COMPUTER VISION

II. The Visual Machinery of the human Brain

INTRODUCTION:

Edge operators are a set of algorithms used in image processing to detect edges in images. These operators use mathematical computations to identify areas in an image where the pixel values change abruptly, which is an indication of an edge. There are several types of edge operators, each with its strengths and limitations.

Some popular edge operators include:

- Sobel operator: This operator is used to calculate the gradient of the image intensity function, and it emphasizes vertical and horizontal edges.
- Prewitt operator: Similar to the Sobel operator, the Prewitt operator is also used to calculate the gradient of the image intensity function. However, it emphasizes diagonal edges in addition to vertical and horizontal edges.
- Laplacian of Gaussian (LoG): This operator applies a Gaussian filter to an image to smooth it out, and then applies the Laplacian operator to the smoothed image to detect edges.
- Canny edge detector: This operator is a multi-stage algorithm that uses edge thinning, hysteresis thresholding, and non-maximum suppression to accurately detect edges in an image.
- Kirsch operator: This operator uses a set of eight 3x3 kernels, each corresponding to a specific gradient direction, to detect edges in an image.

PROCEDURE:

I have performed various edge detection techniques on an input image.

First, the code loads an image in grayscale using OpenCV and defines two Sobel operators for horizontal and vertical gradients. Then, it applies convolution to the input image using the Sobel operators to get the horizontal and vertical gradient values. Next, it calculates the magnitude of the gradient by taking the square root of the sum of the squares of the horizontal and vertical gradients.

Next the code defines the modified edge operator with doubled values. It then applies this operator to the image using the cv2.filter2D function, which performs a convolution with the given kernel.

I also obtained the results using the Canny and Laplacian edge operators.

To combine the edges obtained from different edge detection techniques, the code uses edge map fusion by taking the maximum value of the edges obtained from each technique. Finally, the code displays the original image and the images obtained from each edge detection technique using matplotlib.

Kirsch Operators:

In the next part of the code I defined **Kirsch operators** for 8 different directions and I applied them to the image which was obtained by combining the different edge operators that is sobel, laplacian, canny and My_operator to extract edges using convolution. Kirsch operators are a type of image processing operator that use a 3x3 matrix to detect edges in an image.

The Kirsch operators are defined as 3x3 matrices, each representing a different direction in which edges are to be detected. In this code, 8 different Kirsch operators are defined to detect edges in 8 different directions. The operators are applied to the input image using the **cv2.filter2D()** function, which performs convolution of the image with the Kirsch operator.

The result of applying each Kirsch operator to the image is stored in an array, and the final Kirsch edge image is obtained by taking the maximum value of the edges detected by each operator at each pixel location. This is done to combine the information from all the different directions in which edges can occur in the image.

The final Kirsch edge image is displayed using **matplotlib**. This image shows the edges detected in the input image using the Kirsch operators.

Prewitt Compass Operator:

In the final part of the code I performed the edge detection using Prewitt compass operators, which are a set of 8 gradient filters that detect edges in different directions. The Prewitt compass operators are defined as numpy arrays, with each array representing a filter for a specific direction.

First, the code defines the Prewitt compass operators and stores them in a list. Then, it applies convolution to the input image using the **Kirsch operators** to obtain the initial edge detection results.

Next, the **code applies each of the Prewitt compass operators to the Kirsch edges** to obtain the gradient values in each of the 8 directions. The magnitude of the gradient is then calculated using the Pythagorean theorem, and the resulting magnitudes are normalized to the range between 0 and 255.

Finally, the normalized magnitudes are displayed as a grayscale image using matplotlib.

Overall, the code uses a combination of Kirsch and Prewitt operators to detect edges in an image with different orientations and gradients, and produces a final image that highlights the detected edges.

