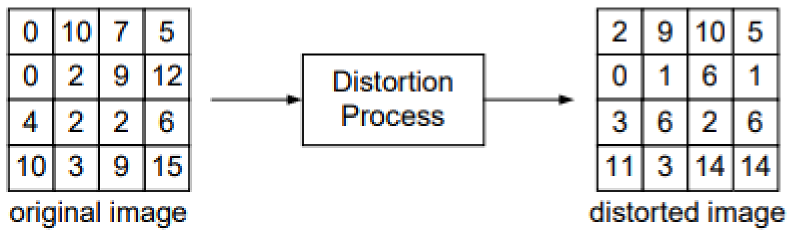
ECE 613, Winter 2022, Homework 1

Tarun Joseph, WatID: 20916617

Q1 Solution:



Let the original image be represented by matrix x and the distorted image by matrix y and N be the total number of pixels. x_i denotes the i^{th} pixel similarly y_i. i ranges from $1-16$ as each image contains 16 pixels in total, therefore $N = 16$.

Mean Absolute Error(MAE)

$$
\begin{aligned}
MAE &= \sum_{i=1}^{N} \frac{|y_i - x_i|}{N} \\
&= \frac{2+1+3+0+1+3+11+1+4+0+0+1+0+5+1}{16} \\
&= \frac{33}{16} \\
&= \mathbf{2.0625}
\end{aligned}
$$

Mean Squared Error(MSE)

$$
\begin{aligned}
MSE &= \sum_{i=1}^{N} \frac{(y_i - x_i)^2}{N} \\
&= \frac{4+1+9+0+1+9+121+1+16+0+0+1+0+25+1}{16} \\
&= \frac{189}{16} \\
&= \mathbf{11.8125}
\end{aligned}
$$

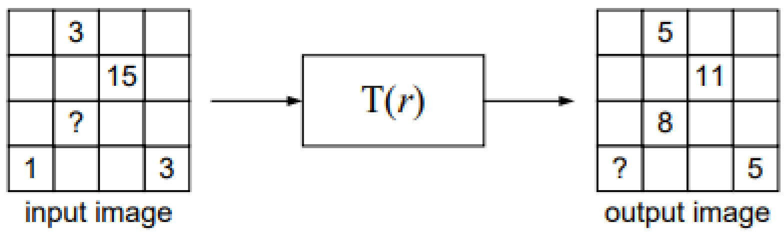
Peak Signal-to-Noise Ratio(PSNR)

for a 4 bits/pixel the MAX_I is the maximum possible pixel intensity which is given by $2^B - 1$, where B is the number of bits. For the given example $MAX_I = (2^4 - 1) = 15$

$$
\begin{aligned}
PSNR &= 10 \log\left(\frac{MAX_I^2}{MSE}\right) \\
&= 10 \log\left(\frac{225}{11.8125}\right) \\
&= \mathbf{12.7984}
\end{aligned}
$$

Q2 Solution:

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



From the given values in the input and output images we can for two equations to solve for α and β. The equations are as follows

$$\begin{aligned} 5 &= \alpha \log(1+3) + \beta \\ 5 &= 2\alpha + \beta \end{aligned} \tag{1}$$

$$\begin{aligned} 11 &= \alpha \log(1+15) + \beta \\ 11 &= 4\alpha + \beta \end{aligned} \tag{2}$$

(a) Solving Equations 1 and 2 **we get** $\alpha = 3$ **and** $\beta = -1$

Substituting the α and β values in the transformation equation we get

$$T(r) = 3\log_2(1+r) - 1 \tag{3}$$

(b) To calculate the value of ? in the output image substitute the corresponding input image pixel value in Equation 3, from this we get the pixel value in the output image is **2**

(c) To calculate the value of ? in the input image substitute the corresponding output image pixel value in Equation 3, from this we get the pixel value in the input image is **7**

Q3 Solution:

Assuming the intensity value 0 corresponds to black pixel and 1 corresponds to white the following would be the result of each of the transformations to the given image

1.) $s = r^2$
This 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**

2.) $s = 1 - r$
This transformation is equivalent to inverting the image and that can be seen in the output image **C**

3.) $s = 0.5r + 0.25$
This transformation halves the original pixel intensity and adds a marginal value back, causing new intensity to be lower than the original when original value is greater than 0.5 and greater than the original when original value is less than 0.5. Therefore the output image becomes lighter in the darker regions and darker in the light regions. This result can be seen in image **D**

4.)

