

Assignment #3: OpenCV Tutorials

Hanyang University

Sooyoung Kim [REDACTED], Gaon Choi [REDACTED]

Introduction

OpenCV(Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, etc.

OpenCV Tutorial LINK: [OpenCV tutorial link](#)

GITHUB LINK: <https://github.com/Gaon-Choi/PBL4009/tree/main/lab4%20-%20opencv%20tutorial>

Experiment

1. Image Thresholding

Image Thresholding

Original Image



Threshold Image



`cv2.threshold(image, threshold_value, max_value, flag)`

Explanation:

If a pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value.

2. Canny Edge Detection

Original Image



Edge Image



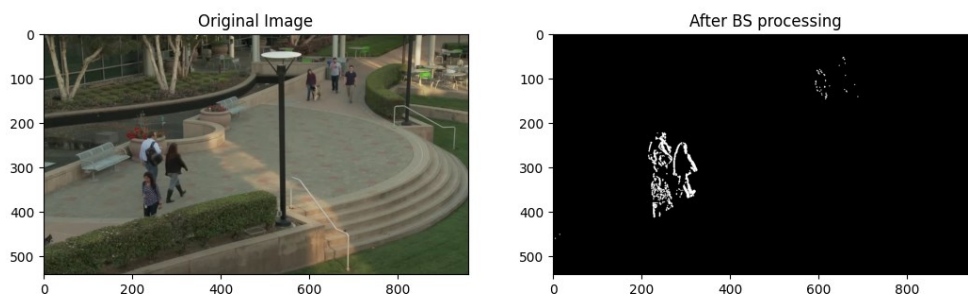
`cv2.Canny(gray_img, thresold1, threshold2)`

If a pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value.

Explanation:

- 1) Apply Gaussian filter to smoothen the image in order to remove the noise
- 2) Find the intensity gradients of the image
- 3) Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
- 4) Apply double threshold to determine potential edges
- 5) Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges

3. Background Subtraction



`cv2.BackgroundSubtractorMOG2(src, op, kernel)`

Explanation:

Background Subtraction(BS) is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. In the case of `cv2.BackgroundSubtractorMOG2`, it is a Gaussian Mixture-based Background/Foreground segmentation algorithm.

4. Histogram Equalization

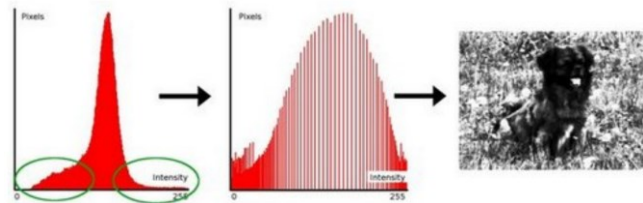
Histogram Equalization



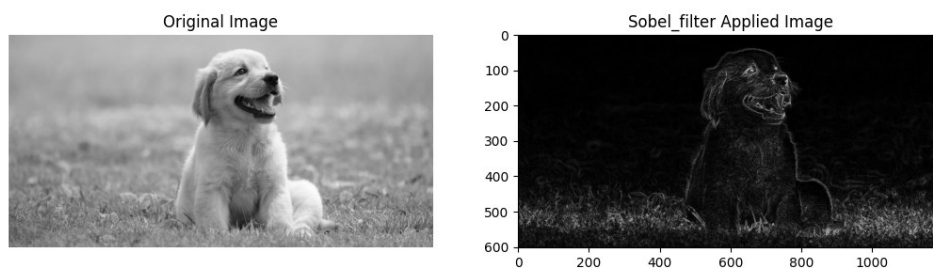
`cv2.equalizeHist(src[, dst]) -> dst`

Explanation:

Image Histogram is a graphical representation of the intensity distribution of an image. It quantifies the number of pixels for each intensity value considered. Histogram Equalization is a method that improves the contrast in an image, in order to stretch out the intensity range.



5. Sobel Filter



```
cv2.Sobel(src, dst, ddepth:int, dx:int, dy:int, ksize:int=3, scale:int=1, delta:int=0, borderType :int=BORDER_DEFAULT)
```

Explanation:

There are many applicative examples of convolutions. One of the most important convolutions is the computation of derivatives in an image. When we want to detect the edges present in an image, we can easily notice that in an edge, the pixel intensity changes in a notorious way. A good way to express changes is by using derivatives. A high change in gradient indicates a major change in the image.

Sobel operator is a discrete differentiation operator. It computes an approximation of the gradient of an image intensity function. The Sobel operator combines Gaussian smoothing and differentiation.

We need to calculate two derivatives: One is horizontal change and the other is vertical change.

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

Conclusion

We learned the process of converting images using OpenCV, not only utilizing APIs, but also learning the mathematical principles inherent in them. We were able to enjoy the learning process in that the results were directly visual.