RESULTS:

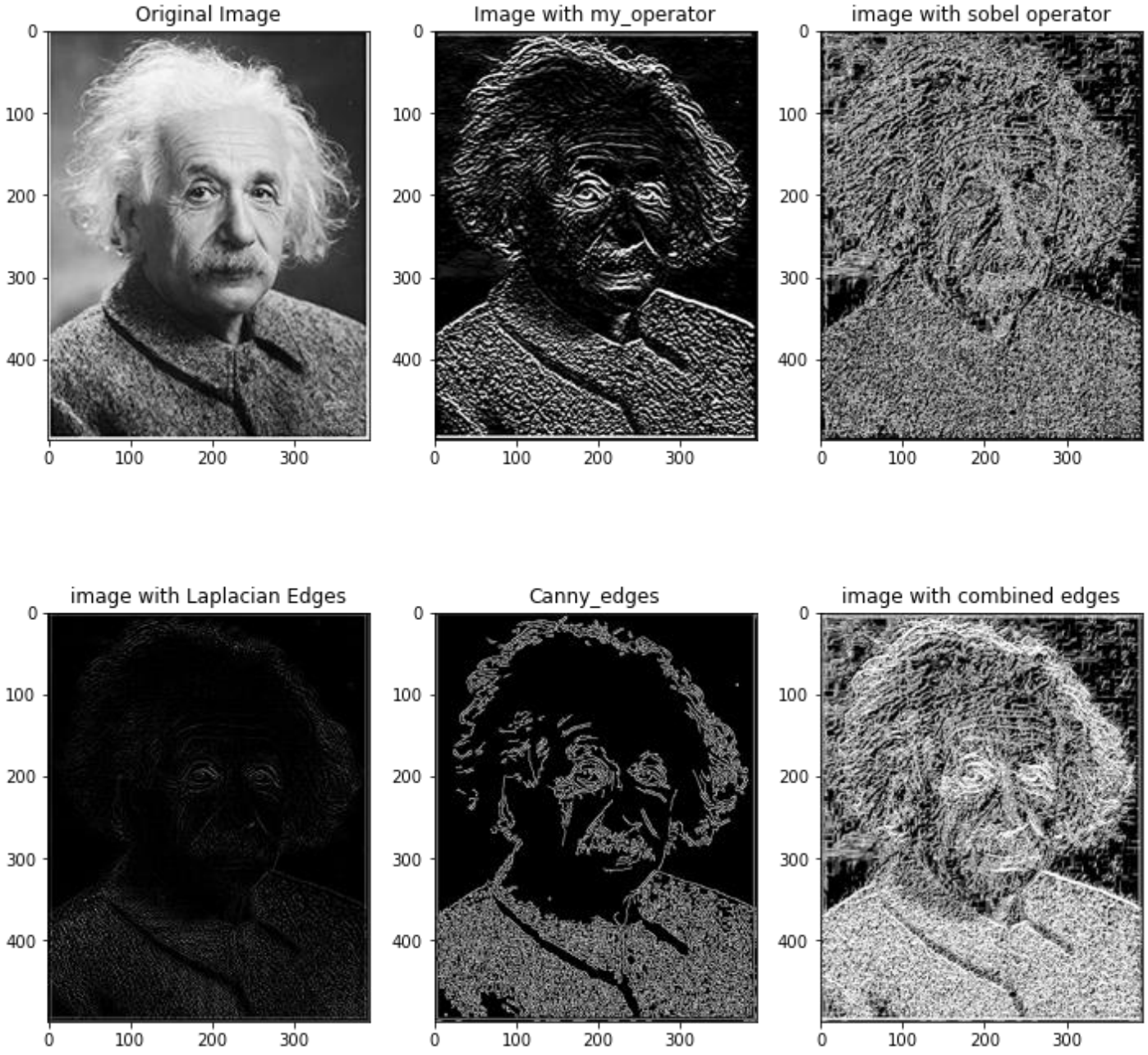


Image with combined edges in the above image is given as input image to the Kirsch operator

