Computer Vision Task 1 Report
Team 06

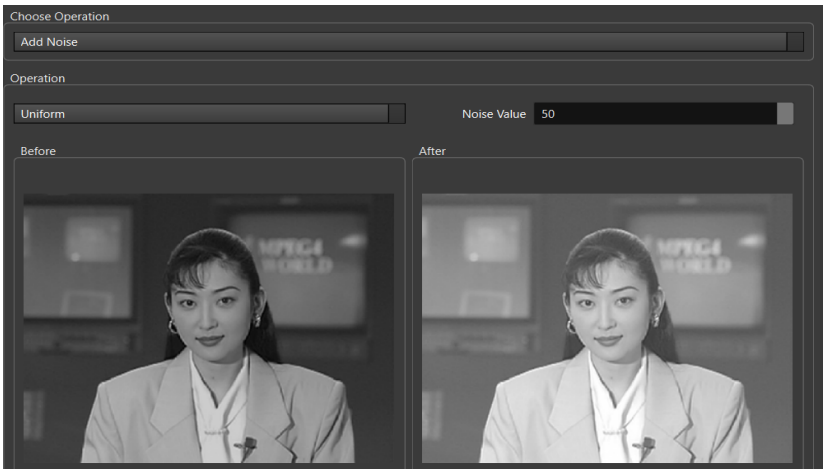
Team Members:

Name	Section	BN
Romaisaa Shrief	1	36
Kamel Mohamed	2	11
Youssef Shaban Mohamed	2	56

Table of Content:

Part 1, Adding Noise

Uniform noise is added to an image by randomly introducing a value within a specified range to each pixel in the image. The range is usually determined by the image's bit depth, and the added noise is uniform across the entire image. This method is often used to replicate electronic noise and test the robustness of image processing algorithms.



Gaussian noise is added by randomly introducing a value to each pixel in an image, following a Gaussian distribution. The degree of noise added is regulated by the standard deviation of the distribution. Gaussian noise is often utilized to simulate random variations in lighting conditions and test image processing algorithms' performance in noise reduction and image enhancement.

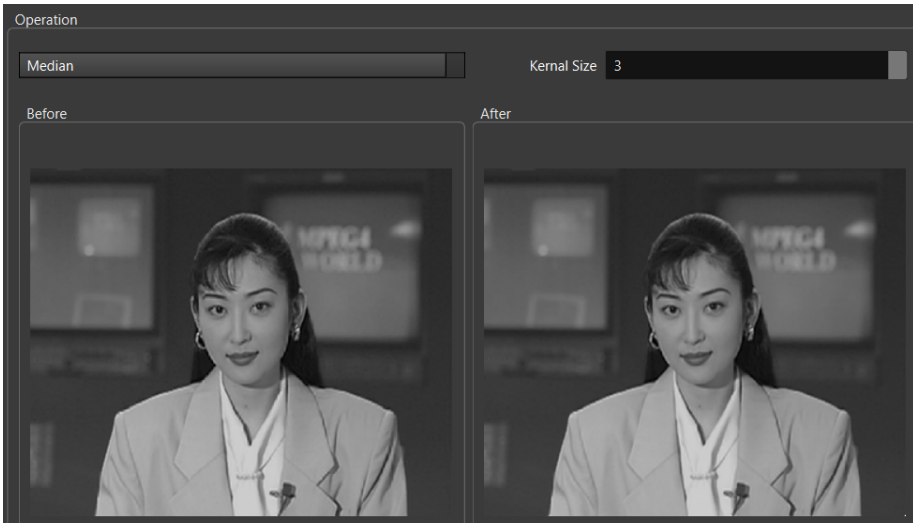
Salt and pepper noise is created by randomly setting some pixels in an image to either the maximum or minimum value of the image's range while leaving others unmodified. This approach replicates issues with image acquisition or transmission, such as missing or corrupted data. Salt and pepper noise can be difficult to remove from an image and can hinder image processing algorithms that depend on consistent pixel values.

