12)^2 + (3-4)^2 + (6-2)^2 + (2-2)^2 + (6-6)^2 + \\ (11-10)^2 + (3-3)^2 + (14-9)^2 + (14-15)^2\end{array}}{16}$$

$$= \frac{4 + 1 + 9 + 0 + 0 + 1 + 9 + 121 + 1 + 16 + 0 + 0 + 1 + 0 + 25 + 1}{16}$$

$$= \frac{189}{16}$$

$$= 11.8125$$

Part (c) Peak Signal-to-Noise Ratio (PSNR:

Here, we are taking 4 bits/pixel. So, the MAX_I is the maximum possible intensity which is given as $2^B - 1$, where B is the number of bits. So for our given example $MAX_I = (2^4 - 1) = 15$.

$$PSNR = 10\log\left(\frac{MAX_I^2}{MSE}\right)$$

$$PSNR = 10\log\left(\frac{225}{11.8125}\right)$$

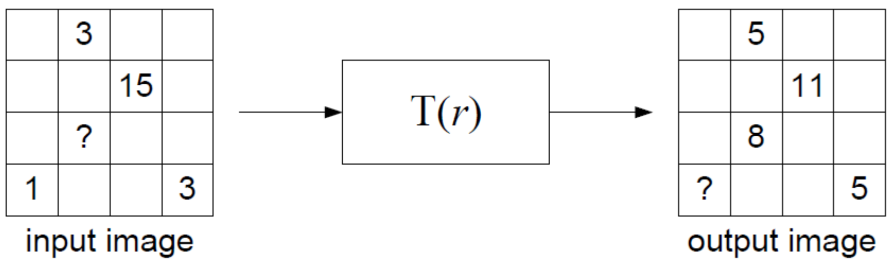
$$PSNR = 12.7984\ dB$$

Q2 Solution:

A 4×4, 4bits/pixel image passes through a point-wise intensity transformation given by

$$s = T(r) = \alpha \log_2(1+r) + \beta$$

where α and β are unknown parameters. Only a few pixels are available in the input and the output images, as shown below.



From the above given values in the input and output images we can form 2 equations to solve the value of α and β. These two equations are shown below:

$$5 = \alpha \log(1 + 3) + \beta$$
$$\mathbf{5 = 2\alpha + \beta}$$

$$11 = \alpha \log(1 + 15) + \beta$$
$$\mathbf{11 = 4\alpha + \beta}$$

Part (a):
Solving the above two equations we get the values of α and β as:

$$\mathbf{\alpha = 3, \beta = -1.}$$

Part (b):

Value of pixel in third row:

Let the value be r. Then,

$8 = \alpha \log_2(1+r) + \beta$
$8 = 3 \log_2(1+r) - 1$ (Substituting the values of α and β)
$3 = \log_2(1+r)$
$\mathbf{r = 7}$

Part (c):
Value of pixel in fourth row:

Let the value of pixel be s. Then,

$s = \alpha \log_2(1+1) + \beta$
$s = 3 \log_2(1+1) - 1$ (Substituting the values of α and β)
$\mathbf{s = 2}$

Q3 Solution:

We know that the intensity value of black pixel corresponds to 0 and the intensity value of white pixel corresponds to 1.

A] $s = r^2$

This above **Gamma transformation** will make the output image darker since the intensity values range between 0 - 1 and square of such a number will be less than itself. Therefore, the resulting image from this transformation should be image **E**.

B] $s = 1 - r$

This above transformation is similar if we invert an image. It follows **inverting pixel intensities** and hence, it can be seen in the output image **C**.

C] $s = 0.5r + 0.25$

From the above **Reducing Contrast** transformation equation, it can be seen that it halving the original pixel intensity and adds a small marginal value back, which result in the new intensity that is lower than the original when original value is greater than 0.5 and greater than the original when the original value is less than 0.5. And hence, the output image shows lightness characteristic in darker regions and visa versa. The same can be seen in image **D**.

D]

$$s = \begin{cases} 0 & r \leq 0.7 \\ 1 & else \end{cases}$$

The above **Thresholding** transformation will convert the input image into a binary image as it is assigning either 0 or 1 depending upon the threshold values. Hence, this can be seen in option **B**.

E]

$$s = \begin{cases} 0 & r < 0.25 \\ 2r - 0.5 & 0.25 \geq r \leq 0.75 \\ 1 & 0.75 \leq r \end{cases}$$

Here, with the help of the above transformation the lower pixel values are made darker (intense) and the higher pixel values are made lighter. The middle range is spreading out using a linear transformation as can be seen from the equation, which makes the image even brighter. This can be seen in **A**.

<u>Q4 Solution:</u>

6	13	12	13
12	6	7	12
13	7	7	12
14	11	11	14

Part (a):

The Full-scale contrast stretch is done with the help of following equation:

$$s = round\left((2^B - 1) \times \frac{r - r_{min}}{r_{max} - r_{min}} \right)$$

Here, r_{max} = maximum value from the above input image = 14, and
r_{min} = minimum value from the above input image = 6. It is also given that the input image is 4 bits/pixel, so $(2^B - 1) = 15$.
Hence,

$$s = round(1.875 \times (r - 6))$$

Substituting all the pixel values from the above given input image we have:

$$s(6) = round(1.875 \times (6 - 6)) = 0$$

$$s(7) = round(1.875 \times (7 - 6)) = 2$$

$$s(11) = round(1.875 \times (11 - 6)) = 9$$

$$s(12) = round(1.875 \times (12 - 6)) = 11$$

$$s(13) = round(1.875 \times (13 - 6)) = 13$$

$$s(14) = round(1.875 \times (14 - 6)) = 15$$

Using the above calculated values and putting those values we get the following image:

0	13	11	13
11	0	2	11
13	2	2	11
15	9	9	15

Part (b):

In order to perform histogram equalization, we first need to compute cumulative histogram Q(k) from the histogram H(k) and finally apply the full-scale contrast stretch to obtain the desired result.