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

The above transformation converts the input image to a binary image as it assign either 0 or 1 to each pixel depending on the threshold. This can be seen in the output image **B**

5.)

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

In this transformation the lower pixel values are made darker and the higher pixel values brighter because of the threshold set. The middle range is spread out using a linear transformation that makes the image brighter than the original one. This enhancement can be seen in the output image **A**

Q4 Solution:

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

a) Full scale contrast stretch is done by using he following equation.

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

For the given image $r_{max} = 14$ and $r_{min} = 6$, It is given that the input image is 4bits/pixel, therefore $(2^B - 1) = 15$.

Therefore,

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

Substituting the pixel values from the given image we have

$$\begin{aligned}
s(6) &= \text{round}(1.875 \times (6 - 6)) = 0 \\
s(7) &= \text{round}(1.875 \times (7 - 6)) = 2 \\
s(11) &= \text{round}(1.875 \times (11 - 6)) = 9 \\
s(12) &= \text{round}(1.875 \times (12 - 6)) = 11 \\
s(13) &= \text{round}(1.875 \times (13 - 6)) = 13 \\
s(14) &= \text{round}(1.875 \times (14 - 6)) = 15
\end{aligned}$$

Using the above calculations as a lookup table, the full scale contrast stretch image is as follows.

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

b) For performing histogram equalisation first we need to compute the cumulative histogram $Q(k)$ from the histogram $H(k)$ and then apply full scale contrast stretch to $Q(k)$ to get 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

Therefore the resulting intermediate image is as follows,

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

Applying full scale contrast stretch to the intermediate image. For the image $r_{max} = 16$ and $r_{min} = 2$, It is given that the input image is 4bits/pixel, therefore $(2^B - 1) = 15$.

Therefore,

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

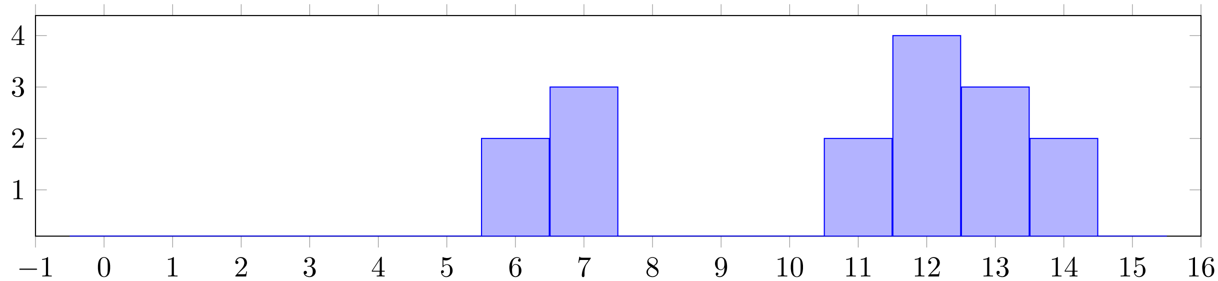
Substituting the pixel values from the intermediate image we have

$$\begin{aligned} s(2) &= \text{round}(1.071 \times (2 - 2)) = 0 \\ s(5) &= \text{round}(1.071 \times (5 - 2)) = 3 \\ s(7) &= \text{round}(1.071 \times (7 - 2)) = 5 \\ s(11) &= \text{round}(1.071 \times (11 - 2)) = 10 \\ s(14) &= \text{round}(1.071 \times (14 - 2)) = 13 \\ s(16) &= \text{round}(1.071 \times (16 - 2)) = 15 \end{aligned}$$

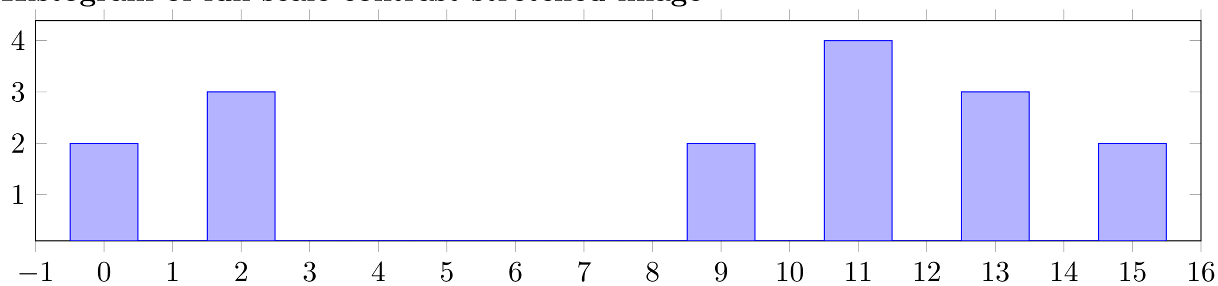
Using the above calculations as a lookup table, the full scale contrast stretch image is as follows.

