Automatic Number Plate Recognition (Python)

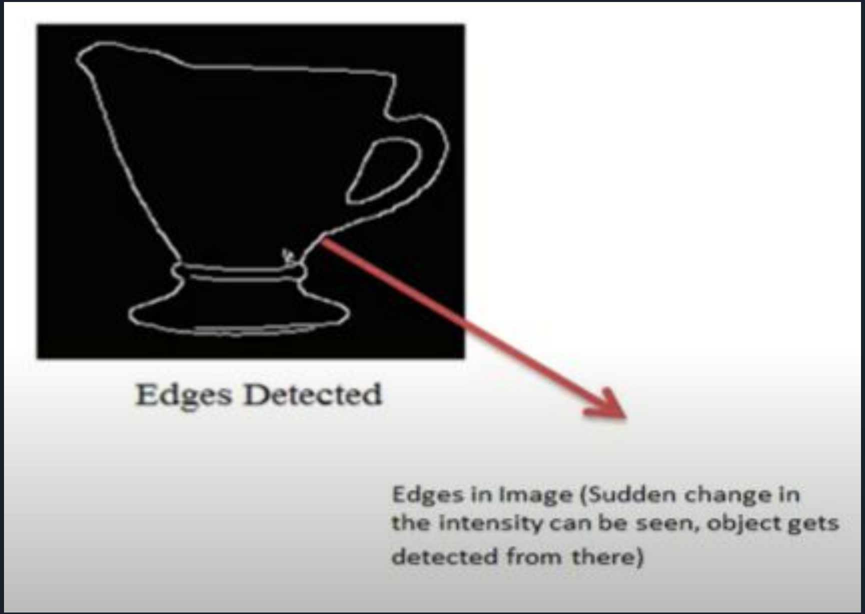
Deepak Goyal

Edge Detection:

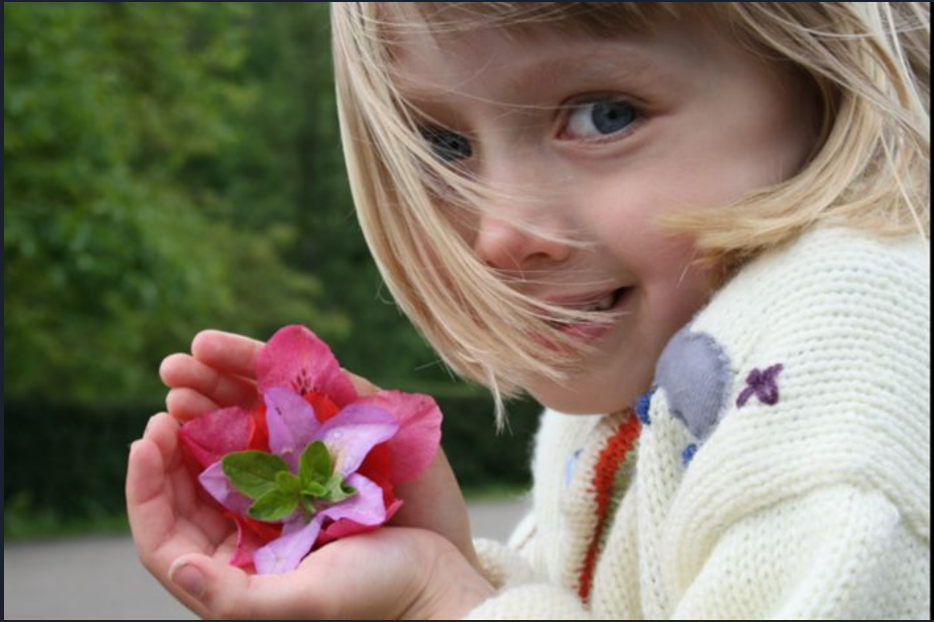
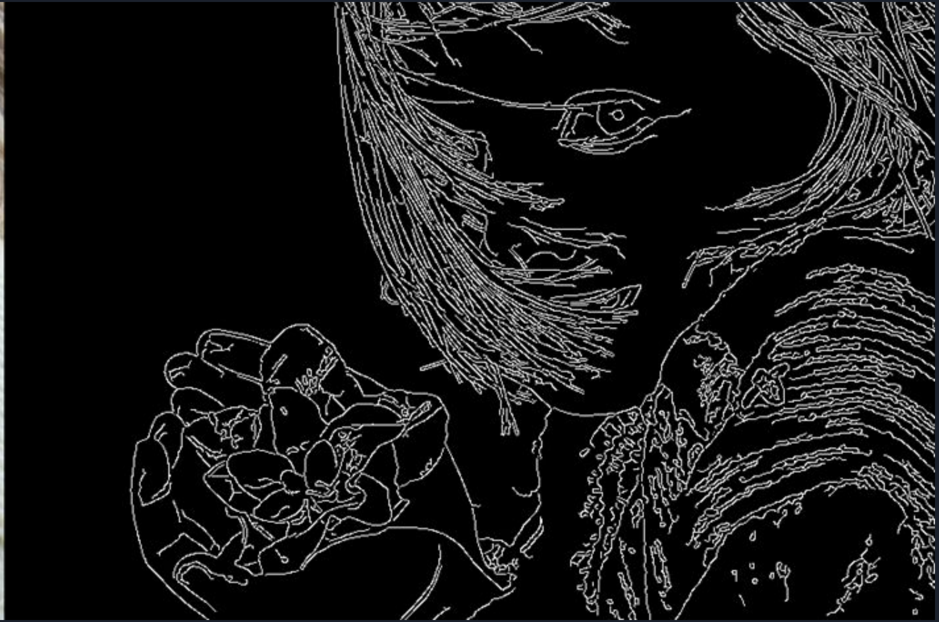
Edges are defined as sudden and significant changes in the intensity of an image.

If we are able to detect the edges we can find out the exact layout of the object.

The points where the image brightness varies sharply are called the edges of the image.

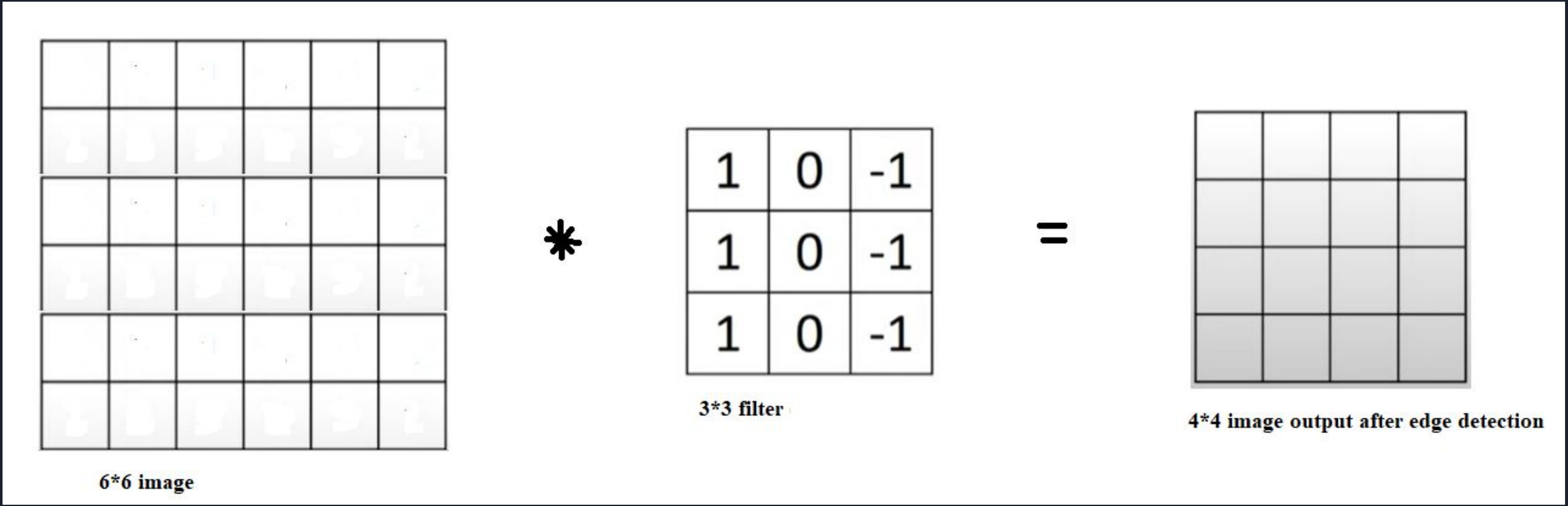

Edges Detected
Edges in Image (Sudden change in the intensity can be seen, object gets detected from there)

Resultant Image :

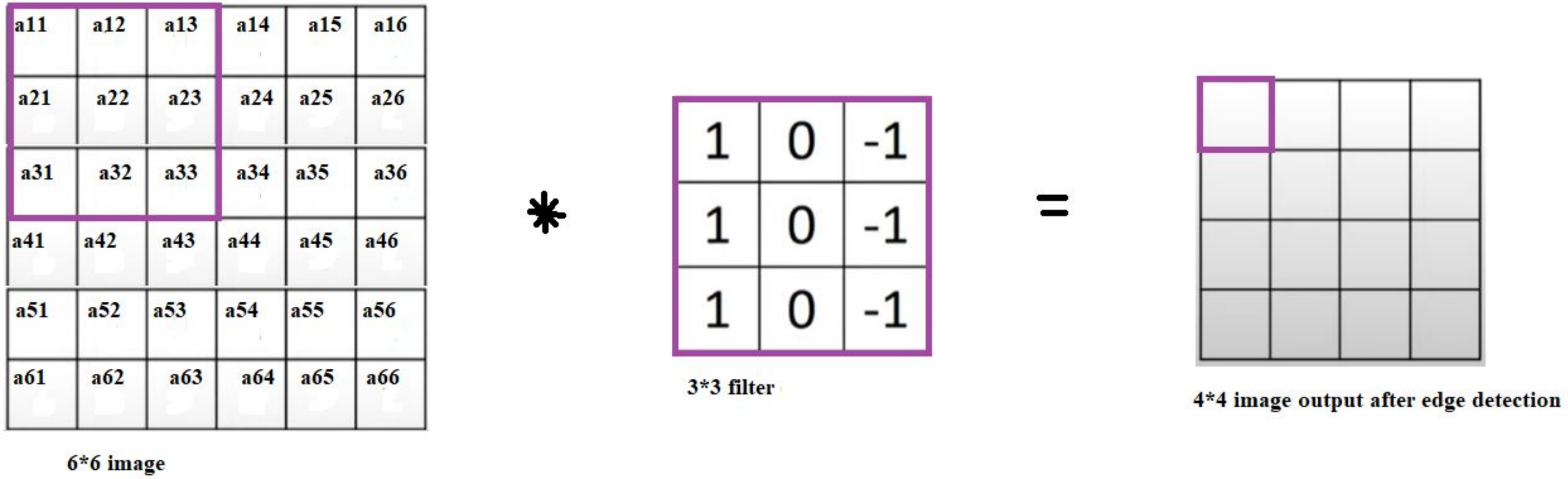



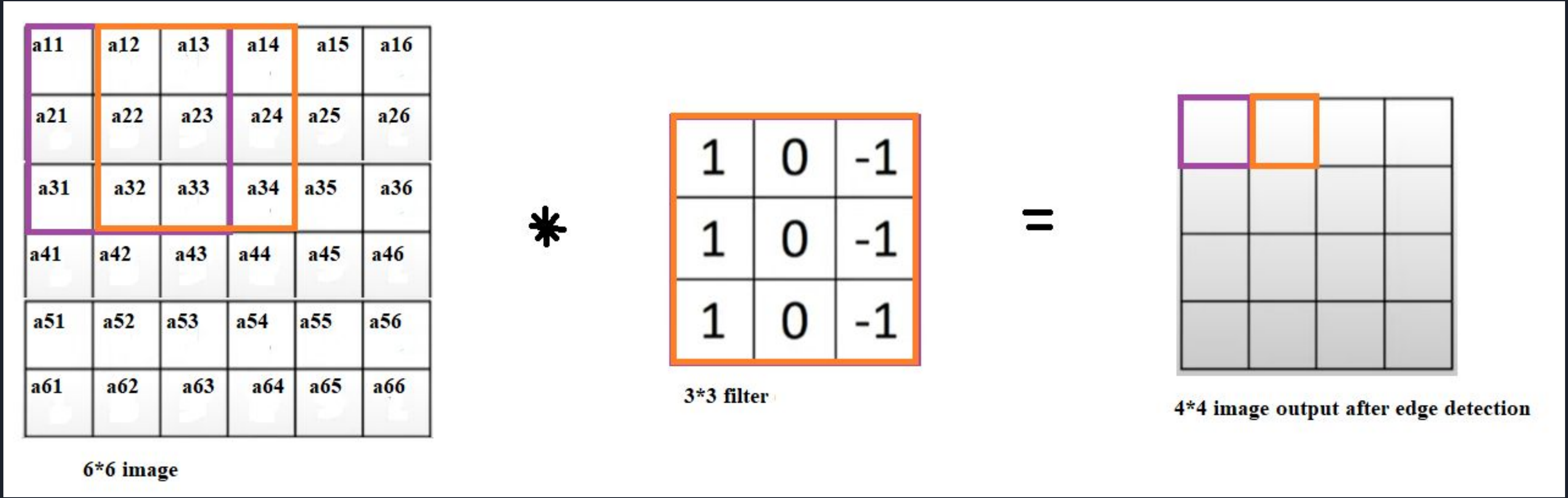
Convolution Operator:

- A convolution is the simple application of a filter to an input that results in an activation
- When we process very high-resolution digital images, convolution techniques come to our rescue. Let us understand the convolution operation (represented in the below image using *) using an example-


6*6 image
*
1 0 -1
1 0 -1
1 0 -1
3*3 filter
=
4*4 image output after edge detection

For this example, we are using 3*3 Prewitt filter as shown in the above image. As shown below, when we apply the filter to perform edge detection on the given 6*6 image (we have highlighted it in purple for our understanding) the output image will contain ((a11*1) + (a12*0) + (a13*(-1))+(a21*1)+(a22*0)+(a23*(-1))+(a31*1)+(a32*0)+(a33*(-1))) in the purple square. We repeat the convolutions horizontally and then vertically to obtain the output image.




a11 a12 a13 a14 a15 a16
a21 a22 a23 a24 a25 a26
a31 a32 a33 a34 a35 a36
a41 a42 a43 a44 a45 a46
a51 a52 a53 a54 a55 a56
a61 a62 a63 a64 a65 a66
6*6 image
*
1 0 -1
1 0 -1
1 0 -1
3*3 filter
=
4*4 image output after edge detection

Methods of Edge Detection:

There are various methods in edge detection, and the following are some of the most commonly used methods-

- Prewitt edge detection
- Sobel edge detection
- Canny edge detection

Prewitt Edge Detection:

This method is a commonly used edge detector mostly to detect the horizontal and vertical edges in images. The following are the Prewitt edge detection filters-

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt filter for vetical edge detection **Prewitt filter for horizontal edge detection**

Sobel Edge Detection:

Sobel Edge Detection: This uses a filter that gives more emphasis to the centre of the filter. It is one of the most commonly used edge detectors and helps reduce noise and provides differentiating, giving edge response simultaneously. The following are the filters used in this method-

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Sobel filter for vertical edge detection

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel filter for horizontal edge detection

Canny Edge Detection:

This is the most commonly used highly effective and complex compared to many other methods. It is a multi-stage algorithm used to detect/identify a wide range of edges. The following are the various stages of the Canny edge detection algorithm-

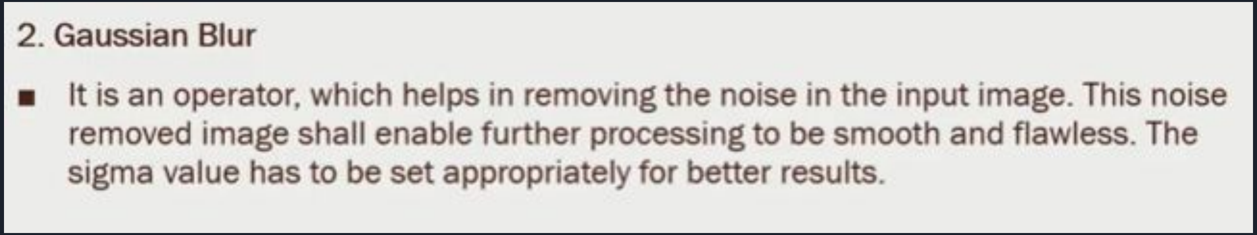
Steps for Canny Edge Detection:

1. Convert the image to grayscale
2. Reduce noise – as the edge detection that using derivatives is sensitive to noise, we reduce it.
3. Calculate the gradient – helps identify the edge intensity and direction.
4. Non-maximum suppression – to thin the edges of the image.
5. Double threshold – to identify the strong, weak and irrelevant pixels in the images.
6. Hysteresis edge tracking – helps convert the weak pixels into strong ones only if they have a strong pixel around them.

Step 1: Convert Image to Grayscale version. The below method can be used for this:

```
cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

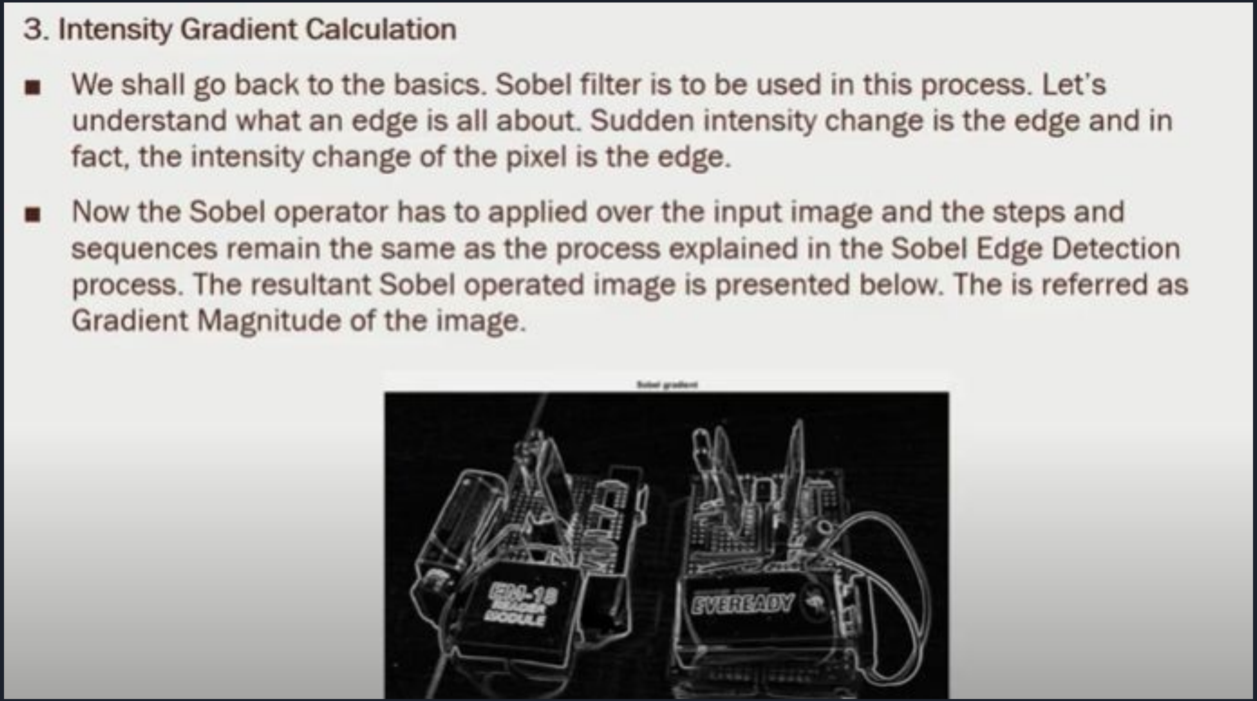
Step 2: Apply Gaussian filter to smooth the image in order to remove the noise.


2. Gaussian Blur

- It is an operator, which helps in removing the noise in the input image. This noise removed image shall enable further processing to be smooth and flawless. The sigma value has to be set appropriately for better results.

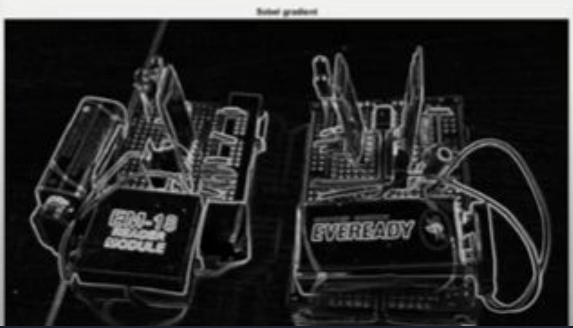
Since all edge detection results are easily affected by the noise in the image, it is essential to filter out the noise to prevent false detection caused by it. To smooth the image, a Gaussian filter kernel is convolved with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector.

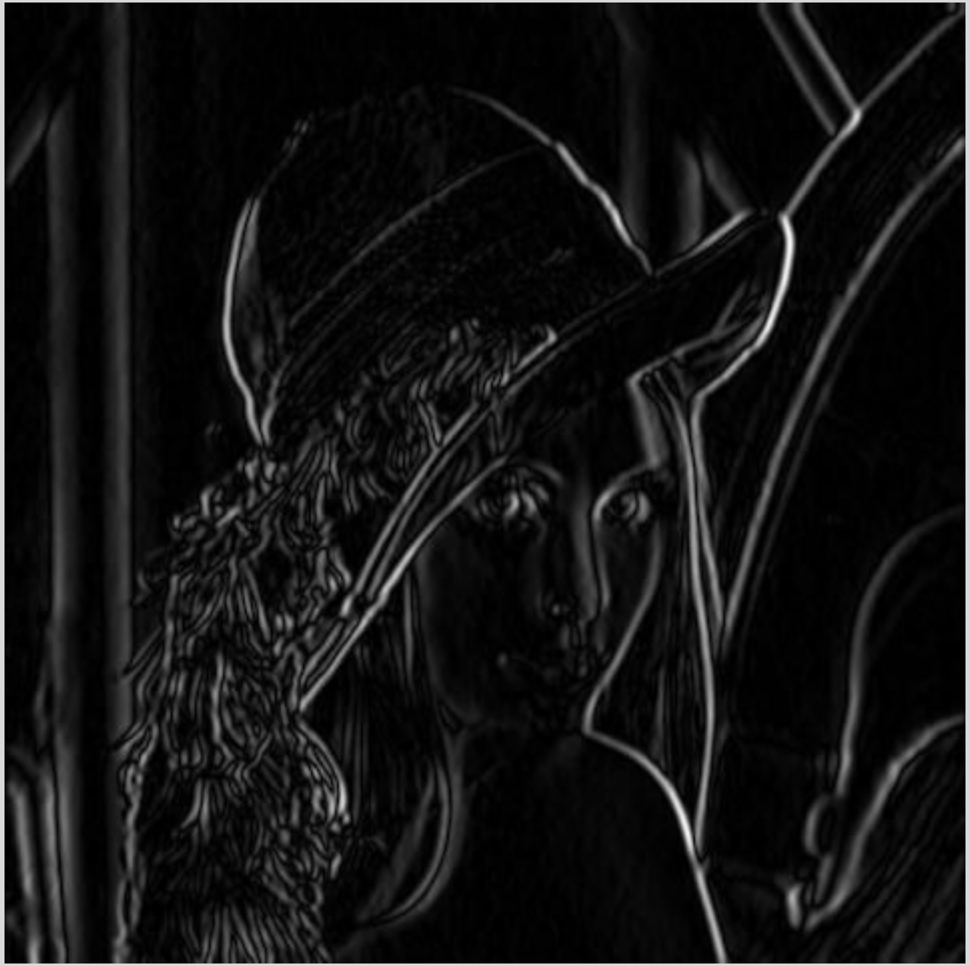
Step 3: Intensity Gradient Calculation



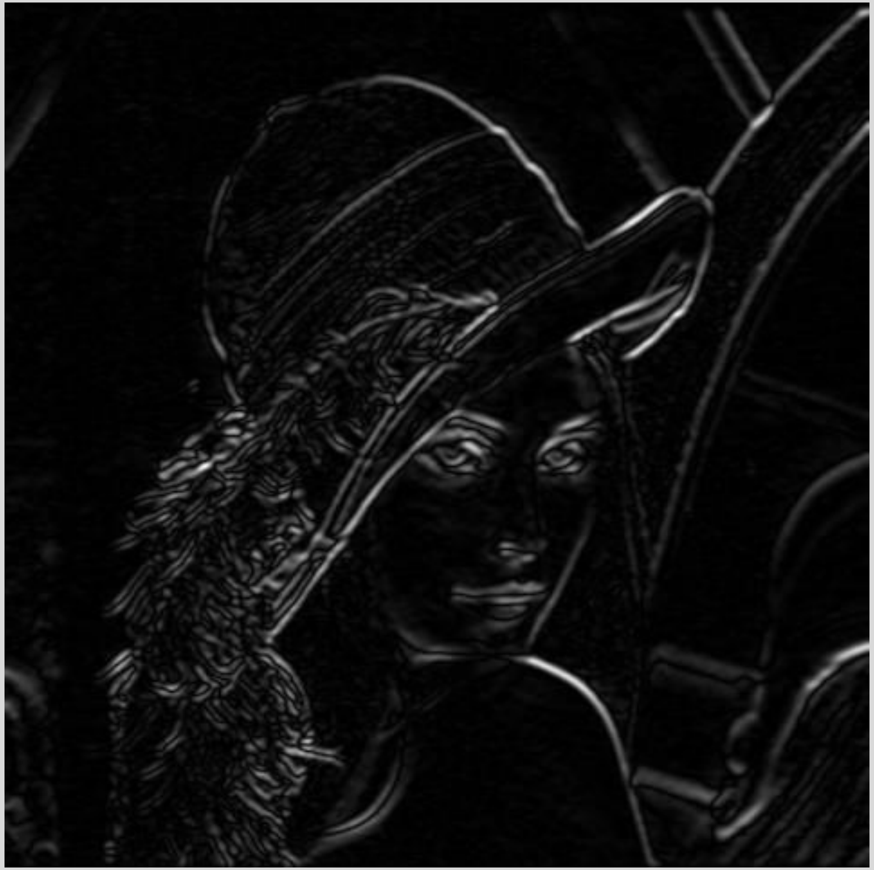
3. Intensity Gradient Calculation

- We shall go back to the basics. Sobel filter is to be used in this process. Let's understand what an edge is all about. Sudden intensity change is the edge and in fact, the intensity change of the pixel is the edge.
- Now the Sobel operator has to applied over the input image and the steps and sequences remain the same as the process explained in the Sobel Edge Detection process. The resultant Sobel operated image is presented below. The is referred as Gradient Magnitude of the image.





Gradient in X direction

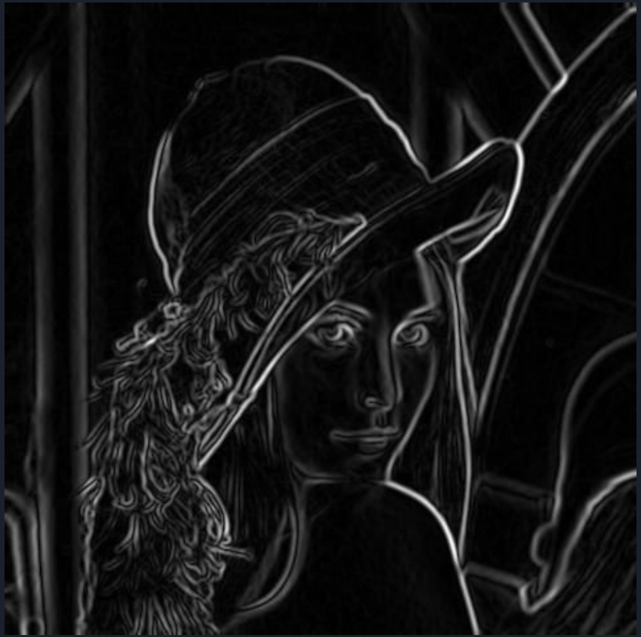


Gradient in Y direction

Calculate the Magnitude and Angle of Gradient

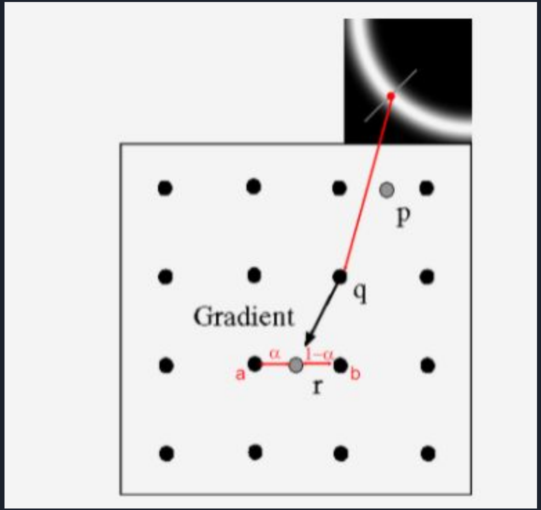
$$|G| = \sqrt{{G_x}^2 + {G_y}^2}$$

$$\angle G = arctan(G_y/G_x)$$



Step 4 : Non Maximum Supression

The image magnitude produced results in thick edges. Ideally, the final image should have thin edges. Thus, we must perform non maximum suppression to thin out the edges.



- Non maximum suppression works by finding the pixel with the maximum value in an edge.
- In the above image, it occurs when pixel q has an intensity that is larger than both p and r where pixels p and r are the pixels in the gradient direction of q.
- If this condition is true, then we keep the pixel, otherwise we set the pixel to zero (make it a black pixel).

Step 5: Double Thresholding

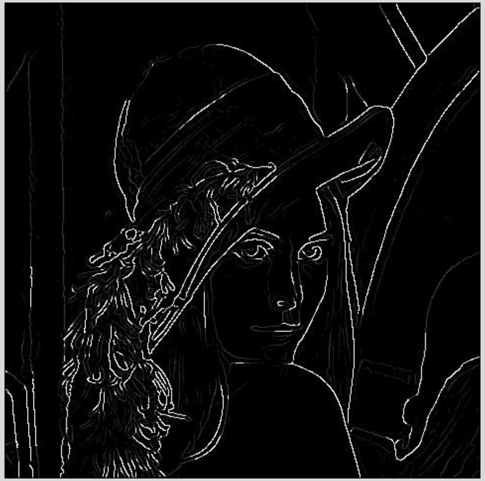
We notice that the result from non maximum suppression is not perfect, some edges may not actually be edges and there is some noise in the image. Double thresholding takes care of this. It sets two thresholds, a high and a low threshold.

Let’s take an example:

High Threshold value=160
Low Threshold value =120

- All the pixels greater than 160 will be strong edges.
- All the pixels smaller than 120 will not be considered as edges so that pixel value will be set to 0.
- And All the pixel values between 120 and 160 will be considered as weak edges

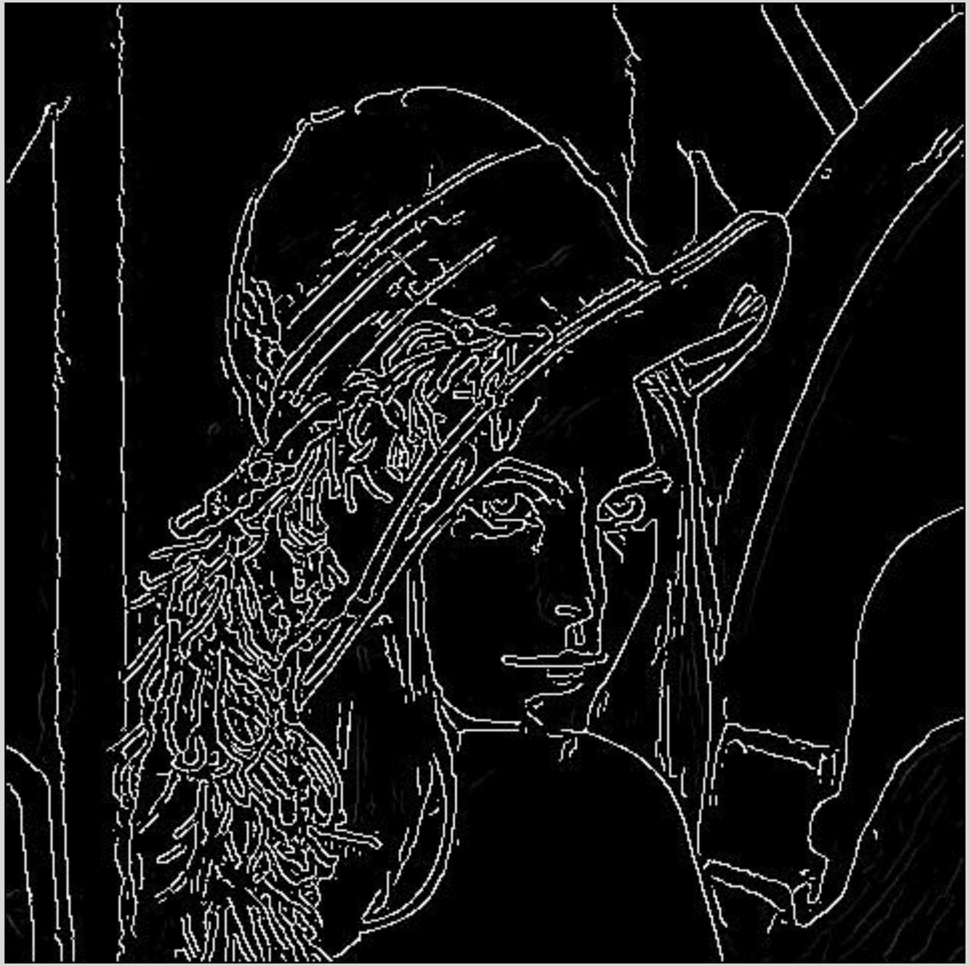
This technique is called double thresholding



Step 6: Edge Tracking by Hysteresis

- Now that we have determined what the strong edges and weak edges are, we need to determine which weak edges are actual edges.
- Weak edges that are connected to strong edges will be actual/real edges.
- Weak edges that are not connected to strong edges will be removed.

Final Output Image



End