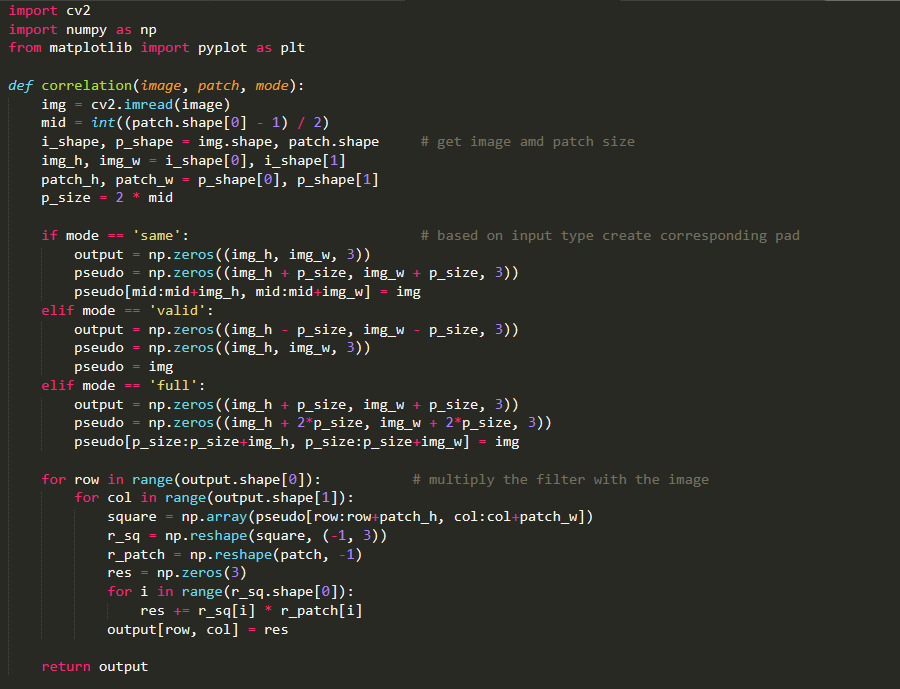
CSC420 Assignment 1

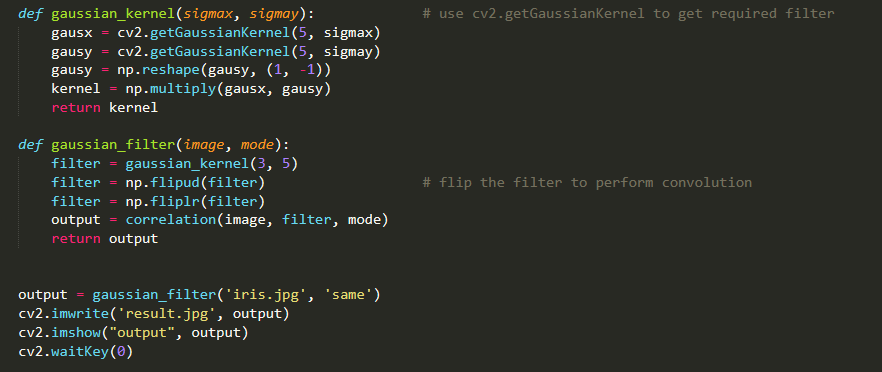
Yubo Wang, 1002138377

1. Code



1. Code: Result:

Function correlation from question 1





1. Convolution is commutative.

Proof:

Assume

Therefore, convolution is commutative

Correlation is not commutative

As the slides in lecture, convolution is equivalent to flipping the filter in both dimensions and apply correlation.

If we apply that to formula, we need to negate the parameter in function to change correlation to convolution.

While

Therefore, correlation is not commutative

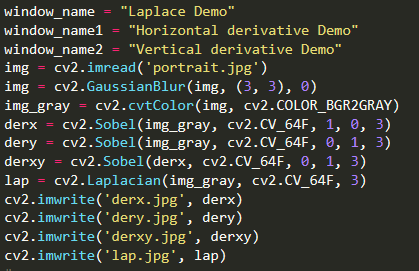
1. The horizontal derivative of a Gaussian Filter G is a separable filter.

We have separated the filter into product of 2 1D filters. Therefore, the horizontal derivative of a Gaussian Filter G is a separable filter.

1. Since the image size is n x n and filter size is m x m, there will be n2 pixels need to be convolved with the filter. The convolution need m2 operations per pixel. Therefore, the time complexity should be O(n2m2)

If the filter is separable, the filter can be separated into two filter with size 1 x m and m x 1. Then convolution just need 2m operations per pixel. Therefore, the time complexity will be O(mn2)

1. Sobel x filter is separable. By flipping the matrix, we get which is also a separable filter. Add these two matrixes we get , which is of course a separable filter.
2. Apply the derivative of Gaussian filter and Laplacian of Gaussian filter to portrait.jpg



Sobel X Sobel Y Sobel XY

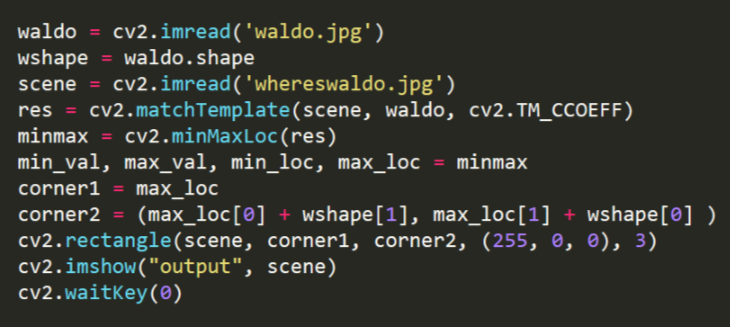




Laplacian



1. Detect waldo.jpg in whereswaldo.jpg using correlation





1. Steps of Canny edge detection
2. Apply Gaussian filter

By applying the Gaussian filter, we smooth the image in order to remove the noise, which will increase our accuracy of detecting edges since noise might be considered as edges.

1. Find the intensity gradients of the image

By getting the magnitude and direction of the gradient of the image, we can count the pixel with high gradient as pixels lie on the edge. Combine all such pixels we can get edges in the image.

1. Apply non-maximum suppression

NMS set all gradient values to 0 except the local maxima to get the location of pixel with biggest change of magnitude of intensity. After finishing this step, we can get thinner edges, but we might lose some edges and gain some gaps.

1. Apply double threshold

We apply a low and a high threshold to the result get from previous step to fill the gap mentioned above and eliminate spurious edges caused by color variation and noise.

Pixels with gradient value larger than high threshold will be marked as strong edge pixel. Pixels with gradient value smaller than high threshold but smaller than low threshold will be marked as weak edge pixel. Pixels with gradient value smaller than low threshold will be completely suppressed.

1. Track edge by [hysteresis](https://en.wikipedia.org/wiki/Hysteresis)

Finally suppress all the weak edges that not connected to any strong edges to eliminate all spurious edges.

1. Since Laplacian can be considered as second derivative, zero crossing of Laplacian of Gaussian indicated a huge change of intensity at the point, which means an edge in the original picture.
2. Code of Canny edge detection

