2️⃣

A2 - Jay Vora - Student # - 203321900

Class Name	CP467

▼ Task 1 and Task 2

Averaging Smoothing Filter

- Looking at the results it can be seen that both are not too apart from each other as the averaging is applied using a constant `mask_size = 3` differing in `mask_size` and `padding` could lead to different smoothing effects.
- Another difference, could be in the fact that OpenCV mathematical operations must have precision in them as well as they are optimized.

Gaussian Smoothing Filter

My own gaussian filter implementation

OpenCV gaussian filter implementation

- I believe there are not many differences in my implementation and OpenCV as I have differentiated the filter first as compared to multiplying the image with filter and then differentiating it as convolution is associative. As mentioned before that due to having optimized functions and using accurate mathematical operations there can be minor differences as some of the blurring/noise still remains in my output image.
- I believe in this case probably not, but the constant portion in the Gaussian equation can also make a great impact as the σ increases and the image can become more blurred.

Sobel Sharpening Filter

My own sobel filter implementation

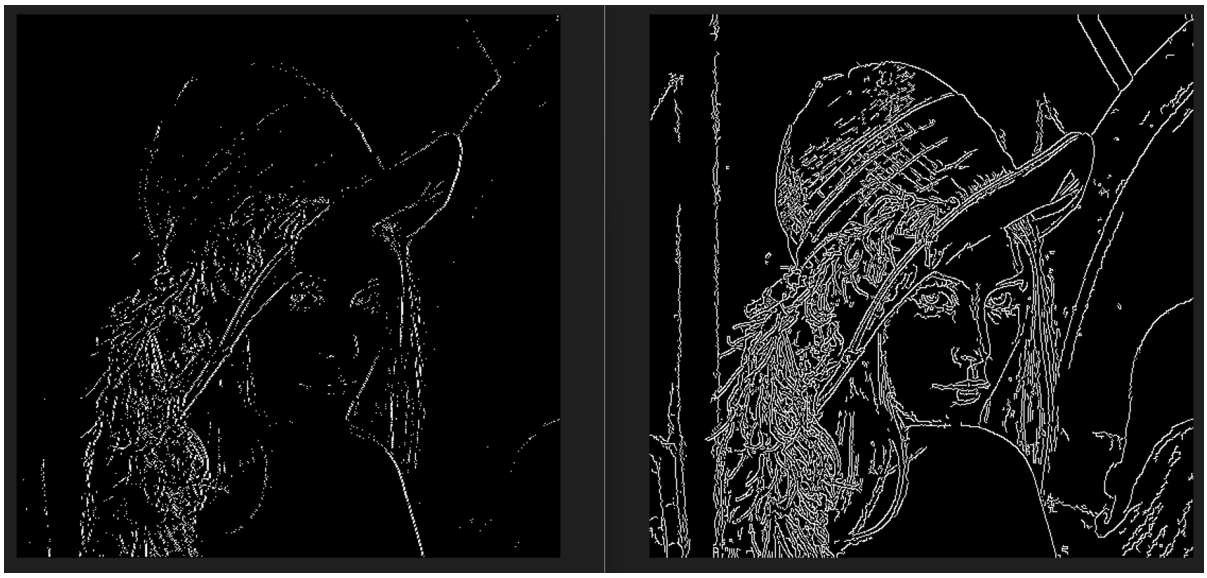
OpenCV sobel filter implementation

- The way edges usually are detected can yield different edge responses. I can see the difference in the face edges and hair edges, OpenCV is capturing way more details than mine.
- The precision of calculations may differ and convolution implementation can be differing thus yielding in different results.
- Boundary effects can also yield in different result if the edge cases aren't tested properly.