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	0	0	0	0	0	0	2	3	0	0	0	2	4	3	2	0
Q(k)	0	0	0	0	0	0	2	5	5	5	5	7	11	14	16	16

Hence, the intermediate image will be like this:

2	14	11	14
11	2	5	11
14	5	5	11
16	7	7	16

Now, we will apply the full-scale contrast stretch to this image. So, from the above image we get $r_{max} = 16$ and $r_{min} = 2$. It is also given that the input image is 4 bits/pixel, so $(2^B - 1) = 15$.

Therefore,

$$s = round(\ 1.071 \times (\ r - 2)\ )$$

Substituting all the pixel values from the above given input image we have:

$$s(\ 2) = round(\ 1.071 \times (\ 2 - 2)\ ) = 0$$

$$s(\ 5) = round(\ 1.071 \times (\ 5 - 2)\ ) = 3$$

$$s(\ 7) = round(\ 1.071 \times (\ 7 - 2)\ ) = 5$$

$$s(\ 11) = round(\ 1.071 \times (\ 11 - 2)\ ) = 10$$

$$s(\ 14) = round(\ 1.071 \times (\ 14 - 2)\ ) = 13$$

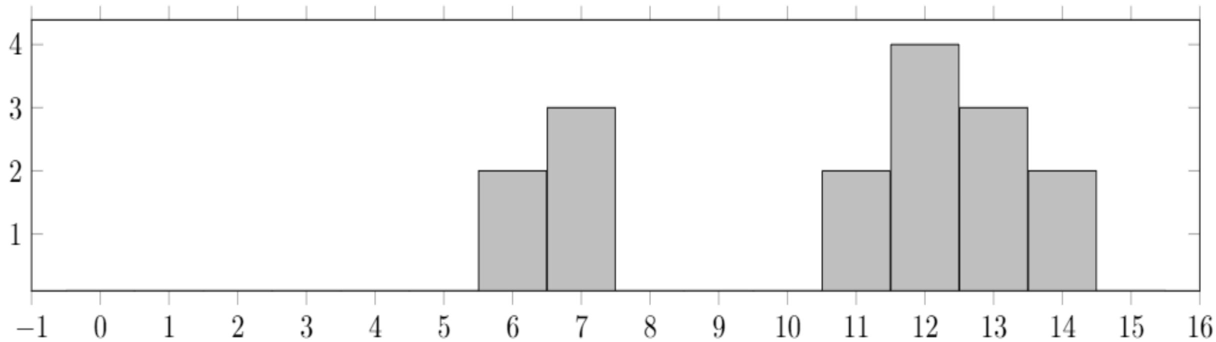
$$s(\ 16) = round(\ 1.071 \times (\ 16 - 2)\ ) = 15$$

Using the above calculated values and putting those values we get the following image:

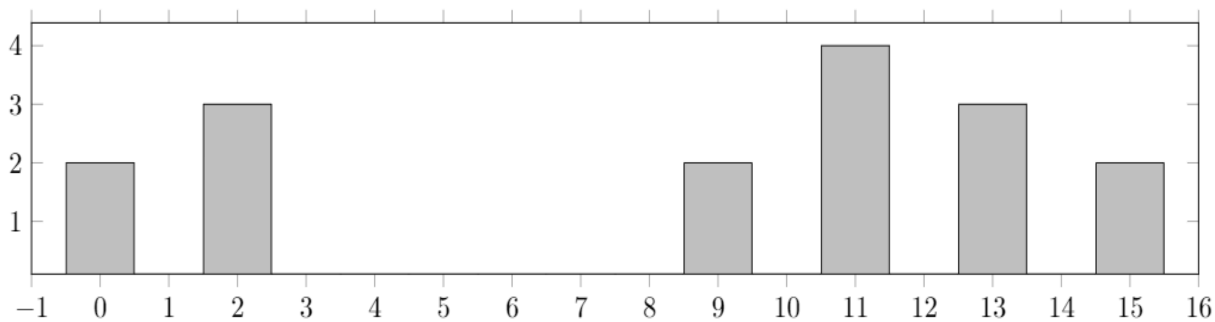
0	13	10	13
10	0	3	10
13	3	3	10
15	5	5	15