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

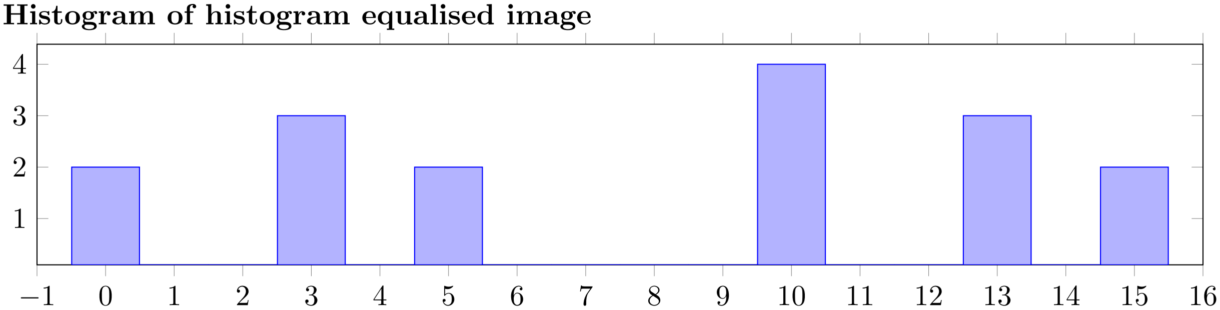
c) **Histogram of original image**



Histogram of full scale contrast stretched image



Histogram of histogram equalised image



Q5 Solution:

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

original image

Zero padding the original image to retain the same size as input in the output after the filter operation.

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

a) 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 slide the filter F_1 on the zero padded image with a stride of 1 and take the inner product. The output pixel is the sum all the elements of the inner product. $O(m,n) = \sum \langle I, F_1 \rangle$.
Therefore the filtered output image 1 is,

$$O_1 = \begin{array}{|c|c|c|c|}\hline 4 & 3.25 & 3.75 & 2.75 \\ \hline 8.25 & 9.25 & 3.5 & 3 \\ \hline 8.25 & 4 & 8.75 & 3.25 \\ \hline 3.25 & 8.5 & 3.75 & 3.5 \\ \hline \end{array}$$

b) Filter 2

$$F_2 = \frac{1}{4} \begin{array}{|c|c|c|}\hline -1 & 0 & -1 \\ \hline 0 & 4 & 0 \\ \hline -1 & 0 & -1 \\ \hline \end{array}$$

Following similar steps as in part a) 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

c Filter 3 is a combination of filter 1 and filter 2 i.e $F_{3=}F_1 + F_2$. Hence the output image can be obtained by adding the results from part a) and part b)

Therefore,

$$\begin{aligned} O_3 &= (F_1 + F_2) \otimes I \\ &= (F_1 \otimes I) + (F_2 \otimes I) \\ &= O_1 + O_2 \end{aligned}$$

$O_3 =$

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

Q6 Solution: Given,

$$X = \begin{bmatrix} 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \end{bmatrix} \qquad F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \qquad F_4^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

a) Computing 2D-DFT of the original image $\widetilde{X} = F_4 X F_4$ for a 4x4 image

$$\begin{aligned} \widetilde{X} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \\ 5 & 3 & 5 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \\ &= \begin{bmatrix} 24 & 8 & 24 & 8 \\ 0 & 0 & 0 & 0 \\ 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \\ &= \begin{bmatrix} 64 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

b) The inverse 2D-DFT of the given three 4×4 patterns is computed as follows $X = \frac{1}{N^2} F_4^* \widetilde{X} F_4^*$
For Pattern 1,

$$\begin{aligned} X_1 &= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \\ &= \frac{1}{16} \begin{bmatrix} 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \\ &= \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix} \end{aligned}$$