▼ Task 3

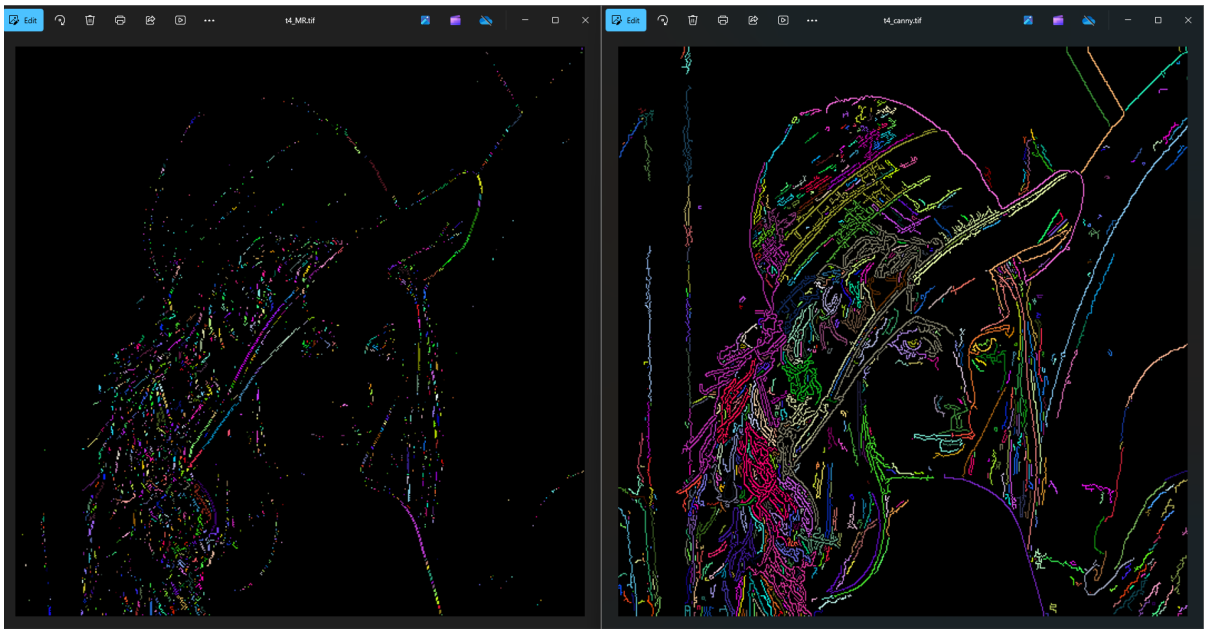
a) The Marr-Hildreth Edge Detector

- on the left is MH edge detector and on the right it is Canny edge detector



- since Canny involves multiple steps such as smoothing, finding gradients, applying non -maximum suppression and double thresholding with hysteresis it is more robust to noise and provide better edges than the MH edge detector.
- Less precise in edge localization compared to Canny that is one of the biggest difference noticed in the image.
- if the MH shows broader, more pronounced edges while the Canny shows sharper, more defined edges, it highlights the strengths of Canny in edge localization.
- I think using the double thresholding allows the Canny method to be less sensitive to noise as compared to MH.

▼ Task 4



- As clearly as it can be seen from the images that Canny edge map resulted in more connected components. Here, the technique is to maintain the image's outlines while strengthening the connections between its key components. Canny detector typically performs better at identifying continuous and well-defined edges.
- The **Canny output** is much better at capturing the overall structure and finer details, such as facial features, hair strands, and hat edges,

▼ Bonus Task

Task	From-scratch Implementation Time	OpenCV Implementation Time	Difference
Averaging Smoothing Filter	O(H x W) where H= height and W =width	O(H x W) where H= height and W =width	OpenCV's implementation with optimizations that make it significantly faster in practice compared to my custom implementation.

Gaussian Smoothing Filter	O(H x W x M^2) where H= height and W =width, M = mask size	O(H x W x M^2) where H= height and W =width, M = mask size	OpenCV's implementation with optimizations that make it significantly faster in practice compared to my custom implementation.
Sobel Sharpening Filter	O(H x W) where H= height and W =width	O(H x W) where H= height and W =width	OpenCV's faster because of optimized convolution algorithms
Marr-Hildreth Edge Detector	-	O(H x W x k^2) where H= height and W =width	
Canny Edge Detector	-	O(H x W x k^2) where H= height and W =width	
Group Adjacent Pixels	O(H x W) where H= height and W =width	-	