<u>Part (c):</u>

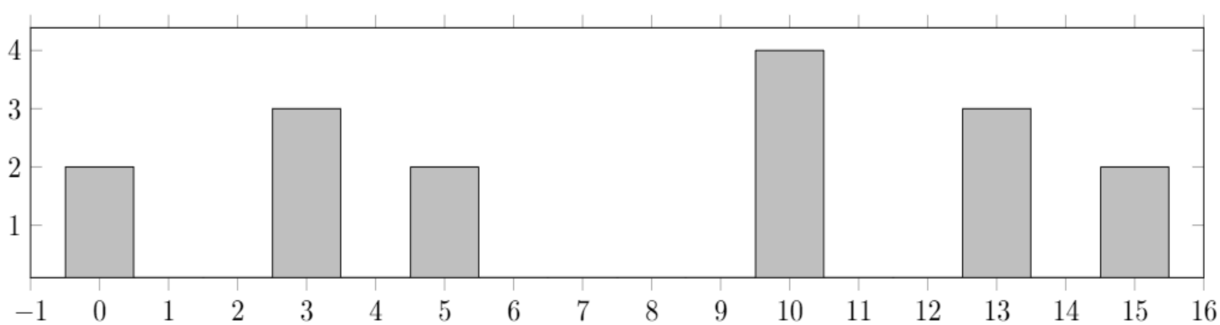
Histogram of the original image:



Histogram of Fill-Scale Contrast Stretched image:



Histogram of histogram equalized image:



Q5 Solution:

2	9	10	0
7	1	6	1
10	15	2	6
11	3	8	10

Original Image

Zero padding is done in order to get the same dimensions of the output image as that of the input image. Hence, the zero padded image is given below:

$$I = \begin{array}{|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 2 & 9 & 10 & 0 & 0 \\ \hline 0 & 7 & 1 & 6 & 1 & 0 \\ \hline 0 & 10 & 15 & 2 & 6 & 0 \\ \hline 0 & 11 & 3 & 8 & 10 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

Part (a):

Shift-Invariant Linear Filter is given as:

Filter 1:

$$F_1 = \frac{1}{4} \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

To calculate the output, we will slide the filter F_1 on the zero padded image I with a stride of 1. Here, we will also take the inner product. Thus, the pixel values of the output image will be the sum of all the elements of the inner product.

$O(m,n) = \sum < I,\ F_1 >$

Hence, the filtered output image 1 is,

O_1 =

4	3.25	3.75	2.75
3.25	9.25	3.5	3
8.25	4	8.75	3.25
3.25	8.5	3.75	3.5

Part (b):

Shift-Invariant Linear Filter is given as:

Filter 2:

$F_2 = \frac{1}{4}$

-1	0	-1
0	4	0
-1	0	-1

To calculate the output, we will slide the filter F_1 on the zero padded image I with a stride of 1. Here, we will also take the inner product. Thus, the pixel values of the output image will be the sum of all the elements of the inner product.

Hence, the filtered output image 2 is,

$O_2 =$

1.75	5.75	9.5	-1.5
1	-5	-1.5	-2
9	7	-1.75	2.5
7.25	0	2.75	9.5

Part (c):

Here, the filter 3 is the combination of filter 1 and filter 2. So, $F_3 = F_1 + F_2$. So, we can get the penultimate step image by addition of images we get from Part (a) and Part (b) without the division operations. We can directly add those images and perform division operation on the resulting image to get the final image as the result.

Hence,

$$O_3 = (F_1 + F_2) \otimes I$$

$$O_3 = (F_1 \otimes I) + (F_2 \otimes I)$$

$$O_3 = O_1 + O_2$$

$O_3 =$

5.75	9	13.25	1.25
4.25	4.25	2	1
17.25	11	7	5.75
10.5	8.5	6.5	13

<u>Q6 Solution:</u>

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \qquad F_4^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

Part (a): Compute the 2D-DFT of the 4 x 4 gray-scale image given below:

$$X = \begin{bmatrix} 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \end{bmatrix}$$

We have given our original input image X.

Computing 2D-DFT of the original image $\tilde{X} = F_4 X F_4$ for a 4 x 4 image.

$$\tilde{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\tilde{X} = \begin{bmatrix} 24 & 8 & 24 & 8 \\ 0 & 0 & 0 & 0 \\ 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\tilde{X} = \begin{bmatrix} 64 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Part (b): The inverse 2D-DFT of the given three 4 x 4 images is computed by

$X = \frac{1}{N^2} F_4^* \tilde{X} F_4^*$. Here, N = 4.

For Pattern 1,

$$X_1 = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_1 = \frac{1}{16} \begin{bmatrix} 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

For Pattern 2,

$$X_2 = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 0 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_2 = \frac{1}{16}\begin{bmatrix} 0 & 0 & 32 & 0 \\ 0 & 0 & 32 & 0 \\ 0 & 0 & 32 & 0 \\ 0 & 0 & 32 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \end{bmatrix}$$

For Pattern 3,

$$X_3 = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_3 = \frac{1}{16}\begin{bmatrix} 0 & 0 & -16 & 0 \\ 0 & 0 & -16 & 0 \\ 0 & 0 & -16 & 0 \\ 0 & 0 & -16 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

<u>Part (c):</u>

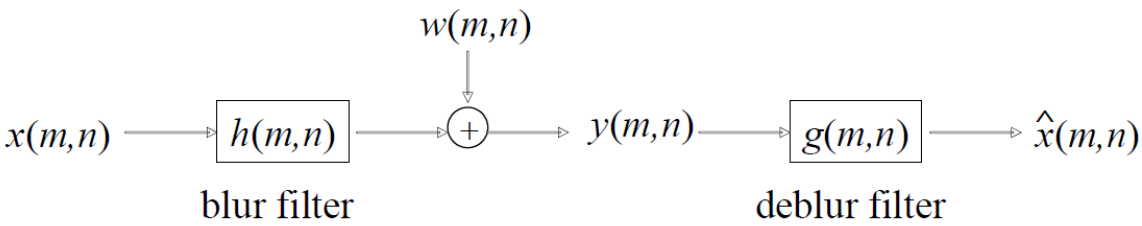
Now, we will add the results which we obtained in part (b).
That is, $X = X_1 + X_2 + X_3$.
So,

$$X = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix} + \begin{bmatrix} 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \\ 2 & -2 & 2 & -2 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \end{bmatrix}$$