Part 2, Image Filtering

Average Filter:
The average filter is another type of linear filter that is used for smoothing images. It works by taking the average of the pixel values in a neighborhood around each pixel in the image. The size of the neighborhood can be adjusted, which determines the level of smoothing.

Gaussian Filter:

The Gaussian filter is a type of linear filter that is commonly used for image processing tasks, particularly for reducing noise and smoothing images. The filter works by convolving each pixel in the image with a Gaussian function, which essentially blurs the image and reduces the high-frequency noise. The amount of blurring can be controlled by adjusting the standard deviation of the Gaussian function.



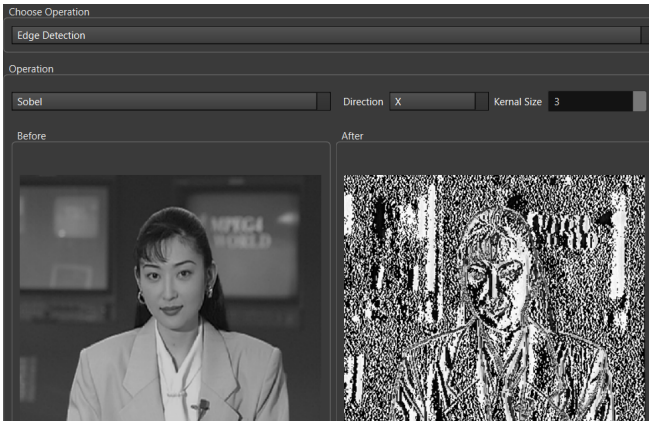
Median Filter:

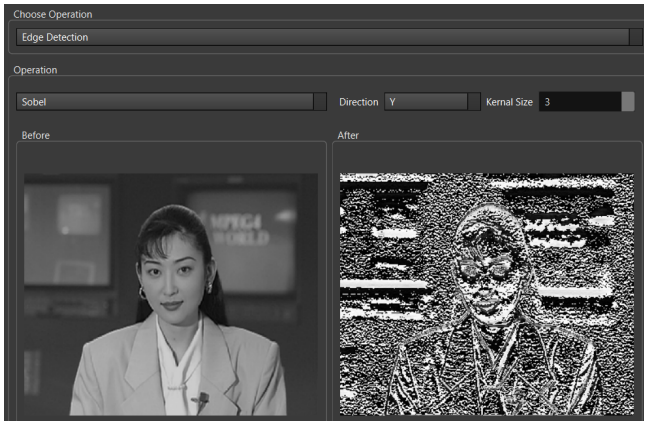
The median filter is a nonlinear filter that is also used for noise reduction. It works by replacing each pixel in the image with the median value of the pixel values in a neighborhood around it. This is particularly effective at removing salt-and-pepper noise, where individual pixels have very high or very low values.

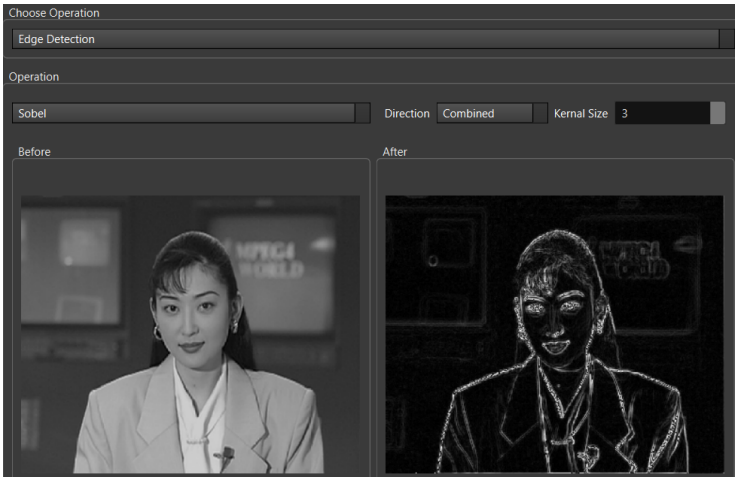
PART 3, Edge Detection

Sobel Edge Detection:

The Sobel operator is a type of gradient-based edge detection technique. It works by convolving the image with a small kernel that calculates the gradient at each pixel. The gradient is then used to determine the edges in the image by looking for places where the gradient magnitude is high.

