Canny Edge Detection.

The Image is **smoothened** first.

Edges correspond to the change in pixels intensity.

The derivatives Ix and Iy are calculated by convolving Kx and Ky filters to the image.

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Sobel filters for both direction (horizontal and vertical)

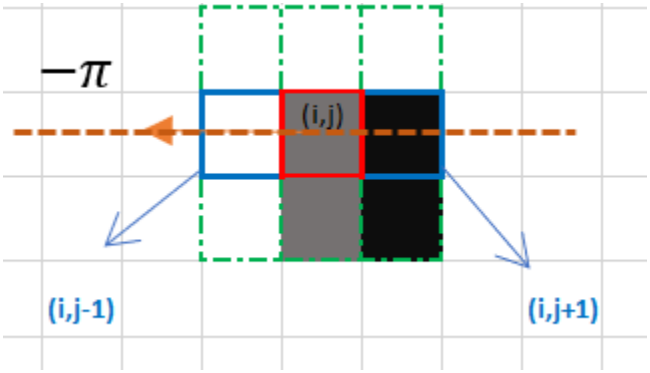
The gradient and the slope of the gradient is calculated.

$$|G| = \sqrt{I_x^2 + I_y^2},$$
$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

Gradient is calculated by using np.hypot method

Theta is calculated by using the np.arctan2 method.

Now , there are some edges which are a bit thicker so we have to apply **non maximum suppression**.



Here the orange line is the edge direction and Redbox is selected pixel.

So the highest intensity from its direction is chosen which is the left pixel as while has higher intensity then black.

Double Threshold

Important for segregating the pixels in to strong, weak and non-relevant.

High threshold- for identifying the strong pixels(which have higher intensity)

Low threshold- for identifying the strong pixels(which have higher intensity)

The pixels which lie between these two thresholds are classified using **Hysteresis.**

Hysteresis

Ever if one pixel is found around the disputed pixel then it is converted to string pixel.