Colab Link:

https://colab.research.google.com/drive/1CJV-I8xHmyz9VU2lUviyPRNJ5ZYFwGve?usp=sharing

Original Image:

Smoothing filter in spatial domain:

For smoothing the image in spatial domain, a type of mean filter called the average filter is used. In case of such filters, the filter is the average of the pixels contained in the neighborhood of the filter mask. The value of every pixel in the image is replaced with the average of grey levels in the neighborhood defined in the mask where all the coefficients in the filter are equal. As a result, the pixels with lower grey levels are replaced by the average resulting in the smoothening of the blurry part of the image on the right side and also the higher grey levels are also replaced with the average, resulting in loss of information in the high contrast part in the right side of the image as well, which smoothens and blurs the original image and results in the image below:

Filter used:

Blurring in the Spatial Domain:

Averaging = convolution with $\frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

= point multiplication of the transform with sinc

Smoothing filter in spatial domain:

For sharpening the original image, the Laplacian operator was used. It is a second-order derivative operator which is the Laplacian for a function (image) f(x,y). It is particularly good at finding the fine detail in an image. The equations that are used to derive a Laplacian operator for the filter are shown below:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Image sharpening in the way of using the Laplacian:

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

where,

$f(x, y)$ is input image,

$g(x, y)$ is sharpenend images,

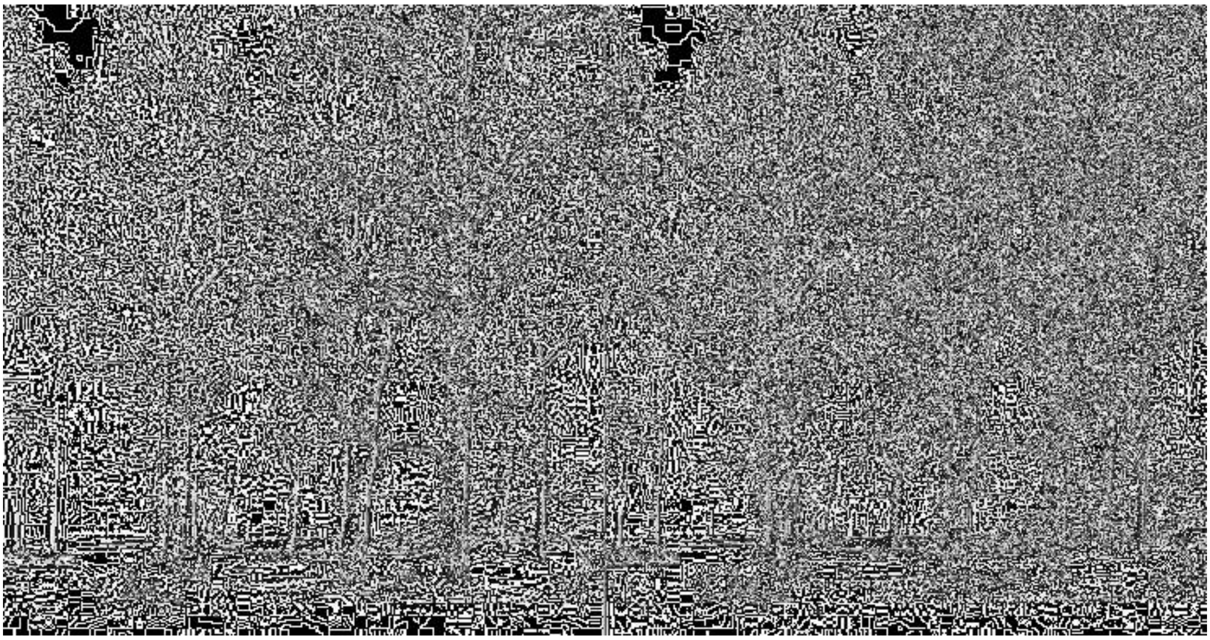
$c = -1$ if $\nabla^2 f(x, y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.

Filter used:

−1	−1	−1
−1	8	−1
−1	−1	−1

The above filter extracts the higher grey level values of the pixels in the image as a result of which only the pixels with the higher grey levels is extracted and everything else is left out, thus only extracting bright parts in both the blurry and high contrast part of the image, thus we can only see the edges of the trunks of the trees are extracted. The resultant image of using the Laplacian operator is shown below:



Smoothing using low pass filter in frequency domain:

In my work, the low pass filter was used to smoothen in the image to obtain the same result as using the average filter. Using the low pass filter only lets the lower values of the grey level histogram pass through the filter as a result making the image have less detail and thus blurring the blurry part of the image as well as the high contrast part on the right side of the image. The low pass filter used in my work is shown below:

```
[[0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0],
 [0, 0, 1, 0, 0],
 [0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0]
```

The resultant image of using this filter is also shown below:

Sharpening using High Pass Filter in the frequency domain:

Filter Used:

```
[[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1],
[1, 1, 0, 1, 1],
[1, 1, 1, 1, 1],
[1, 1, 1, 1, 1]
```

Using the above filter blocks the low grey value parts of the image, and passes the high grey value pixels of the image in the grey level-frequency histogram. Thus, only the brighter parts of the image both in the blurry part on the left and the high contrast part on the right is passed resulting in a sharper image similar to that of the sharpened image using the Laplacian operator.