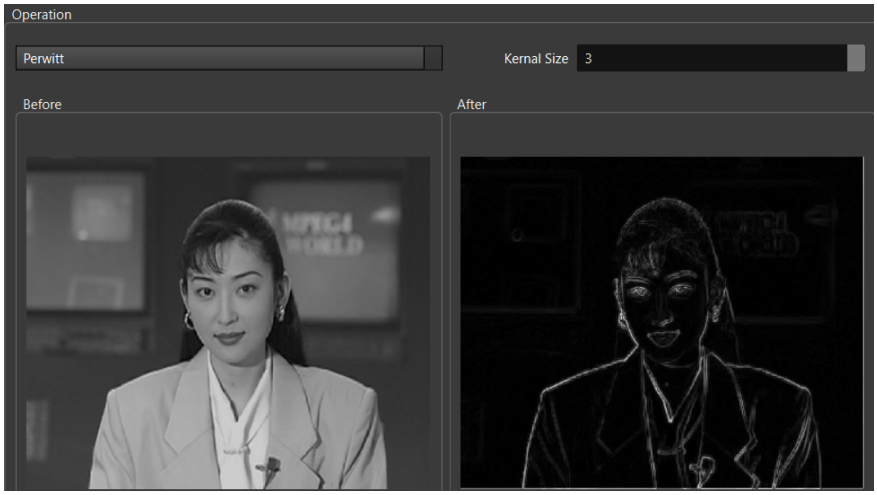






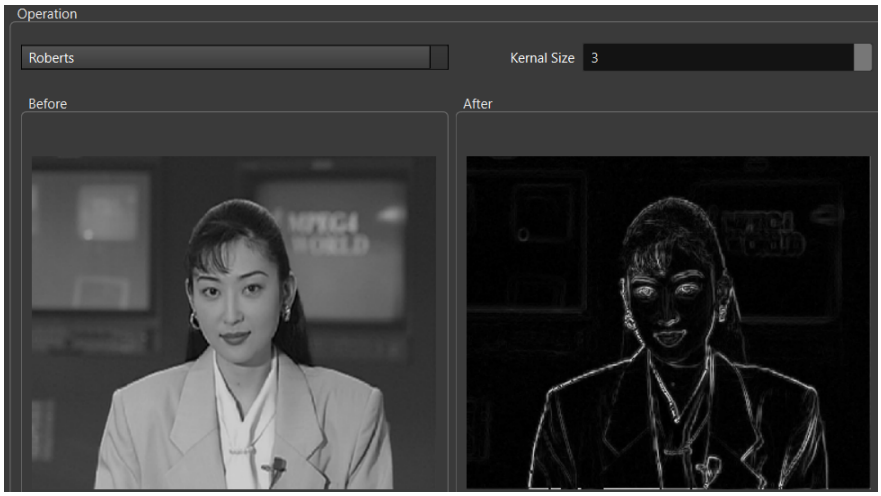
Prewitt Edge Detection:

The Prewitt operator is similar to the Sobel operator, but it uses a slightly different kernel to calculate the gradient. Like the Sobel operator, it is a gradient-based edge detection technique that looks for areas where the gradient magnitude is high.