For Pattern 2,

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

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

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

For Pattern 3,

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

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

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

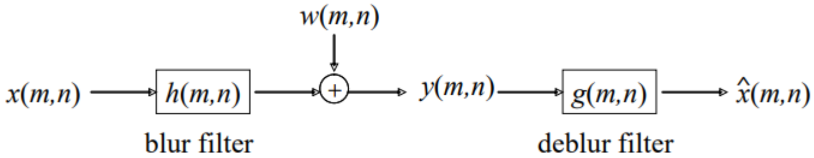
c) Adding the results of part b) we see that

$$X_1 + X_2 + X_3 = \begin{bmatrix}4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\end{bmatrix} + \begin{bmatrix}2 & -2 & 2 & -2\\ 2 & -2 & 2 & -2\\ 2 & -2 & 2 & -2\\ 2 & -2 & 2 & -2\end{bmatrix} + \begin{bmatrix}1 & -1 & 1 & -1\\ -1 & 1 & -1 & 1\\ 1 & -1 & 1 & -1\\ -1 & 1 & -1 & 1\end{bmatrix}$$

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

2-D DFT is invertible and has the preserving property. Therefore the sum of the inverse 2D-DFT of all the three patterns add up to the original image. This is due to the distribution law of inverse 2D-DFT, hence computing them independently and adding the results produces the original image.

Q7 Solution:



Given: $,\sigma_X^2 = 100,\qquad \sigma_W^2 = 100 \qquad H(u,v) = \begin{bmatrix} 1 & -0.2-0.2j & 0 & -0.2+0.2j \\ -0.2-0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2+0.2j & 0.05 & 0 & -0.05j \end{bmatrix}$

a) Design deblur filter $G(u,v)$ using inverse filtering approach.

$$\begin{aligned} G(u,v) &= \frac{1}{H(u,v)} \\ &= \frac{1}{\begin{bmatrix} 1 & -0.2-0.2j & 0 & -0.2+0.2j \\ -0.2-0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2+0.2j & 0.05 & 0 & -0.05j \end{bmatrix}} \\ &= \begin{bmatrix} 1 & -2.5+2.5j & \text{inf} & -2.5-2.5j \\ -2.5+2.5j & -20j & 0 & 20 \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ -2.5-2.5j & 20 & 0 & 20j \end{bmatrix} \end{aligned}$$

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

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

$$G(u,v) = \begin{cases} \frac{1}{H(u,v)} & |H(u,v)| > 0.03 \\ 0 & |H(u,v)| \leq 0.03 \end{cases} \quad = \begin{bmatrix} 1 & -2.5+2.5j & 0 & -2.5-2.5j \\ -2.5+2.5j & -20j & 0 & 20 \\ 0 & 0 & 0 & 0 \\ -2.5-2.5j & 20 & 0 & 20j \end{bmatrix}$$

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

$$G(u,v) = \begin{cases} \frac{1}{H(u,v)} & |H(u,v)| > 0.1 \\ 0 & |H(u,v)| \leq 0.1 \end{cases} \quad = \begin{bmatrix} 1 & -2.5+2.5j & 0 & -2.5-2.5j \\ -2.5+2.5j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2.5-2.5j & 0 & 0 & 0 \end{bmatrix}$$

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

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

e) Design a deblur filter G(u,v) using the Wiener filtering approach. Given $\sigma_X^2 = 100, \sigma_W^2 = 25$. Therefore, $K = \frac{\sigma_W^2}{\sigma_x^2} = 0.25$

$$\begin{aligned} G(u,v) &= \frac{H^*(u,v)}{|H(u,v)|^2 + K} \\ &= \frac{\begin{bmatrix} 1 & -0.2+0.2j & 0 & -0.2-0.2j \\ -0.2+0.2j & -0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2-0.2j & 0.05 & 0 & 0.05j \end{bmatrix}}{\left(\begin{bmatrix} 1 & 0.2828 & 0 & 0.2828 \\ 0.2828 & 0.05 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0.2828 & 0.05 & 0 & 0.05 \end{bmatrix}\right)^2 + 0.25} \\ &= \frac{\begin{bmatrix} 1 & -0.2+0.2j & 0 & -0.2-0.2j \\ -0.2+0.2j & -0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2-0.2j & 0.05 & 0 & 0.05j \end{bmatrix}}{\begin{bmatrix} 1.25 & 0.33 & 0.25 & 0.33 \\ 0.33 & 0.2525 & 0.25 & 0.2525 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.33 & 0.2525 & 0.25 & 0.2525 \end{bmatrix}} \\ &= \begin{bmatrix} 0.8 & -0.61+0.61j & 0 & -0.61-0.61j \\ -0.61+0.61j & -0.2j & 0 & 0.2 \\ 0 & 0 & 0 & 0 \\ -0.61-0.61j & 0.2 & 0 & 0.2j \end{bmatrix} \end{aligned}$$