Observation: Here, we can clearly see that adding all the three images gives the original input image. That is the distribution law of inverse 2D-DFT which implies that computing them independently and adding the consecutive results produces the initial input image. 2D-DFT is inversible and also has the preserving property.

Q7 Solution:



We have given that:

$$\sigma_X^2 = 100, \ \sigma_W^2 = 25$$

$$H(u,v) = \begin{bmatrix} 1 & -0.2 - 0.2j & 0 & -0.2 + 0.2j \\ -0.2 - 0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2 + 0.2j & 0.05 & 0 & -0.05j \end{bmatrix}$$

Part (a): Design deblur filter *G(u,v)* using inverse filtering approach.

$$G(u,v) = \frac{1}{H(u,v)}$$

$$G(u,v) = \frac{1}{\begin{bmatrix} 1 & -0.2-0.2j & 0 & -0.2+0.2j \\ -0.2-0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2+0.2j & 0.05 & 0 & -0.05j \end{bmatrix}}$$

$$G(u,v) = \begin{bmatrix} 1 & -2.5+2.5j & inf & -2.5-2.5j \\ -2.5+2.5j & -20j & inf & 20 \\ inf & inf & inf & inf \\ -2.5-2.5j & 20 & inf & 20j \end{bmatrix}$$

Part (b): Design deblur filter *G(u,v)* using pseudo inverse filtering approach with $\delta = 0.03$.

$$\left|H(u,v)\right| = \begin{bmatrix} 1 & 0.2828 & 0 & 0.2828 \\ 0.2828 & 0.05 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0.2828 & 0.05 & 0 & 0.05 \end{bmatrix}$$

$$G(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & \left|H(u,v)\right| > 0.03 \\ 0 & \left|H(u,v)\right| \le 0.03 \end{cases} = \begin{bmatrix} 1 & -2.5+2.5j & 0 & -2.5-2.5j \\ -2.5+2.5j & -20j & 0 & 20 \\ 0 & 0 & 0 & 0 \\ -2.5-2.5j & 20 & 0 & 20j \end{bmatrix}$$

Part (c): Design deblur filter *G(u,v)* using pseudo inverse filtering approach with $\delta = 0.1$. Using |*H(u,v)*| from part (b).

$$G(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & |H(u,v)| > 0.1 \\ 0 & |H(u,v)| \leq 0.1 \end{cases}$$

$$G(u,v) = \begin{bmatrix} 1 & -2.5 + 2.5j & 0 & -2.5 - 2.5j \\ -2.5 + 2.5j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2.5 - 2.5j & 0 & 0 & 0 \end{bmatrix}$$

Part (d): Design deblur filter *G(u,v)* using pseudo inverse filtering approach with $\delta = 0.3$. Using |*H(u,v*| from part (b).

$$G(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & |H(u,v)| > 0.3 \\ 0 & |H(u,v)| \leq 0.3 \end{cases} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Part (e): Design deblur filter *G(u,v)* using Wiener Filtering approach.

We already know that $\sigma_X^2 = 100,\ \sigma_W^2 = 100$. Hence, $K = \dfrac{\sigma_W^2}{\sigma_X^2} = 0.25$

$$G(u,v) = \frac{H^*(u,v)}{\left|H(u,v)\right|^2 + K}$$

$$G(u,v) = \frac{\begin{bmatrix} 1 & -0.2+0.2j & 0 & -0.2-0.2j \\ -0.2+0.2j & -0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2-0.2j & 0.05 & 0 & 0.05j \end{bmatrix}}{\left(\begin{bmatrix} 1 & 0.2828 & 0 & 0.2828 \\ 0.2828 & 0.05 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0.2828 & 0.05 & 0 & 0.05 \end{bmatrix}\right)^2 + 0.25}$$

$$G(u,v) = \frac{\begin{bmatrix} 1 & -0.2+0.2j & 0 & -0.2-0.2j \\ -0.2+0.2j & -0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2-0.2j & 0.05 & 0 & 0.05j \end{bmatrix}}{\begin{bmatrix} 1.25 & 0.33 & 0.25 & 0.33 \\ 0.33 & 0.2525 & 0.25 & 0.2525 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.33 & 0.2525 & 0.25 & 0.2525 \end{bmatrix}}$$

$$G(u,v) = \begin{bmatrix} 0.8 & -0.61+0.61j & 0 & -0.61-0.61j \\ -0.61+0.61j & -0.2j & 0 & 0.2 \\ 0 & 0 & 0 & 0 \\ -0.61-0.61j & 0.2 & 0 & 0.2j \end{bmatrix}$$

Part (f): To compute *h(m,n)*, we find inverse 2D-DFT of *H(u,v)*.