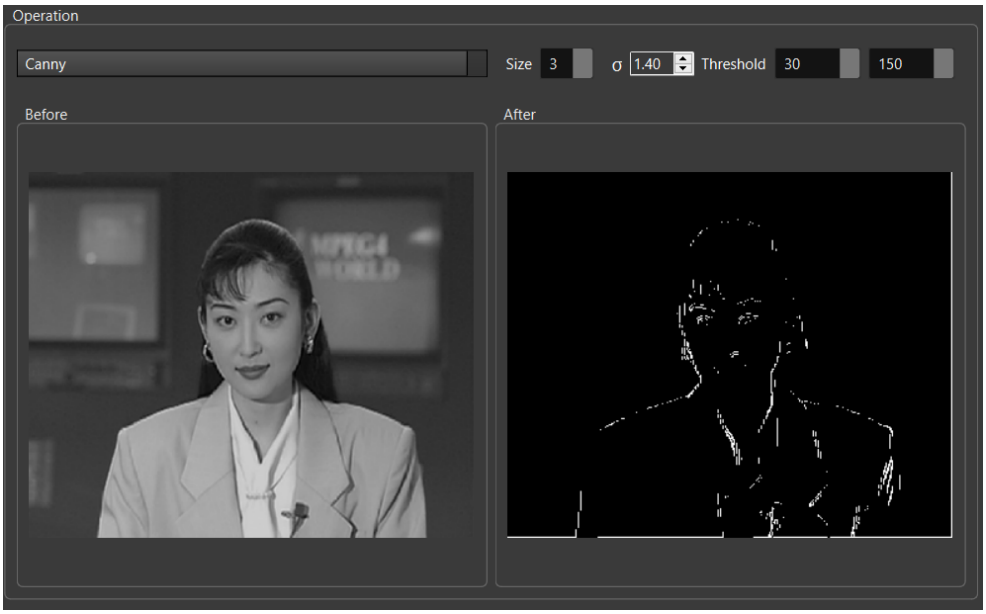
Roberts Edge Detection:

The Roberts operator is another type of gradient-based edge detection technique. It works by calculating the gradient in two diagonal directions using a small kernel. The gradient magnitudes are then combined to determine the edges in the image.



Canny Edge Detection:

The Canny edge detection algorithm is a multi-stage process that is commonly used for detecting edges in images. It works by first smoothing the image with a Gaussian filter, then calculating the gradient using the Sobel operator.
Non-maximum suppression is then used to thin out the edges, and hysteresis thresholding is used to identify the strong and weak edges. The final output is a binary image that shows the locations of the detected edges.

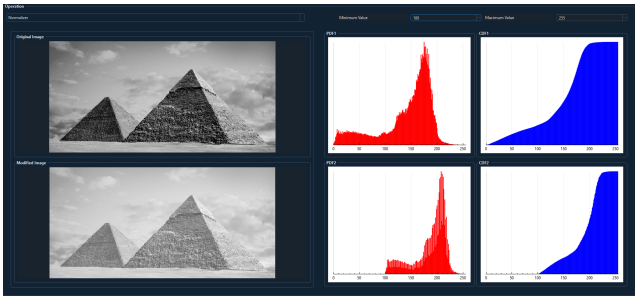


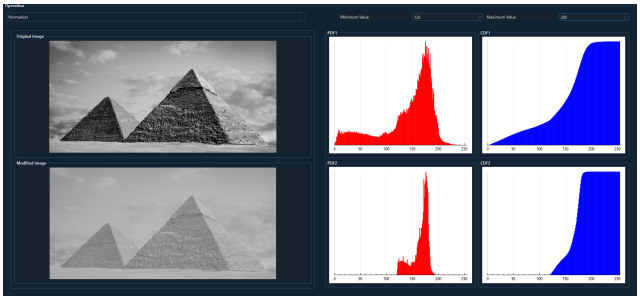
Part 4 - Image Normalization

In this method we rescale grayscale image values from their original scale to any new scale from 0 to 255 that comes from the user and then remap all pixels to this new scale according to this simple equation:

$$I_N = (I - \text{Min})\frac{\text{newMax} - \text{newMin}}{\text{Max} - \text{Min}} + \text{newMin}$$