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

$$h(m,n) = \frac{1}{N^2} F_4^* H(u,v) F_4^*$$
$$= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 1 & -0.2-0.2j & 0 & -0.2+0.2j \\ -0.2-0.2j & 0.05j & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ -0.2+0.2j & 0.05 & 0 & -0.05j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$
$$= \begin{bmatrix} 0.0188 & 0.0563 & 0.0563 & 0.0188 \\ 0.0563 & 0.1188 & 0.1188 & 0.0562 \\ 0.0562 & 0.1188 & 0.1188 & 0.0562 \\ 0.0187 & 0.0562 & 0.0562 & 0.0187j \end{bmatrix}$$

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

$$g(m,n) = \frac{1}{N^2} F_4^* G(u,v) F_4^*$$
$$= \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 1 & -2.5+2.5j & 0 & -2.5-2.5j \\ -2.5+2.5j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2.5-2.5j & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$
$$= \begin{bmatrix} -0.5625 & -0.5625 & 0.0625 & 0.0625 \\ -0.5625 & -0.5625 & 0.0625 & 0.0625 \\ 0.0625 & 0.0625 & 0.6875 & 0.6875 \\ 0.0625 & 0.0625 & 0.6875 & 0.6875 \end{bmatrix}$$

Q8 Solution:

Given: , $\sigma_X^2 = 400, \qquad \sigma_W^2 = 100 \qquad h(m,n) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

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

$$
\begin{aligned}
H(u,v) &= F_4 h(m,n) F_4 \\
&= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \\
&= \begin{bmatrix} 20 & -2-2j & 0 & -2+2j \\ -2-2j & 2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2+2j & 2 & 0 & -2j \end{bmatrix}
\end{aligned}
$$

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

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

$$
\begin{aligned}
G(u,v) &= \frac{1}{H(u,v)} \\
&= \frac{1}{\begin{bmatrix} 20 & -2-2j & 0 & -2+2j \\ -2-2j & 2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2+2j & 2 & 0 & -2j \end{bmatrix}} \\
&= \begin{bmatrix} 0.05 & -0.25+0.25j & \text{inf} & -0.25-0.25j \\ -0.25+0.25j & -0.5j & 0 & 0.5 \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ -0.25-0.25j & 0.5 & 0 & 0.5j \end{bmatrix}
\end{aligned}
$$

$$
G(u,v) = \begin{cases} \frac{1}{H(u,v)} & |H(u,v)| > 1 \\ 0 & |H(u,v)| \le 1 \end{cases} \quad = \begin{bmatrix} 0.05 & -0.25+0.25j & 0 & -0.25-0.25j \\ -0.25+0.25j & -0.5j & 0 & 0.5 \\ 0 & 0 & 0 & 0 \\ -0.25-0.25j & 0.5 & 0 & 0.5j \end{bmatrix}
$$

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

$$G(u,v) = \begin{cases} \frac{1}{H(u,v)} & |H(u,v)| > 3 \\ 0 & |H(u,v)| \leq 3 \end{cases} \quad = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

d) Design a deblur filter $G(u,v)$ using the Wiener filtering approach.Given $\sigma_X^2 = 400, \sigma_W^2 = 100$.

Therefore, $K = \frac{\sigma_W^2}{\sigma_x^2} = 0.25$

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

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

$$= \frac{\begin{bmatrix} 20 & -2+2j & 0 & -2-2j \\ -2+2j & -2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2-2j & 2 & 0 & 2j \end{bmatrix}}{\begin{bmatrix} 400.25 & 8.25 & 0.25 & 8.25 \\ 8.25 & 4.25 & 0.25 & 4.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 8.25 & 4.25 & 0.25 & 4.25 \end{bmatrix}}$$

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

Q9 Solution:

Given, I =

7	2	7	2
2	7	2	7
7	2	7	2
2	7	2	7

a) Zero padding the original image to retain the same size as input in the output after the filter operation.

