CIS580 Problem Set 7

Sheil Sarda <sheils@seas.upenn.edu>
CIS580 Spring 2021

Contents

1 Scale Invariant Detection

1.1 Laplacian of Gaussian

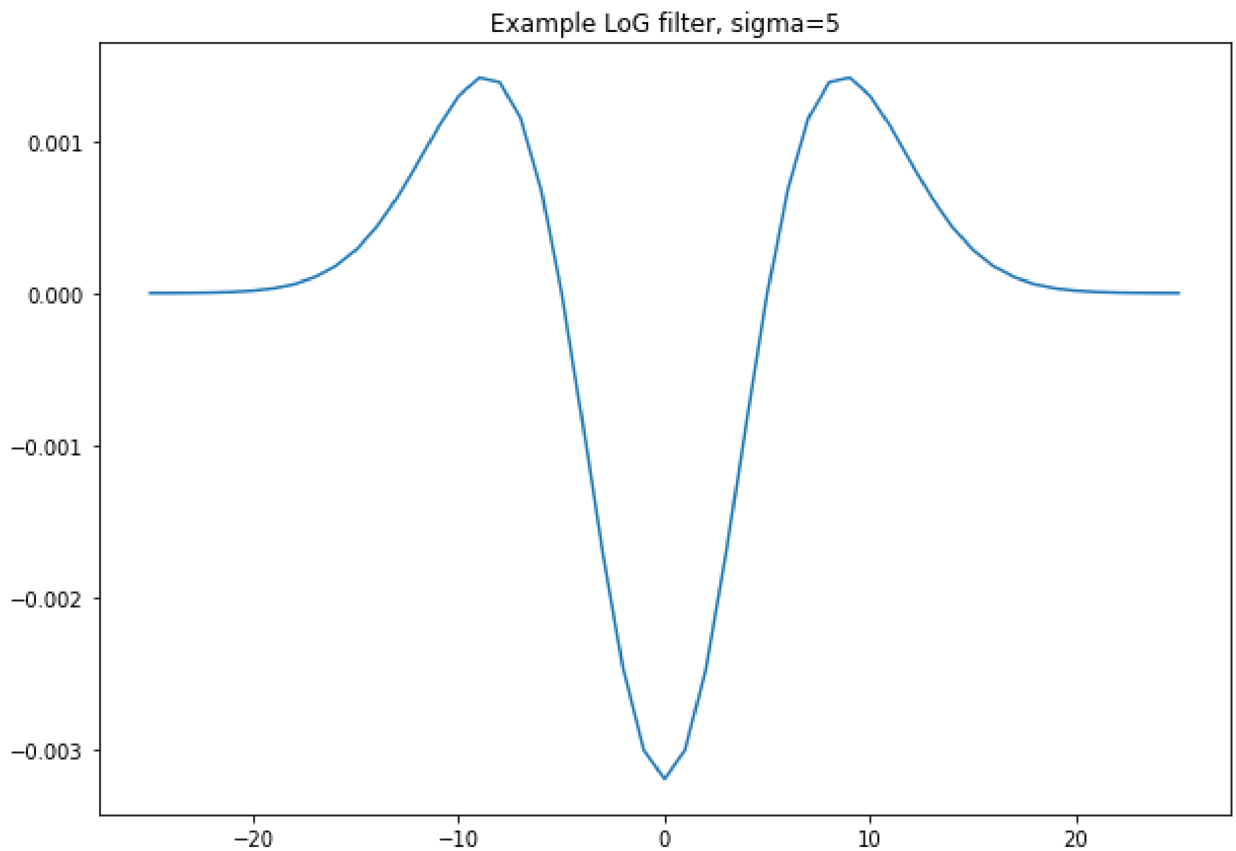


Figure 1

1.2 Approximating a LoG by a DoG

1.2.1 Difference of Gaussians

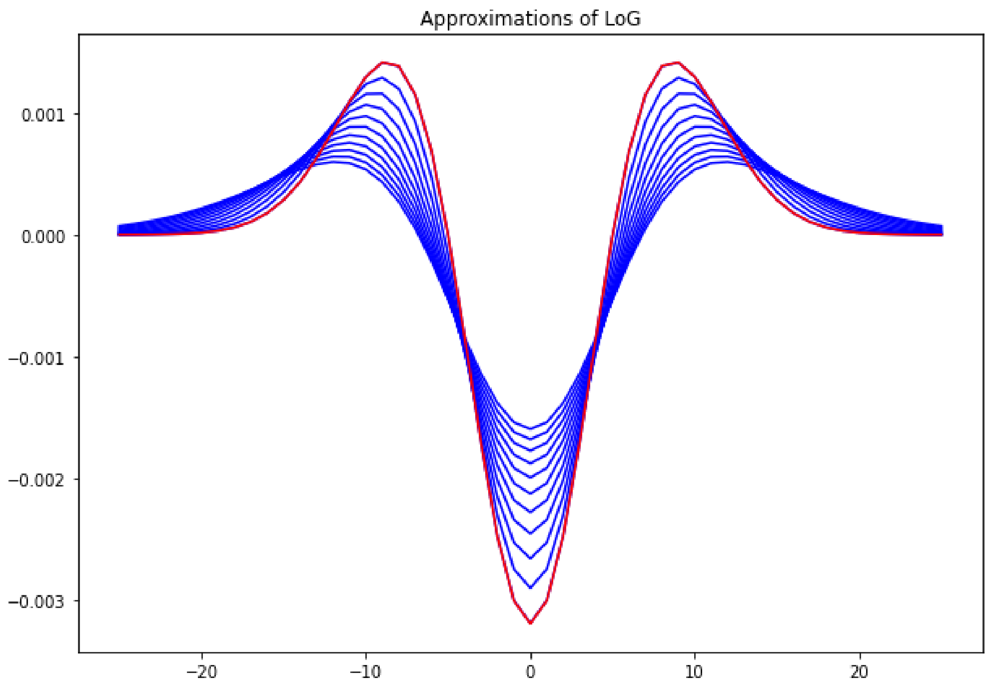


Figure 2

The plots obtained indicate that as k increases from $1 \to 2$, the Difference of Gaussians approximation is scaled down in magnitude.