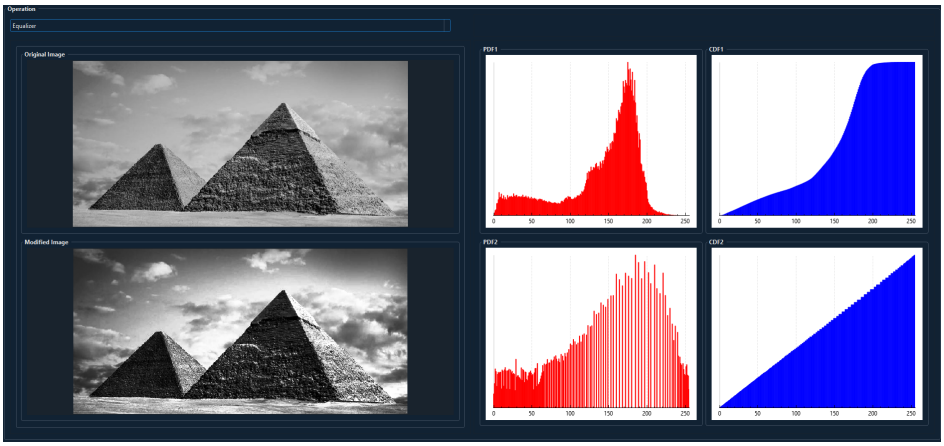
We can see the output from this method we have implemented:





Part 5 - Histogram Equalization

Using image histogram we can see the distribution of value, and in this method, our target is to remap values to make CDF like a straight line and enhance the contrast of an image.



Part 6 - Thresholding

In this part, we want to split our image into two splits 0 and 255, any value above this threshold becomes 255, and any value below become 0, we decide this threshold using the Otsu method, with these equations:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \qquad \mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \qquad \mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)}$$

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_0(t)(\mu_0 - \mu_T)^2 + \omega_1(t)(\mu_1 - \mu_T)^2$$
$$= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$$

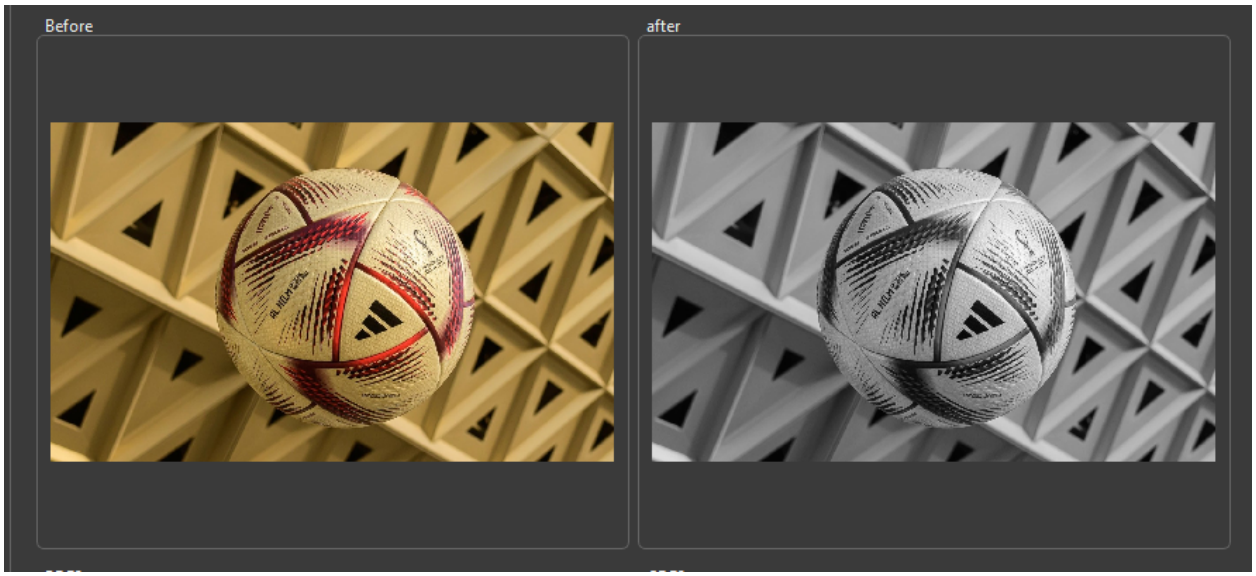
There are 2 ways of Thresholding:

1- Global Thresholding:
Using this method in the whole image

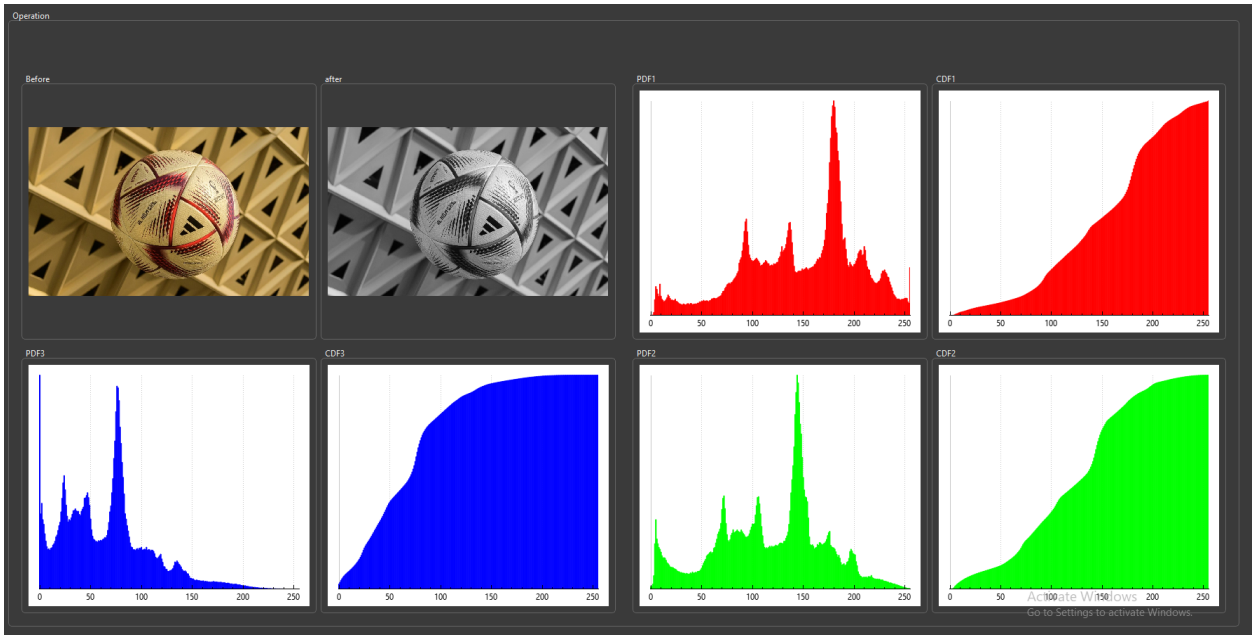
2- Local Thresholding:
Split image into small pieces and apply this method on every piece separately.

Part 7: RGB to Grayscale with Histogram:

Converting from RGB to Grayscale includes different ways that may take an average of three channels or take middle HSL with eq: max+min /2. But the most efficient one called the LUMA equation gives every channel a different desirable weight and it's the formula: Gray = R*0.2989 + G*0.587 + B*0.114



Then using the separated image we plot the distribution of each channel as follows:



Part 8: Frequency Domain Filters:

In this type of filter we mainly convert image from spatial domain to frequency/ Fourier domain and choose needed frequencies, neglect others then transform it again to the time domain. Fourier Domain is known to be a sparse domain so, the low frequencies are gathered together in the Fourier image center, and high frequencies are all over the rest of the image.