$I_1 =$

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

Linear shift Invariant Filter $(F_1) = \frac{1}{6}$

0	1	0
1	2	1
0	1	0

To calculate the filtered image we slide the filter F_1 on the zero padded image with a stride of 1 and take the inner product. The output pixel is the sum all the elements of the inner product. $O(m,n) = \text{round}(\sum\langle I_1, F_1\rangle)$.
Therefore the filtered output image is,

$O_1 =$

3	4	3	3
4	4	5	3
3	5	4	4
3	5	4	4

Let the input image I_1 be represented as a matrix x and the output filtered image O_1 be represented by y

Mean Absolute Error(MAE)

$$
\begin{aligned}
MAE &= \sum_{i=1}^{N} \frac{|y_i - x_i|}{N} \\
&= \frac{4+2+4+1+2+3+3+4+4+3+3+2+1+4+2+4}{16} \\
&= \frac{46}{16} \\
&= \mathbf{2.875}
\end{aligned}
$$

Mean Squared Error(MSE)

$$\begin{aligned} MSE &= \sum_{i=1}^{N} \frac{(y_i - x_i)^2}{N} \\ &= \frac{16+4+16+1+4+9+9+16+16+9+9+4+1+16+4+16}{16} \\ &= \frac{150}{16} \\ &= \mathbf{9.375} \end{aligned}$$

Peak Signal-to-Noise Ratio(PSNR)

for a 4 bits/pixel the MAX_I is the maximum possible pixel intensity which is given by $2^B - 1$, where B is the number of bits. For the given example $MAX_I = (2^4 - 1) = 15$

$$\begin{aligned} PSNR &= 10\log\left(\frac{MAX_I^2}{MSE}\right) \\ &= 10\log\left(\frac{225}{9.375}\right) \\ &= \mathbf{13.8021} \end{aligned}$$

b) Passing the original image through a median filter with replicate padding on the input image, Therefore

$I_2 =$

7	7	2	7	2	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

To calculate the filtered image using a 3x3 median filter on the replicate padding input image we take a 3x3 section of the input image and pass the median value from all the intensities in the 3x3 section, ie. $\{x_1, x_2, ..., x_9\} = x_5$. for example,

$$O_2(0,0) = \text{median}\left(\begin{array}{|c|c|c|}\hline 7 & 7 & 2 \\ \hline 7 & 7 & 2 \\ \hline 2 & 2 & 7 \\ \hline \end{array}\right) = \text{median}(2,2,2,2,7,7,7,7,7) = 7$$

Therefore the filtered output image is,

$O_2 =$

7	7	2	2
7	7	2	2
2	2	7	7
2	2	7	7

Let the input image I_2 be represented as a matrix x and the output filtered image O_2 be represented by y

Mean Absolute Error(MAE)

$$\begin{aligned}
MAE &= \sum_{i=1}^{N} \frac{|y_i - x_i|}{N} \\
&= \frac{5+5+5+5+5+5+5+5}{16} \\
&= \frac{40}{16} \\
&= \mathbf{2.5}
\end{aligned}$$

Mean Squared Error(MSE)

$$\begin{aligned}
MSE &= \sum_{i=1}^{N} \frac{(y_i - x_i)^2}{N} \\
&= \frac{25+25+25+25+25+25+25+25}{16} \\
&= \frac{200}{16} \\
&= \mathbf{12.5}
\end{aligned}$$

Peak Signal-to-Noise Ratio(PSNR)

for a 4 bits/pixel the MAX_I is the maximum possible pixel intensity which is given by $2^B - 1$, where B is the number of bits. For the given example $MAX_I = (2^4 - 1) = 15$

$$\begin{aligned}
PSNR &= 10 \log\left(\frac{MAX_I^2}{MSE}\right) \\
&= 10 \log\left(\frac{225}{12.5}\right) \\
&= \mathbf{12.5527}
\end{aligned}$$

Q10 Solution:

$$\text{Given, I} = \begin{array}{|c|c|c|c|}\hline 9 & 8 & 7 & 6 \\ \hline 8 & 7 & 13 & 5 \\ \hline 7 & 6 & 5 & 4 \\ \hline 6 & 1 & 4 & 3 \\ \hline \end{array}$$

a) Applying replicate padding on the input image we get

$$I = \begin{array}{|c|c|c|c|c|c|}\hline 9 & 9 & 8 & 7 & 6 & 6 \\ \hline 9 & 9 & 8 & 7 & 6 & 6 \\ \hline 8 & 8 & 7 & 13 & 5 & 5 \\ \hline 7 & 7 & 6 & 5 & 4 & 4 \\ \hline 6 & 6 & 1 & 4 & 3 & 3 \\ \hline 6 & 6 & 1 & 4 & 3 & 3 \\ \hline \end{array}$$