$$h(m,n) = \frac{1}{N^2} F_4^* \; H(u,v) \; F_4^*$$

$$h(m,n) = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 1 & -0.2-0.2j & 0 & -0.2+0.2j \\ -0.2-0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2+0.2j & 0.05 & 0 & -0.05j \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$h(m,n) = \begin{bmatrix} 0.0188 & 0.0563 & 0.0563 & 0.0188 \\ 0.0563 & 0.1188 & 0.1188 & 0.0562 \\ 0.0562 & 0.1188 & 0.1188 & 0.0562 \\ 0.0187 & 0.0562 & 0.0562 & 0.0187 \end{bmatrix}$$

Part (g): To compute *g(m,n)*, we find inverse 2D-DFT of *G(u,v)* from part (c).

$$g(m,n) = \frac{1}{N^2} F_4^* G(u,v) \; F_4^*$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 1 & -2.5+2.5j & 0 & -2.5-2.5j \\ -2.5+2.5j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2.5-2.5j & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$g(m,n) = \begin{bmatrix} -0.5625 & -0.5625 & 0.0625 & 0.0625 \\ -0.5625 & -0.5625 & 0.0625 & 0.0625 \\ 0.0625 & 0.0625 & 0.6875 & 0.6875 \\ 0.0625 & 0.0625 & 0.6875 & 0.6875 \end{bmatrix}$$

<u>Q8 Solution:</u>

We have given that, $\sigma_X^2 = 400,\ \sigma_W^2 = 100$ and

$$h(m,n) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Part (a):
H(u,v) , the blur filter in the frequency (2D-DFT) domain.

$$H(u,v) = F_4\, h(m,n)\ F_4$$

$$H(u,v) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$H(u,v) = \begin{bmatrix} 20 & -2-2j & 0 & -2+2j \\ -2-2j & 2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2+2j & 2 & 0 & -2j \end{bmatrix}$$

Part (b): Design a deblur filter *G(u,v)* using the pseudo-inverse filtering approach with $\delta = 1$.

$$|H(u,v)| = \begin{bmatrix} 20 & 2.8284 & 0 & 2.8284 \\ 2.8284 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 2.8284 & 2 & 0 & 2 \end{bmatrix}$$

$$G(u,v) = \frac{1}{H(u,v)}$$

$$G(u,v) = \frac{1}{\begin{bmatrix} 20 & -2-2j & 0 & -2+2j \\ -2-2j & 2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2+2j & 2 & 0 & -2j \end{bmatrix}}$$

$$G(u,v) = \begin{bmatrix} 0.05 & -0.25+0.25j & inf & -0.25-0.25j \\ -0.25+0.25j & -0.5j & inf & 0.5 \\ inf & inf & inf & inf \\ -0.25-0.25j & 0.5 & inf & 0.5j \end{bmatrix}$$

$$G(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & \left|H(u,v)\right| > 1 \\ 0 & \left|H(u,v)\right| \leq 1 \end{cases}$$

$$G(u,v) = \begin{bmatrix} 0.05 & -0.25+0.25j & 0 & -0.25-0.25j \\ -0.25+0.25j & -0.5j & 0 & 0.5 \\ 0 & 0 & 0 & 0 \\ -0.25-0.25j & 0.5 & 0 & 0.5j \end{bmatrix}$$

Part (c): Design a deblur filter *G(u,v)* using the pseudo-inverse filtering approach with $\delta = 3$.

$$G(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & |H(u,v)| > 3 \\ 0 & |H(u,v)| \leq 3 \end{cases}$$

$$G(u,v) = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Part (d): Design deblur filter *G(u,v)* using Wiener Filtering approach.

We already know that $\sigma_X^2 = 100,\ \sigma_W^2 = 100$. Hence, $K = \dfrac{\sigma_W^2}{\sigma_X^2} = 0.25$

$$G(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + K}$$

$$G(u,v) = \frac{\begin{bmatrix} 20 & -2+2j & 0 & -2-2j \\ -2+2j & -2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2-2j & 2 & 0 & 2j \end{bmatrix}}{\left(\begin{bmatrix} 20 & 2.8284 & 0 & 2.8284 \\ 2.8284 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 2.8284 & 2 & 0 & 2 \end{bmatrix}\right)^2 + 0.25}$$

$$G(u,v) = \frac{\begin{bmatrix} 20 & -2+2j & 0 & -2-2j \\ -2+2j & -2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2-2j & 2 & 0 & 2j \end{bmatrix}}{\begin{bmatrix} 400.25 & 8.25 & 0.25 & 8.25 \\ 8.25 & 4.25 & 0.25 & 4.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 8.25 & 4.25 & 0.25 & 4.25 \end{bmatrix}}$$

$$G(u,v) = \begin{bmatrix} 0.05 & -0.24+0.24j & 0 & -0.24-0.24j \\ -0.24+0.24j & -0.47j & 0 & 0.47 \\ 0 & 0 & 0 & 0 \\ -0.24-0.24j & 0.47 & 0 & 0.47j \end{bmatrix}$$

Q9 Solution:

We have given that, $I = \begin{bmatrix} 7 & 2 & 7 & 2 \\ 2 & 7 & 2 & 7 \\ 7 & 2 & 7 & 2 \\ 2 & 7 & 2 & 7 \end{bmatrix}$

Part (a): Zero padding basically means that we will add 0's to the boundary of the image so that during sliding the filter on the image no pixel in the image remains unencountered. By using zero padding we can easily be able to retain the original size of the input image after the filter operation.

$$I_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 2 & 7 & 2 & 0 \\ 0 & 2 & 7 & 2 & 7 & 0 \\ 0 & 7 & 2 & 7 & 2 & 0 \\ 0 & 2 & 7 & 2 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Linear\ Shift\ Invariant\ Filter\ (F_1) = \frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

In order to calculate the filtered image we will slide the filter F_1 on the zero padded image with stride 1 and take the inner product. The output pixel values are the sum of all the elements of the inner product.