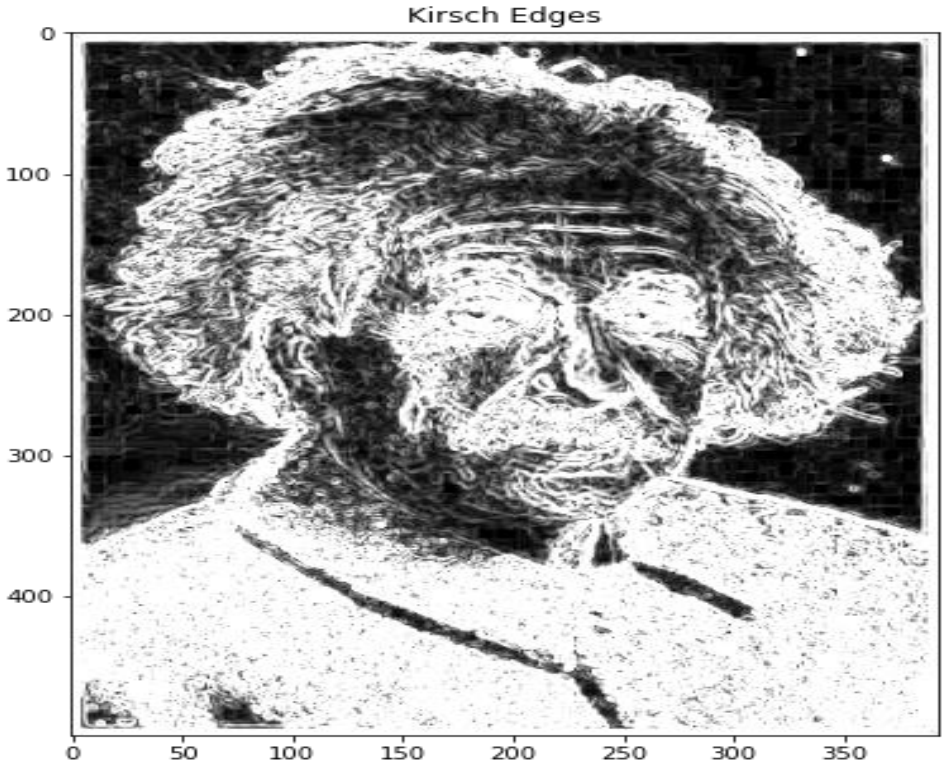
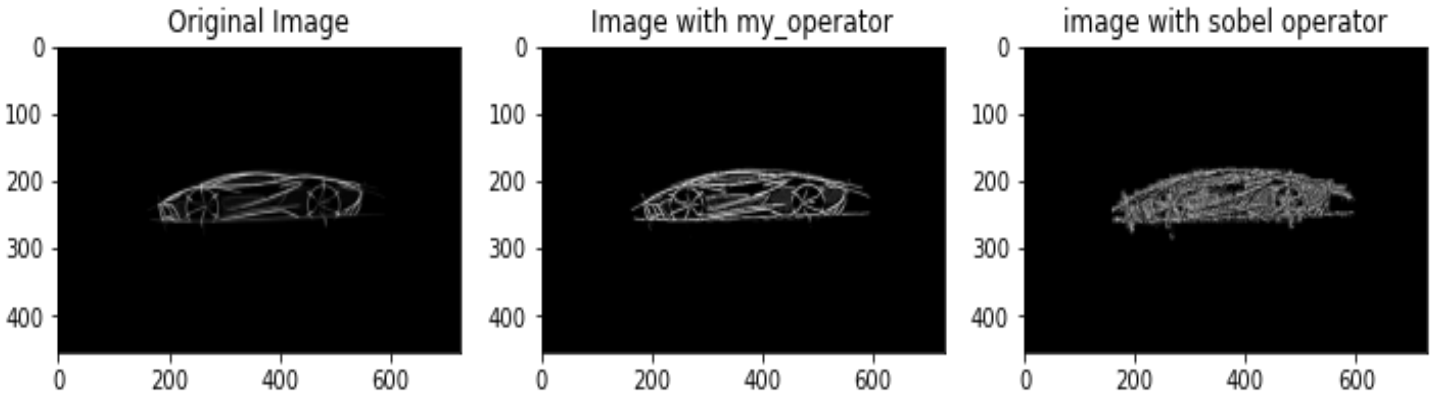


Image obtained with kirsch operator is given as input image to the Prewitt Compass operator