Applying the median filter with the given pattern on the padded image we will get the following result,

$$O(0,0) = \text{median}\left(\begin{array}{|c|c|c|}\hline & 9 & \\ \hline 9 & 9 & 8 \\ \hline & 8 & \\ \hline \end{array}\right) = \text{median}(8, 8, 9, 9, 9) = 9$$

Apply the filter by shifting the filter one stride over the whole input image

Therefore output median filtered image is,

$$O = \begin{array}{|c|c|c|c|}\hline 9 & 8 & 7 & 6 \\ \hline 8 & 8 & 7 & 5 \\ \hline 7 & 6 & 5 & 4 \\ \hline 6 & 4 & 4 & 3 \\ \hline \end{array}$$

b) Use minimum filter of the same pattern as in part a) on the replicate padded image I.

Applying the minimum filter will produce the following result,

$$O(0,0) = \text{minimum}\left(\begin{array}{|c|c|c|}\hline & 9 & \\ \hline 9 & 9 & 8 \\ \hline & 8 & \\ \hline \end{array}\right) = \text{minimum}(8, 8, 9, 9, 9) = 8$$

Apply the filter by shifting the filter one stride over the whole input image

Therefore output minimum filtered image is,

$$O = \begin{array}{|c|c|c|c|}\hline 8 & 7 & 6 & 5 \\ \hline 7 & 6 & 5 & 4 \\ \hline 6 & 1 & 4 & 3 \\ \hline 1 & 1 & 1 & 3 \\ \hline \end{array}$$

c) Use maximum filter of the same pattern as in part a) on the replicate padded image I.

Applying the maximum filter will produce the following result,

$$O(0,0) = \text{minimum}\left(\begin{array}{|c|c|c|}\hline & 9 & \\ \hline 9 & 9 & 8 \\ \hline & 8 & \\ \hline\end{array}\right) = \text{maximum}(8,8,9,9,9) = 9$$

Apply the filter by shifting the filter one stride over the whole input image

Therefore output maximum filtered image is,

$$O = \begin{array}{|c|c|c|c|}\hline 9 & 9 & 13 & 7 \\ \hline 9 & 13 & 13 & 13 \\ \hline 8 & 7 & 13 & 5 \\ \hline 7 & 6 & 5 & 4 \\ \hline\end{array}$$

d) Filter the replicate padded image I with the order statistics filter $w_i = \{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\}$

Applying the filter will produce the following result,

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

Apply the filter by shifting the filter one stride over the whole input image

Therefore output maximum filtered image is,

$$O = \begin{array}{|c|c|c|c|}\hline 9 & 8 & 7 & 6 \\ \hline 8 & 8 & 6 & 5 \\ \hline 7 & 6 & 5 & 4 \\ \hline 6 & 4 & 4 & 3 \\ \hline\end{array}$$

Q11 Solution:
Input image(I) =

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 $= |g(m,n)| = |g_1(m,n) + |g_2(m,n)||$, Given $T = 22$

For Prewitt operator,

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

Applying each filter on the input image we get the following results,

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

Approximating the gradient magnitude by adding the modulus of $O_1(m,n)$ and $O_2(m,n)$ we get,