$O(m,n) = \sum < I_1, F_1 >$

Hence, the filtered output image is,

$$O_1 = \begin{bmatrix} 3 & 4 & 3 & 3 \\ 4 & 4 & 5 & 3 \\ 3 & 5 & 4 & 4 \\ 3 & 3 & 4 & 3 \end{bmatrix}$$

Let the input image I_1 be represented by p and the output filtered image O_1 be represented by q.

Mean Absolute Error (MAE):

$$MAE = \sum_{i=1}^{N} \frac{|q_i - p_i|}{N}$$

$$MAE = \frac{4 + 2 + 4 + 1 + 2 + 3 + 3 + 4 + 4 + 3 + 3 + 2 + 2 + 1 + 4 + 2 + 4}{16}$$

$$MAE = \frac{46}{16}$$

$$MAE = 2.875$$

Mean Squared Error (MSE):

$$MSE = \sum_{i=1}^{N} \frac{(q_i - p_i)^2}{N}$$

$$MSE = \frac{16 + 4 + 16 + 1 + 4 + 9 + 9 + 16 + 16 + 9 + 9 + 4 + 1 + 16 + 4 + 16}{16}$$

$$MSE = \frac{150}{16}$$

$$MSE = 9.375$$

Peak Signal-to-noise Ratio (PSNR):

Here, we are taking 4 bits/pixel. So, the MAX_I is the maximum possible intensity which is given as 2B – 1, where B is the number of bits. So for our given example MAX_I = (24 – 1) = 15.

$$PSNR = 10\log\left(\frac{MAX_I^2}{MSE}\right)$$

$$PSNR = 10\log\left(\frac{225}{9.375}\right)$$

$$PSNR = 13.8021\ dB$$

Part (b): In this case we will pass the original input image through a median filter with replicate padding,

$$I_2 = \begin{bmatrix} 7 & 7 & 2 & 7 & 7 & 2 \\ 7 & 7 & 2 & 7 & 2 & 2 \\ 2 & 2 & 7 & 2 & 7 & 7 \\ 7 & 7 & 2 & 7 & 2 & 2 \\ 2 & 2 & 7 & 2 & 7 & 7 \\ 2 & 2 & 7 & 2 & 7 & 7 \end{bmatrix}$$

To calculate the filtered image using a 3 x 3 median filter with the replicate padding input image we take a 3 x 3section of the input image and pass the median value from all the intensities in the 3 x 3 section. For example,

$$O_2(0,0) = median\left(\begin{bmatrix} 7 & 7 & 2 \\ 7 & 7 & 2 \\ 2 & 2 & 7 \end{bmatrix}\right)$$

$$O_2(0,0) = median(2,2,2,2,7,7,7,7,7) = 7$$

Hence, the filtered output image is given as:

$$O_2 = \begin{bmatrix} 7 & 7 & 2 & 2 \\ 7 & 7 & 2 & 2 \\ 2 & 2 & 7 & 7 \\ 2 & 2 & 7 & 7 \end{bmatrix}$$

Let the input image I_2 be represented by p and the output filtered image O_2 be represented by q.

Mean Absolute Error (MAE):

$$MAE = \sum_{i=1}^{N} \frac{\left| q_i - p_i \right|}{N}$$

$$MAE = \frac{5 + 5 + 5 + 5 + 5 + 5 + 5 + 5}{16}$$

$$MAE = \frac{40}{16}$$

$$MAE = 2.5$$

Mean Squared Error (MSE):

$$MSE = \sum_{i=1}^{N} \frac{\left(q_i - p_i\right)^2}{N}$$

$$MSE = \frac{25 + 25 + 25 + 25 + 25 + 25 + 25 + 25}{16}$$

$$MSE = \frac{200}{16}$$

$$MSE = 12.5$$

Peak Signal-to-noise Ratio (PSNR):

Here, we are taking 4 bits/pixel. So, the MAXI is the maximum possible intensity which is given as 2B – 1, where B is the number of bits. So for our given example MAXI = (24 – 1) = 15.

$$PSNR = 10\log\left(\frac{MAX_I^2}{MSE}\right)$$

$$PSNR = 10\log\left(\frac{225}{12.5}\right)$$

$$PSNR = 12.5527\ dB$$

Q10 Solution:

We have given that,

$$I = \begin{bmatrix} 9 & 8 & 7 & 6 \\ 8 & 7 & 13 & 5 \\ 7 & 6 & 5 & 4 \\ 6 & 1 & 4 & 3 \end{bmatrix}$$

Part (a): First, we will apply replicate padding on the input image given above.