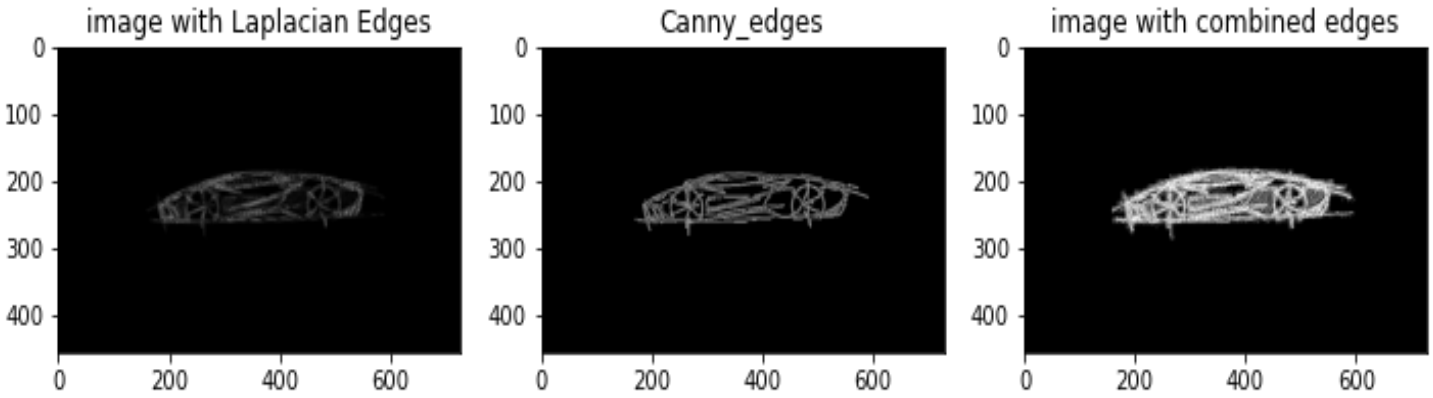


Image with kirsch operator contrasting with the above results of all operators

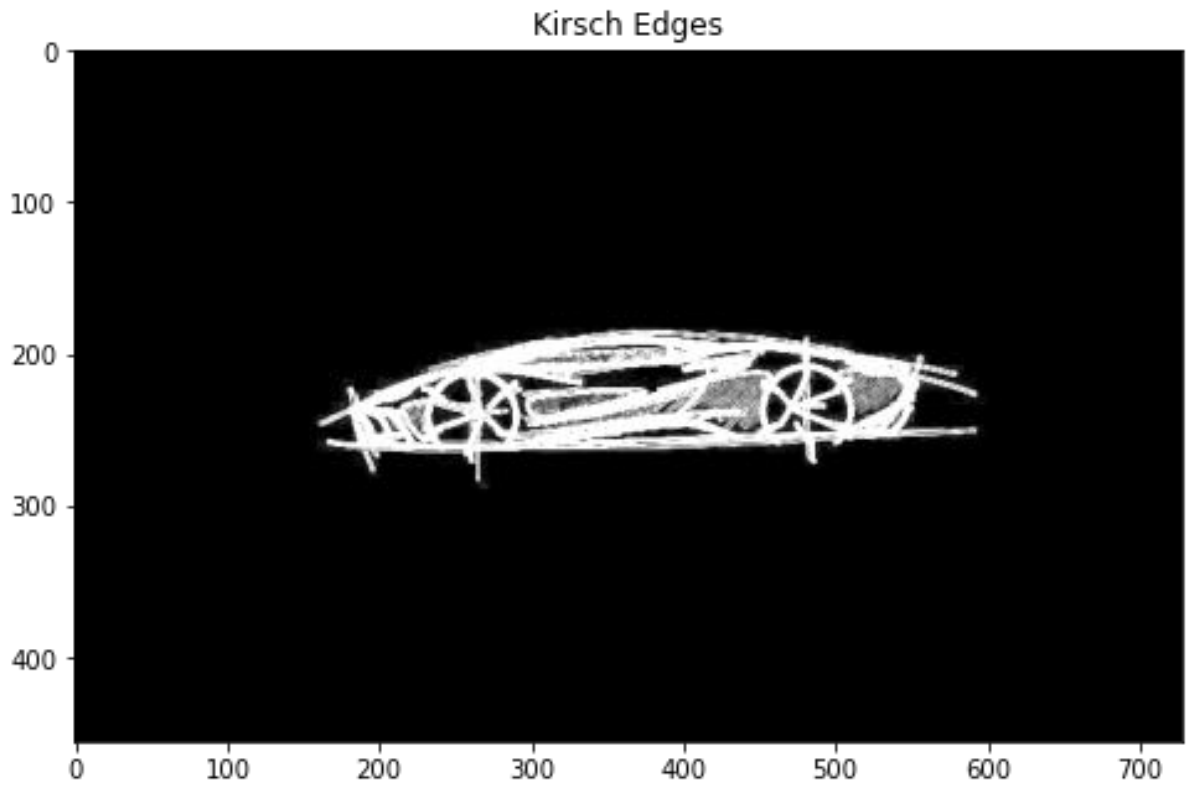