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

Applying the threshold of $T = 22$ on the output image

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

Therefore the edge map is given by,

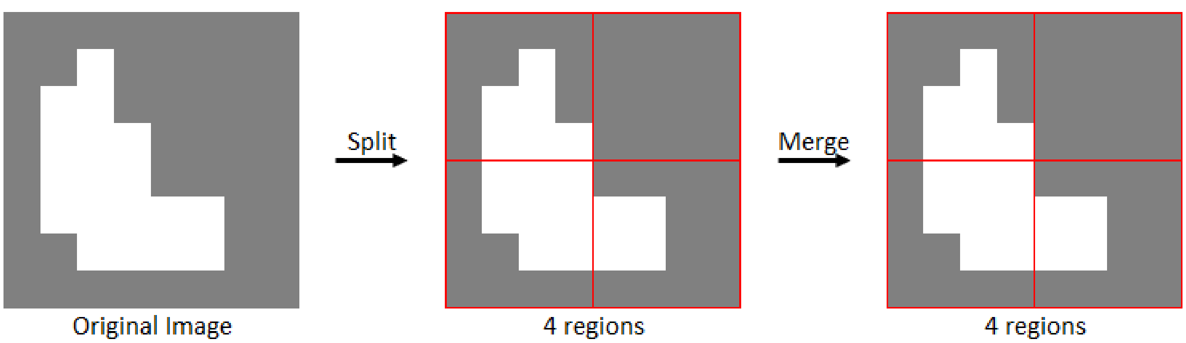
$$e(m,n) = \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

Q12 Solution:

Input image(I) =

The iterative quadtree split-and-merge algorithm has 1 split operation followed by a merge operation in each iteration. The split operation happens only on non uniform regions. The process is continued till no more non-uniform regions remain.

Iteration 1

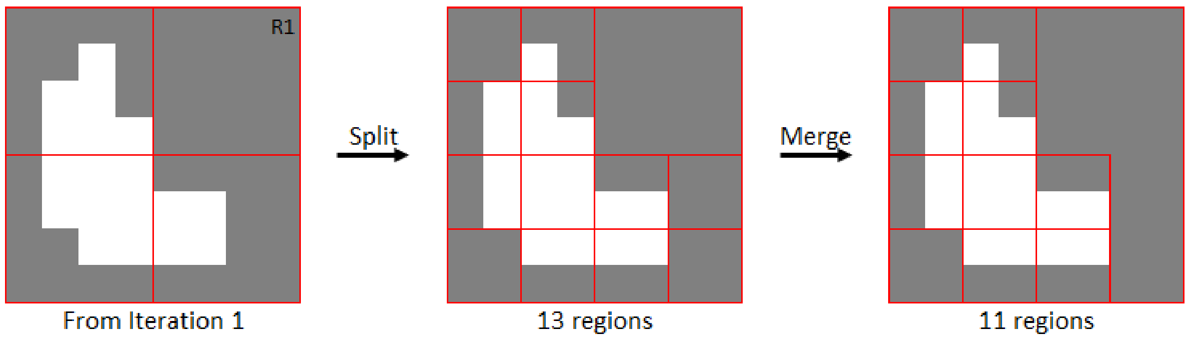


a) The input image is first divided into four equal parts.

b) There are no multiple uniform regions so no merge operation happens.

c) The output is passed as the input to the next iteration

Iteration 2

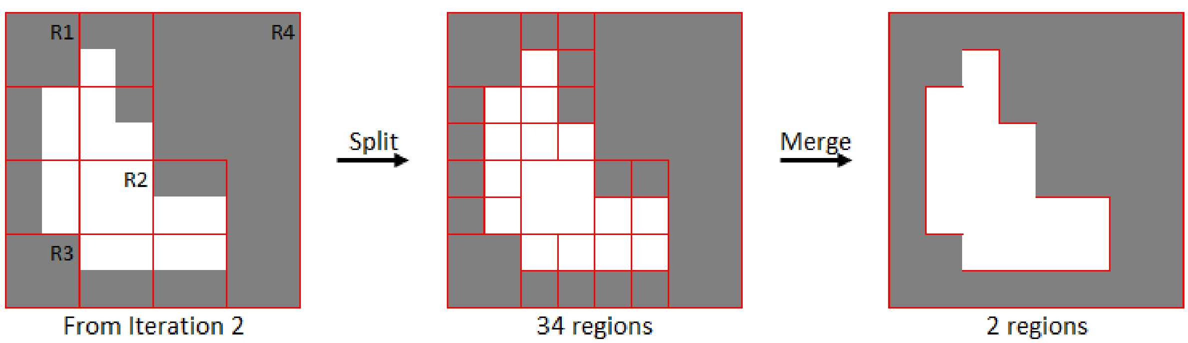


a) Regions R1 is the only uniform region the other regions are split again into 4 regions

b) The uniform regions after the split are merged together.

c) The output is passed as the input to the next iteration

Iteration 3



a) Regions R1, R2, R3 and R4 are the uniform regions, the other regions are split again into 4 regions

b) The uniform regions after the split are merged together.

c) The output of this iteration has correctly segmented the image into 2 parts.