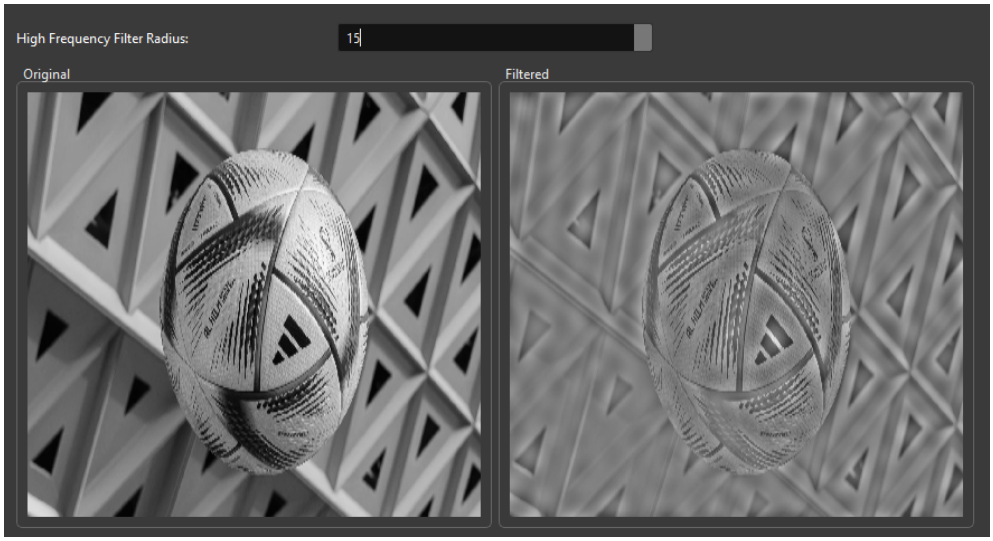
1- Low pass filter:

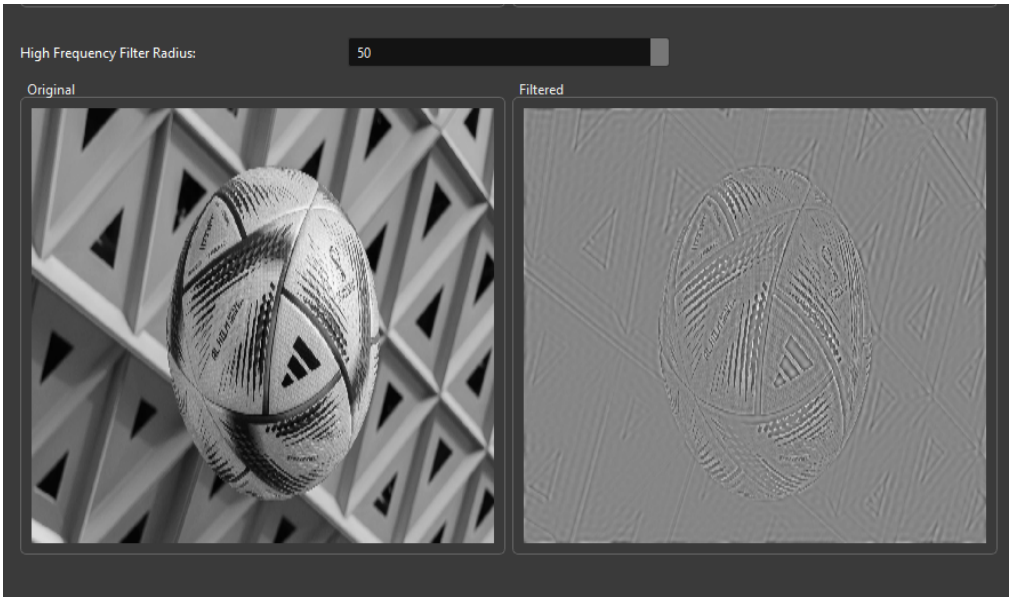
Here we included the circular center (including low frequencies) giving the users ability to change the circle radius to be included as(cut off frequency)

Comment: Low pass filters: Make images more smooth and blur, increasing the radius leads to sharpener one as we include higher frequencies.

2- High pass filter:

The opposite of low pass one, we just exclude the circular center:





Comment: High pass filters care only about high changes/edges, so, we notice that filtered images show image edges. Decreasing the radius of the circle excluded leads to a tightening range of changed notices(sharper).

Part 9: Hybrid Images:

A Hybrid image is a combination of 2 different images one is low-frequency filtered and the other is high-frequency filtered one, then we sum them together with including gain for each one or not according to equation: