

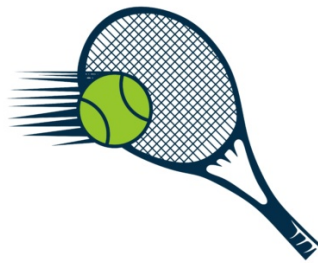
Ph21 Assignment 3

Kyriacos Xanthos

February 2020

Part 1.1

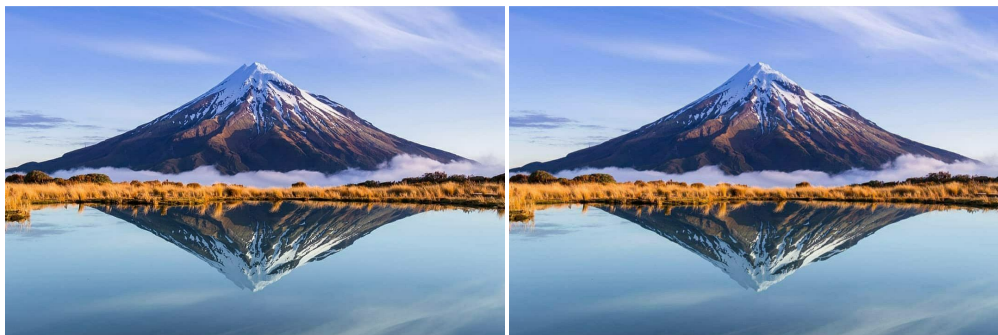
First I will present the images which will be used for edge detection. On the left we can see the original images and on the right using a blurred Gaussian filter of radius 0.2.



(a) original tennis image



(b) blurred with $r = 0.2$



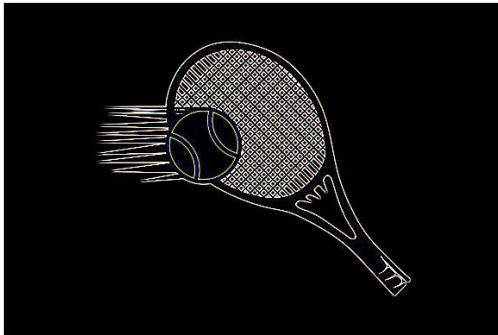
(a) original volcano image



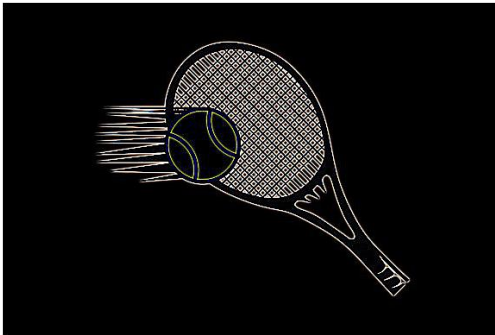
(b) blurred with $r = 0.2$

Part 1.2

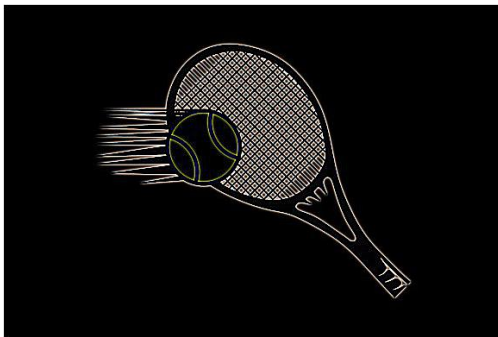
Now I am using the Image Filter called FIND EDGES. Here are some plots using different blurs of the tennis image.



(a) edges on tennis image with no blur



(b) edges using blur $r=0.2$



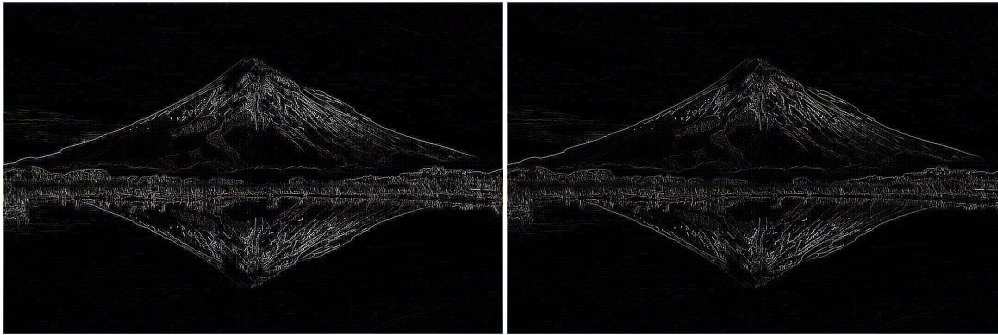
(a) edges using blur $r=0.5$



(b) edges using blur $r=1$

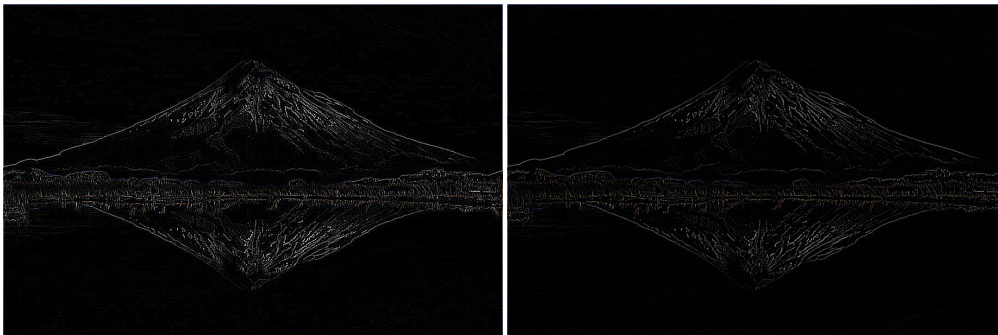
As it can be observed, the blurring filter doesn't work very well on a simple image, especially at large radii.

However looking at the volcano image, it can be seen that blurring works better and produces a clearer pattern of edges. It basically gets rid of some noise but we still need to keep the radius around 0.2.



(a) edges on volcano image with no blur

(b) edges using blur $r=0.2$



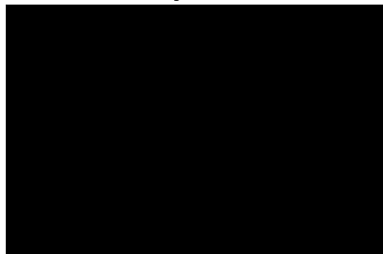
(a) edges using blur $r=0.5$

(b) edges using blur $r=1$

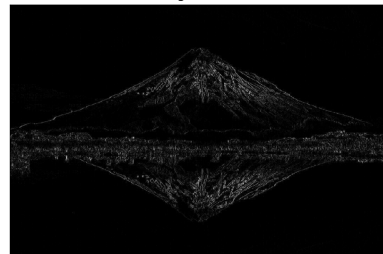
Part 1.3

In this section I will use the stencil of the first derivative of a Gaussian. I have produced different plots with different width parameters. The blurring filter radius I am using is 0.1. It seems like the width of 0.5 is the best for edge detection.

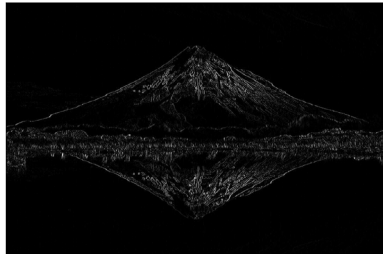
First derivative of gaussian with width: 0.1



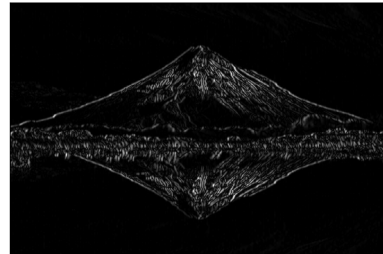
First derivative of gaussian with width: 0.5



First derivative of gaussian with width: 1.0



First derivative of gaussian with width: 2.0



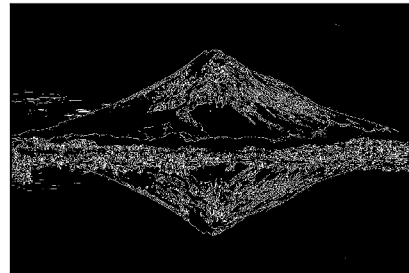
Part 1.4

In this section I will show multiple trials of using different kernels. All of the plots are using a blurring filter of radius 0.2.

Original Image



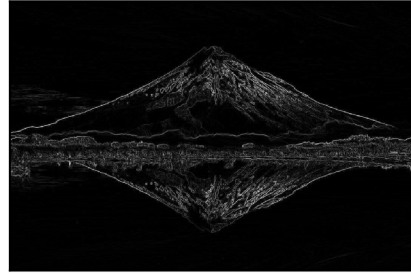
Edge using Canny



Original Image



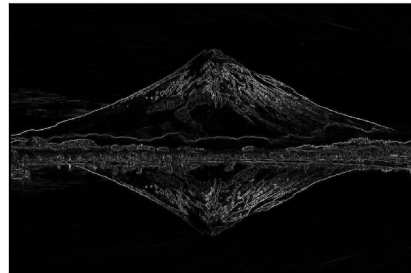
Edge using Sobel



Original Image



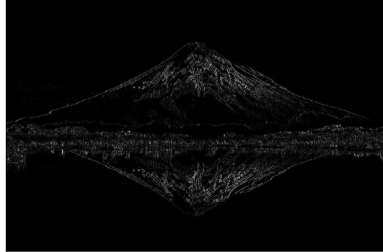
Edge using Scharr



It seems like Canny has more vivid edges than Sobel and Scharr.

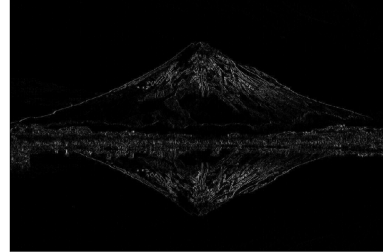
Now I will demonstrate the effect of different blurs using the gaussian derivative stencil. The width I will be using will be 0.5.

First derivative of gaussian with width: 0.5



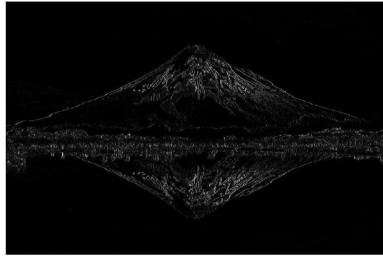
(a) edges using blur $r=0.01$

First derivative of gaussian with width: 0.5



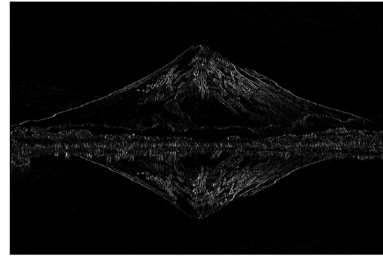
(b) edges using blur $r=0.1$

First derivative of gaussian with width: 0.5



(a) edges using blur $r=0.5$

First derivative of gaussian with width: 0.5



(b) edges using blur $r=1$

Conclusion

It seems like PIL does a very good job in detecting edges on images. It amazed me how even in complex images, it was able to detect edges. It still needed some adjusting by using blur for example to get rid of the noise. I think edge detectors to defeat CAPTCHAs by extracting text from images. I can imagine finding edges can be very useful for systems in self-driving cars!