Sobel Filter

Sobel filter or Sobel Derivatives is an edge detection method in OpenCV that uses first derivative to detect the edge. It uses kernels to detect horizontal changes and vertical changes in image then combines both results to detect the final filter applied to the image. We should first remove the noise from image.

The following line contains the code:

Sobel(src_gray, grad_x, ddepth, 1, 0, ksize, scale, delta, BORDER_DEFAULT)

- src_gray: the input image.
- grad_x (also applied to grad_y): The output image.
- ddepth: The depth of the output image.
- x_order: The order of the derivative in x direction.
- y_order: The order of the derivative in y direction.
- scale, delta and BORDER_DEFAULT: We use default values.

Laplacian Filter

Laplacian filter or Laplacian Operator is an edge detection method in OpenCV that uses second derivative to detect edges. It combines the second derivative on the x-axis and on the y-axis. We should first remove the noise from image.

The following line contains the code:

dst = cv.Laplacian(src_gray, ddepth, ksize=kernel_size)

- src_gray: The input image.
- ddepth: Depth of the destination image.
- kernel_size: The kernel size of the Sobel operator to be applied internally.

Canny Edge Detector

Canny edge detector is a edge detection algorithm that goes through some steps in order to detect the edges of image.

1. Noise reduction: any noise should be removed.

2. Finding Intensity Gradient of the Image: using Sobel kernel, we get the first derivative in both horizontal and vertical directions then get its direction
3. Non-maximum Suppression: here we remove any unwanted pixels by comparing it to its neighborhood.
4. Hysteresis Thresholding: here we check each edge to ensure if it’s a real edge or not. We use 2 threshold values: min and max. If the edge is above the maximum, it is considered an edge. If the edge is lower than the minimum, it is considered non-edge. Any value in between is considered edge or non-edge based on its connectivity. If it’s connected to an edge, it’s considered an edge. If it’s not connected to an edge, its considered non-edge.

The following line contains the code:

```
edges = cv.Canny(img, minVal, maxVal)
```

- img: source image.
- minVal: minimum threshold.
- maxVal: maximum threshold.

Contours in Image processing

A contour is 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 a better accuracy, we should use binary images.

```
im2, contours, hierarchy = cv.findContours(thresh, retrieve_mode, contour_approximation)
```

- img: source image.
- retrieval_mode: the type of edges to return.
- contour_approximation: approximates the edges.

And it outputs a modified image, the contours and hierarchy. contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object.