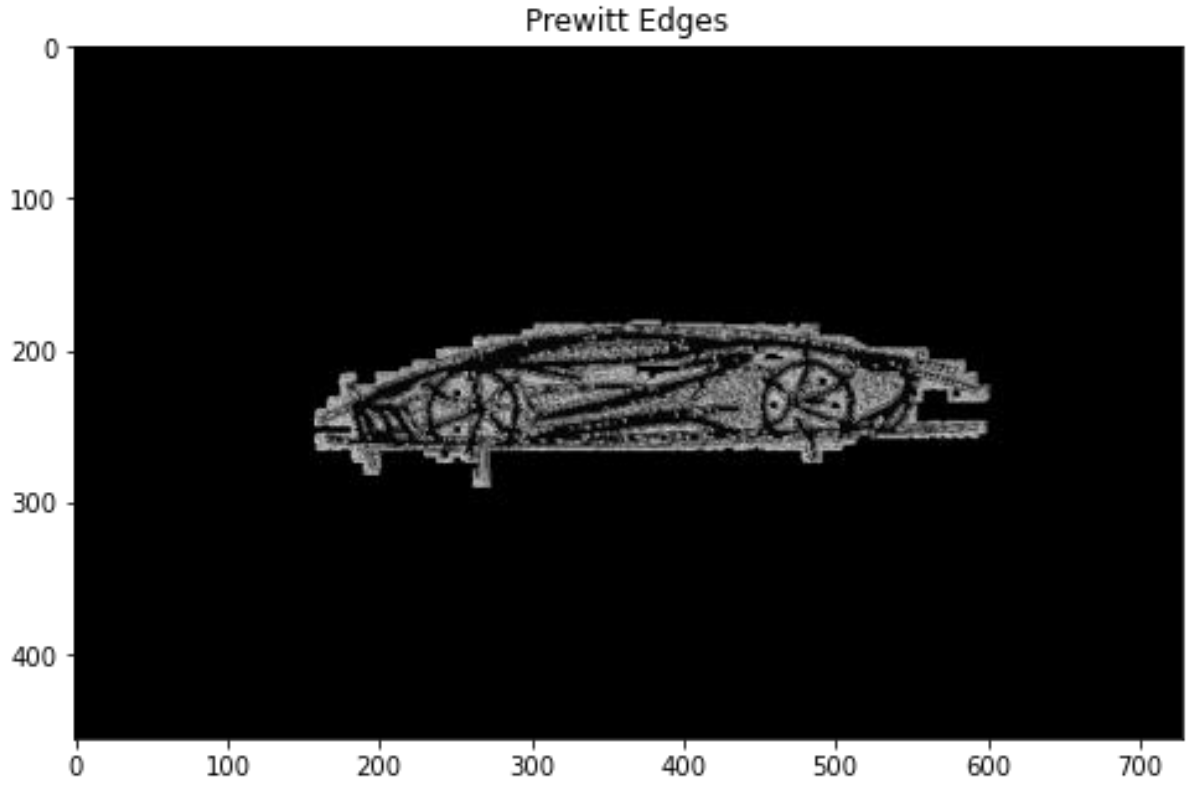
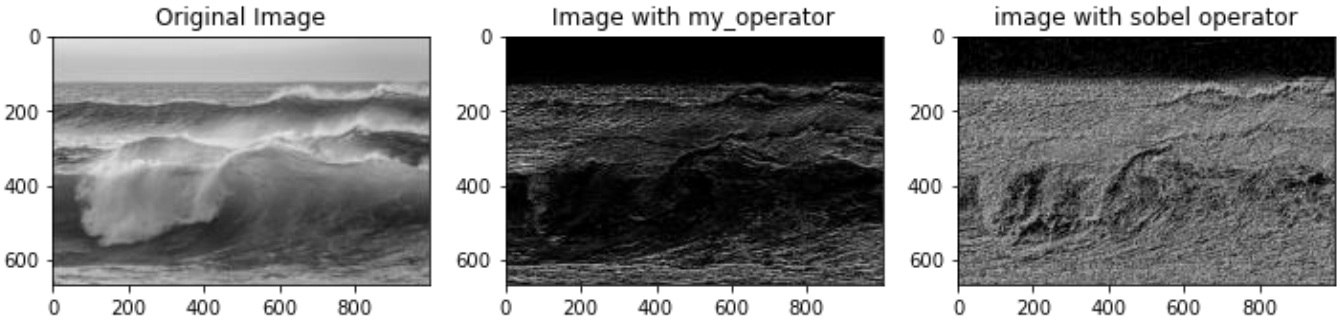
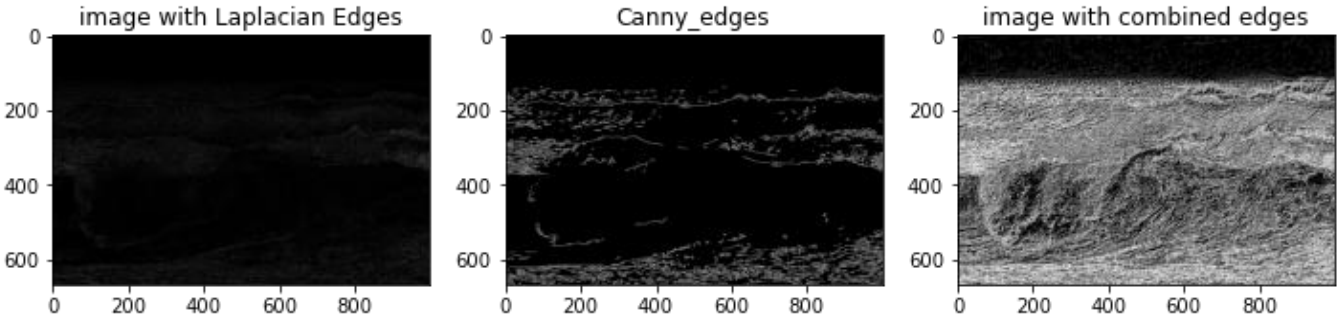


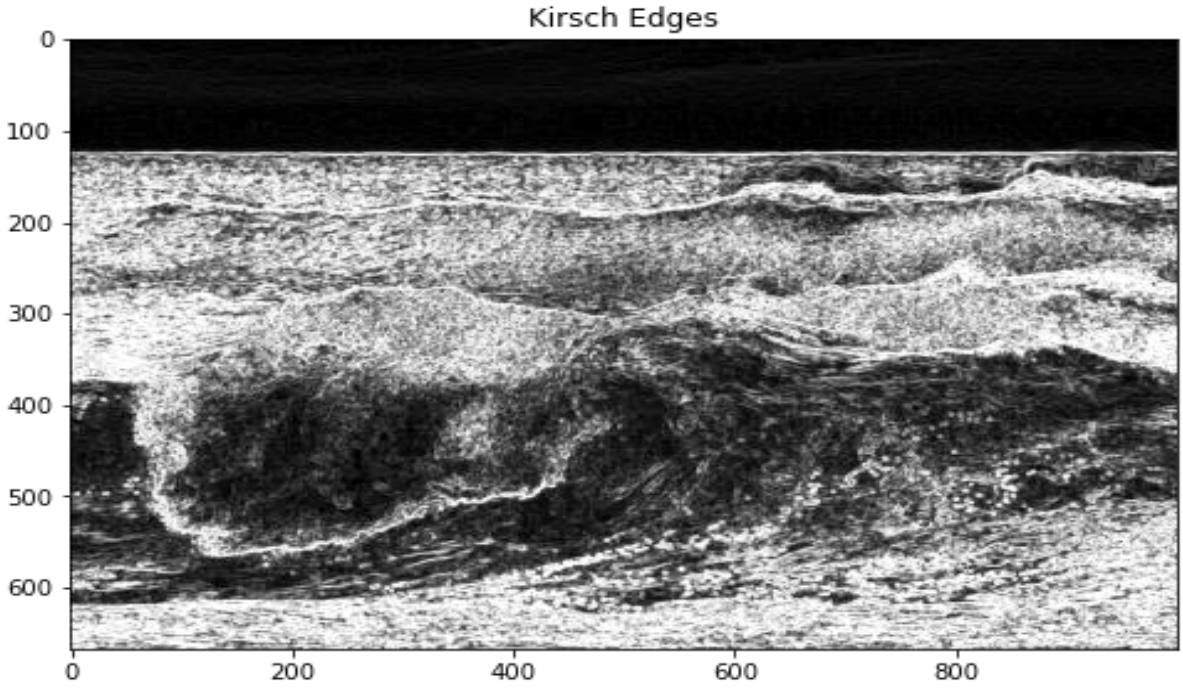
Image with prewitt compass operator contrasting with result obtained from the kirsh operator.