$$I = \begin{bmatrix} 9 & 9 & 8 & 7 & 6 & 6 \\ 9 & 9 & 8 & 7 & 6 & 6 \\ 8 & 8 & 7 & 13 & 5 & 5 \\ 7 & 7 & 6 & 5 & 4 & 4 \\ 6 & 6 & 1 & 4 & 3 & 3 \\ 6 & 6 & 1 & 4 & 3 & 3 \end{bmatrix}$$

Now, we will apply median filter which is given in the problem on the above replicate padded image we will get the following image,

$$O(0,0) = median\left(\begin{bmatrix} & 9 & \\ 9 & 9 & 8 \\ & 8 & \end{bmatrix}\right)$$

$$O(0,0) = median(8,8,9,9,9) = 9$$

Apply the filter by shifting the filter one stride over the whole input image. So, the output median filtered image is given as:

$$O = \begin{bmatrix} 9 & 8 & 7 & 6 \\ 8 & 8 & 7 & 5 \\ 7 & 6 & 5 & 4 \\ 6 & 4 & 4 & 3 \end{bmatrix}$$

Part (b): Here, we will use the minimum filter of the same pattern in part (a) on the replicate padded image *I*. We will get the following resulting image:

$$O(0,0) = minimum\left(\begin{bmatrix} & 9 & \\ 9 & 9 & 8 \\ & 8 & \end{bmatrix}\right)$$

$$O(0,0) = minimum(8,8,9,9,9) = 8$$

Apply the filter by shifting the filter one stride over the whole input image. So, the output minimum filtered image is given as:

$$O = \begin{bmatrix} 8 & 7 & 6 & 5 \\ 7 & 6 & 5 & 4 \\ 6 & 1 & 4 & 3 \\ 1 & 1 & 1 & 3 \end{bmatrix}$$

Part (c): Here, we will use the maximum filter of the same pattern in part (a) on the replicate padded image *I*. We will get the following resulting image:

$$O(0,0) = maximum\left(\begin{bmatrix} & 9 & \\ 9 & 9 & 8 \\ & 8 & \end{bmatrix}\right)$$

$$O(0,0) = maximum(8,8,9,9,9) = 9$$

Apply the filter by shifting the filter one stride over the whole input image. So, the output maximum filtered image is given as:

$$O = \begin{bmatrix} 9 & 9 & 13 & 7 \\ 9 & 13 & 13 & 13 \\ 8 & 7 & 13 & 5 \\ 7 & 6 & 5 & 4 \end{bmatrix}$$

Part (d): Here, there is a slight change. We will filter the replicate padded image *I* with the order statistics filter $w_i = \left\{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\right\}$. After applying this filter, we will get the following result:

$$O(0,0) = round\left(sum\left(\begin{bmatrix} & 9 & \\ 9 & 9 & 8 \\ & 8 & \end{bmatrix} * \left\{0,\ \frac{1}{3},\ \frac{1}{3},\ \frac{1}{3},\ 0\right\}\right)\right)$$

$$O(0,0) = round\left(sum\left([8,8,9,9,9] * \left[0,\frac{1}{3},\frac{1}{3},\frac{1}{3},0\right]\right)\right)$$

$$O(0,0) = 9$$

Now, we will apply the filter by shifting the filter one stride over the whole input image and we will get the below result:

$$O = \begin{bmatrix} 9 & 8 & 7 & 6 \\ 8 & 8 & 6 & 5 \\ 7 & 6 & 5 & 4 \\ 6 & 4 & 4 & 3 \end{bmatrix}$$

Q11 Solution:

Input Image(I) is Given as:

3	3	1	3	3	3	4
0	3	3	3	3	3	3
3	3	3	2	3	3	12
12	3	3	3	3	12	12
10	12	2	3	3	12	12
12	14	12	12	12	12	11
11	12	12	12	10	12	12

Prewitt gradient operator = $|g(m,n)| = |g_1(m,n)| + |g_2(m,n)|$, we have given the value of threshold, T = 22.
For Prewitt operator,

we

$$g_1(m,n) = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad g_2(m,n) = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Now, we will apply these filters to the input image(I) and we get the following result:

$$O_1(m,n) = \begin{bmatrix} 1 & -1 & 2 & 1 & 10 \\ -6 & -7 & 0 & 10 & 18 \\ -17 & -10 & 1 & 19 & 27 \\ -17 & -11 & 1 & 18 & 17 \\ -7 & -11 & -1 & 9 & 10 \end{bmatrix} \qquad O_2(m,n) = \begin{bmatrix} 2 & 1 & 1 & -1 & 8 \\ 12 & 0 & 0 & 9 & 18 \\ 15 & 9 & 0 & 10 & 9 \\ 20 & 29 & 27 & 18 & 8 \\ 11 & 19 & 26 & 16 & 7 \end{bmatrix}$$

We will approximate the gradient magnitude by adding the modulus of $O_1(m,n)$ and $O_2(m,n)$ as shown below:

$$O(m,n) = \begin{bmatrix} 3 & 2 & 3 & 2 & 18 \\ 18 & 1 & 0 & 19 & 36 \\ 32 & 19 & 1 & 29 & 36 \\ 37 & 40 & 28 & 36 & 25 \\ 18 & 30 & 27 & 25 & 17 \end{bmatrix}$$

Applying the threshold, T = 22 on the output image, we get

$$e(m,n) = \begin{cases} 1 & O(m,n) > 22 \\ 0 & , otherwise \end{cases}$$

Hence, the edge map is given as:

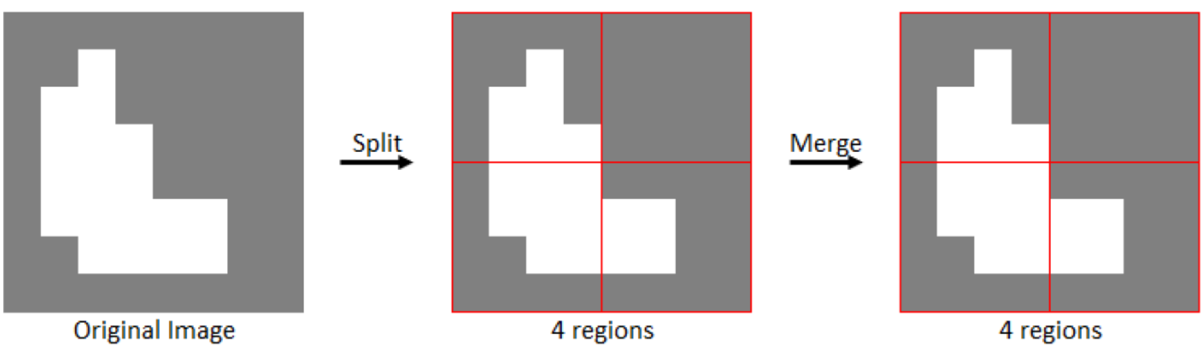
$$e(m,n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Q12 Solution:

Input Image(I):

In each iteration, there is a 1 split operation which is followed by a merge operation in the iterative quadtree split-and-merge algorithm. We can perform the split operation only on non-uniform regions. This process is continued till no more non-uniform regions remain.

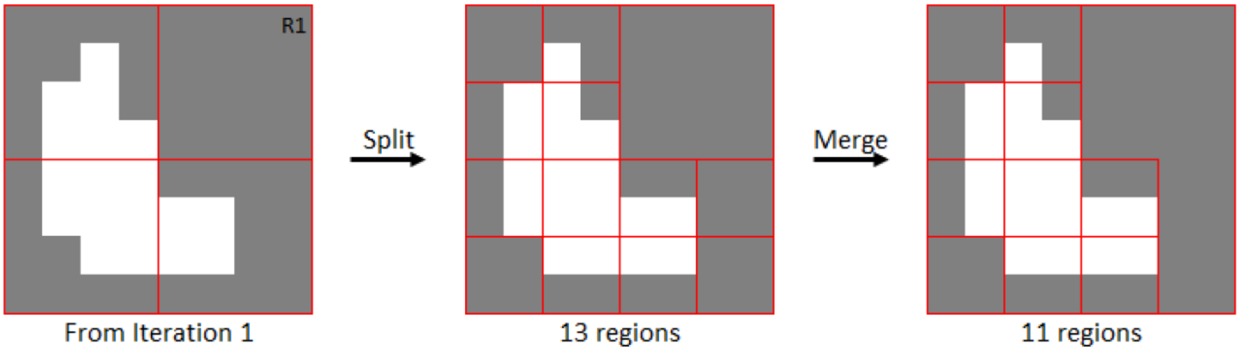
Iteration 1:



There are 3 things which we need to consider:

(a) The input image is divided into four equal parts.
(b) There are no more multiple regions so there is no way to perform merge operation.
(c) The output is passed as the input to the next iteration.

Iteration 2:



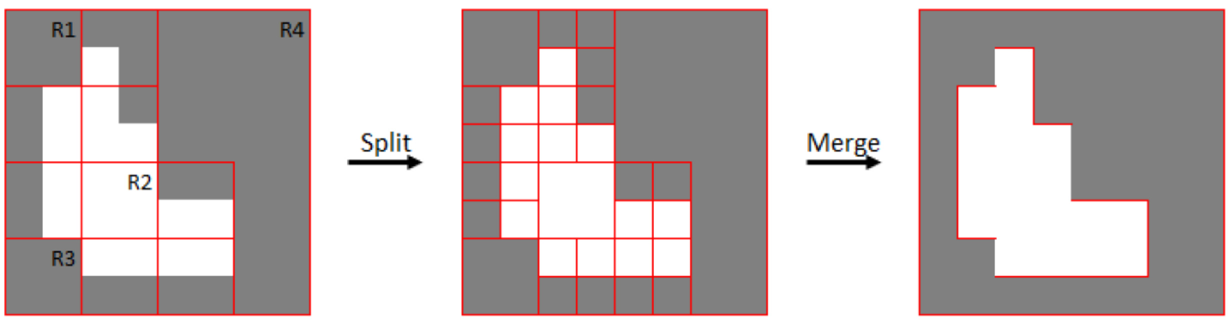
There are 3 things which we need to consider:

(a) Here, R1 is the only region which is uniform and the other regions are divided into 4 equal parts as shown in step 2 (with 13 regions).

(b) All the uniform regions are merged together after the split.

(c) The output is passed to the input of the next iteration.

Iteration 3:



There are 3 things which we need to consider:

(a) Here, R1, R2, R3, and R4, are the uniform regions so we will not split them. All the other regions will be split into 4 equal regions as shown in the step 2 of the above figure (with 32 regions).

(b) The uniform regions are split and merged together.

(c) So, the output of this iteration has correctly segmented the image into 2 parts as shown in step 3 (with 2 regions). This is the final output image.