**Filters and edge detection**

Tarek Mohamed

Alex Eagles

**Sobel Filter**

A Sobel filter, in the context of Computer Science, is a type of filter that is used for edge detection in images. It is defined by a specific matrix or window size, such as 3x3 or 5x5, and is applied to the image to highlight the edges by calculating the gradient intensity.



Sobel filtering involves applying two 3 x 3 (it can be 5 x 5) convolutional kernels (also called filters) to an image. The kernels are usually called *Gx*and *Gy*, and they are shown in the following figure. These two kernels detect the edges in the image in the horizontal and vertical directions. They are applied separately and then combined to produce a pixel value in the output image at each position in the input image.

The output value is approximated by:

G = |Gx| + |Gy|

A screenshot of a graph

Description automatically generated

**In python**

cv2.Sobel(original\_image,ddepth,xorder,yorder,kernelsize)

A computer screen shot of a program code

Description automatically generated

A collage of images of a building

Description automatically generated**Laplacian Filter**

The Laplacian filter is another edge detection algorithm used in image processing. Unlike the Sobel filter, which calculates the gradient, the Laplacian filter computes the second-order derivative of the image. This method highlights regions of rapid intensity change, which typically correspond to edges and other significant features in the image.



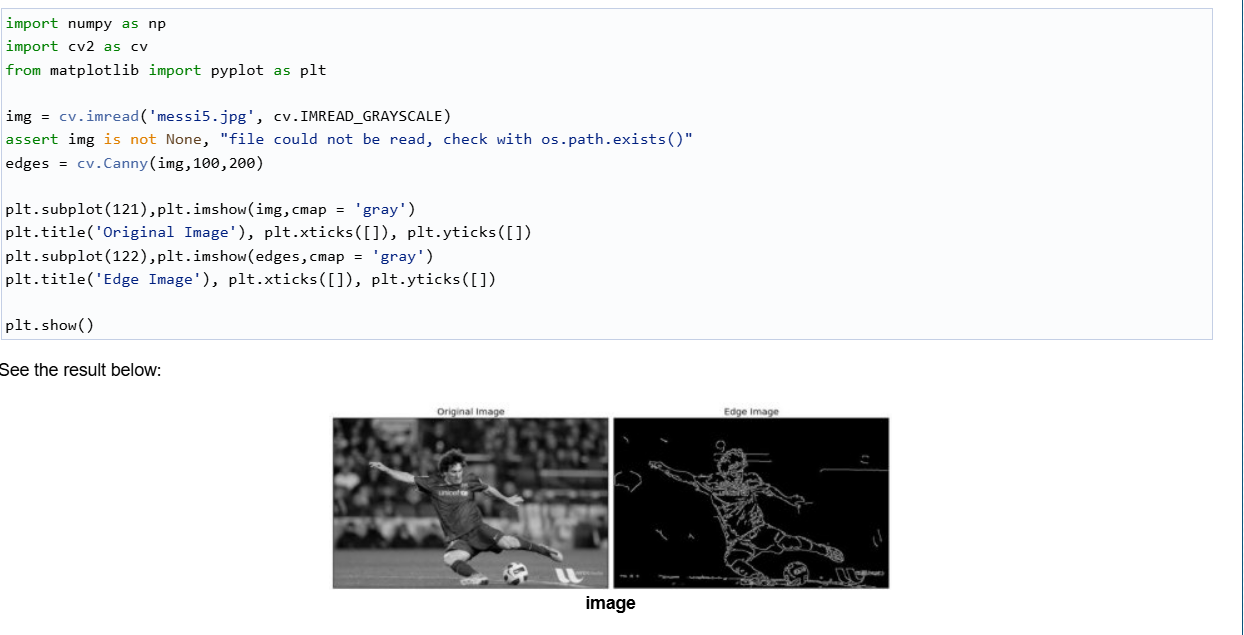
A close up of a person's eye

Description automatically generated

**Canny Edge Detector**

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various [computer vision](https://en.wikipedia.org/wiki/Computer_vision) systems. Canny has found that the requirements for the application of [edge detection](https://en.wikipedia.org/wiki/Edge_detection) on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.



*cv2.Canny(image, lower-threshold, upper-threshold, aperture-size, L2Gradient)*

**Contours in Image processing**

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.

Find contours form binary image

A close-up of a white background

Description automatically generated

Drawing contours

A close-up of a white background

Description automatically generated

Approximation

cv.CHAIN\_APPROX\_NONE doesn't approximate the countours it views all the possible countours

cv.CHAIN\_APPROX\_SIMPLE approximate the coutours to the smallest value as possible