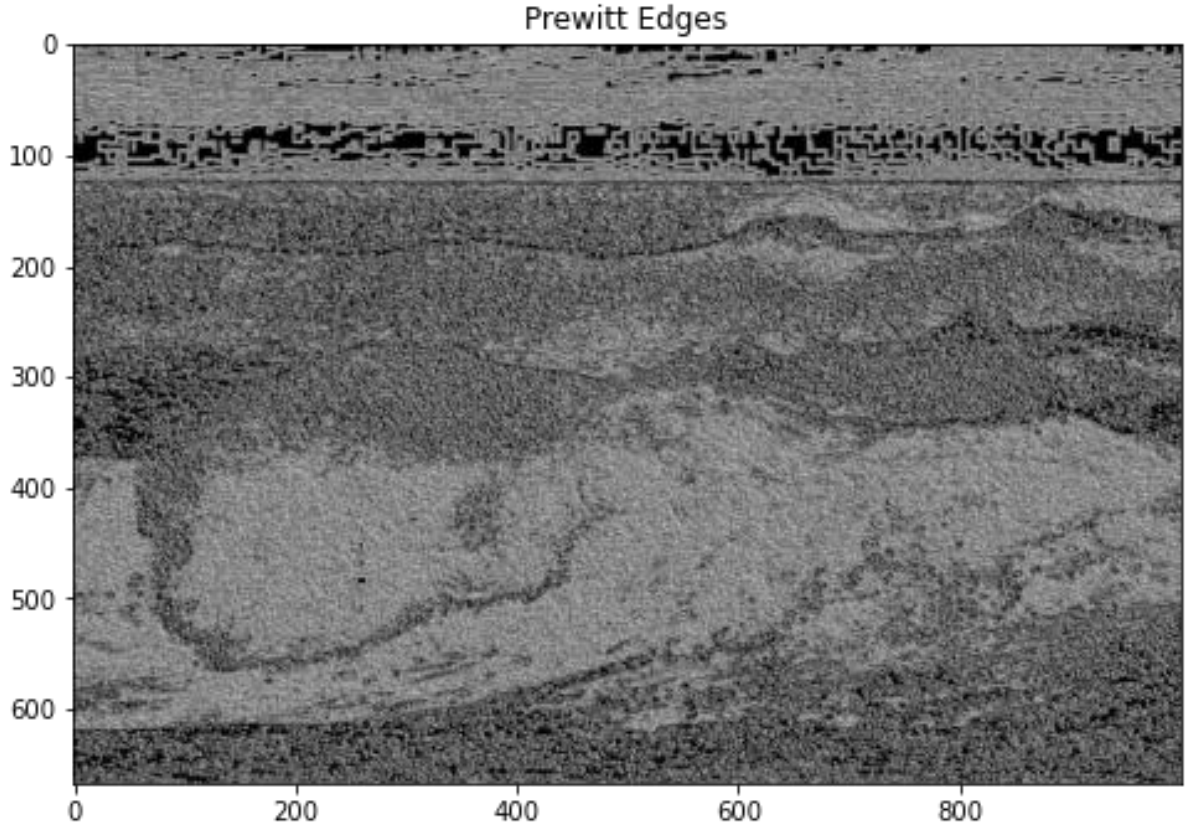




This below image was obtained with kirsch operator contrasting all the operators from the above



This below image was obtained with prewitt compass operator contrasting the above image from the kirsch operator.



CONCLUSIONS:

I have applied different edge detection operators on an image and observed the results. Canny operator gave us the most accurate and detailed edges with low noise. Sobel and Laplacian operators also produced good edges, but with slightly more noise than Canny. Prewitt operators produced edges with higher noise and some missing edges compared to other operators. Kirsch operators gave us edges that are similar to Sobel but with more emphasis on diagonal edges.

Overall, the choice of edge detection operator depends on the specific application and the desired level of accuracy and noise tolerance.