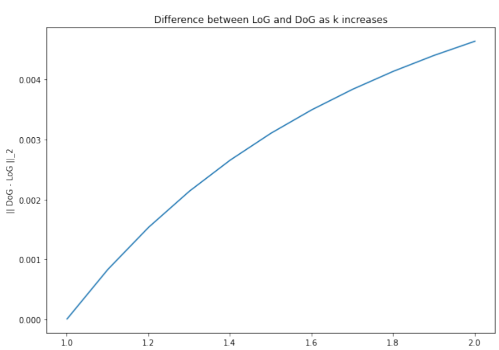


Figure 3

1.2.2 DoG gets closer to LoG as $k \to 1$

We exepect this to happen because in the provided expression, as $k \to 1$, the LoG will approach the DoG.

$$\text{LoG}_\sigma = \frac{1}{(k-1)\sigma^2}\text{DoG}_\sigma$$

1.2.3 Dropping the normalizing factor

We intentionally forget the normalizing factor and just use the difference of Gaussians because the product of a scalar and a Normal distribution will simply apply the constant as a scaling factor to both terms in the DoG expression.

1.3 Detecting sunflowers

1.3.1 Sunflower detection

Figure 4

1.3.2 Blob detection on custom image

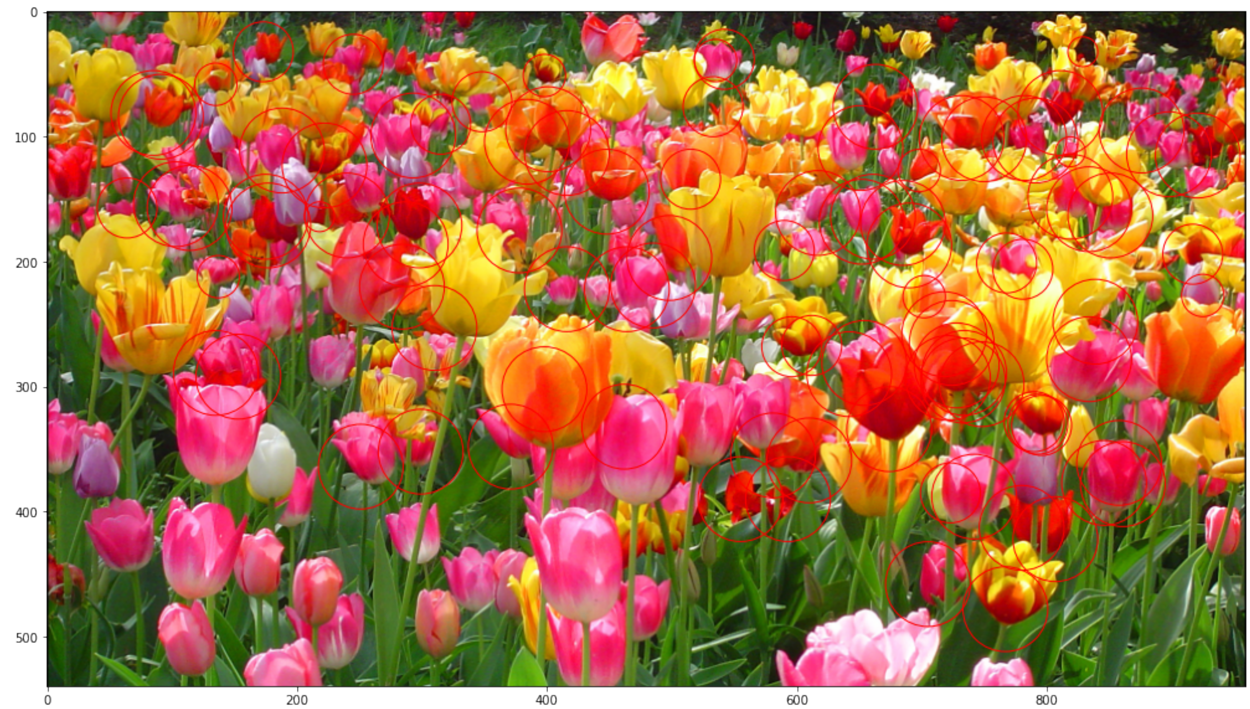


Figure 5

2 Finding Waldo with Gabor Filters

2.1 Gabor Filter in 2D

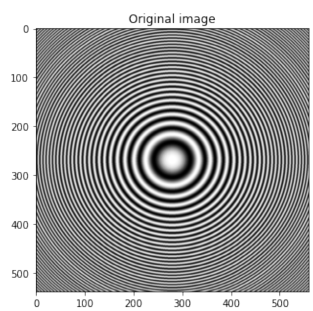
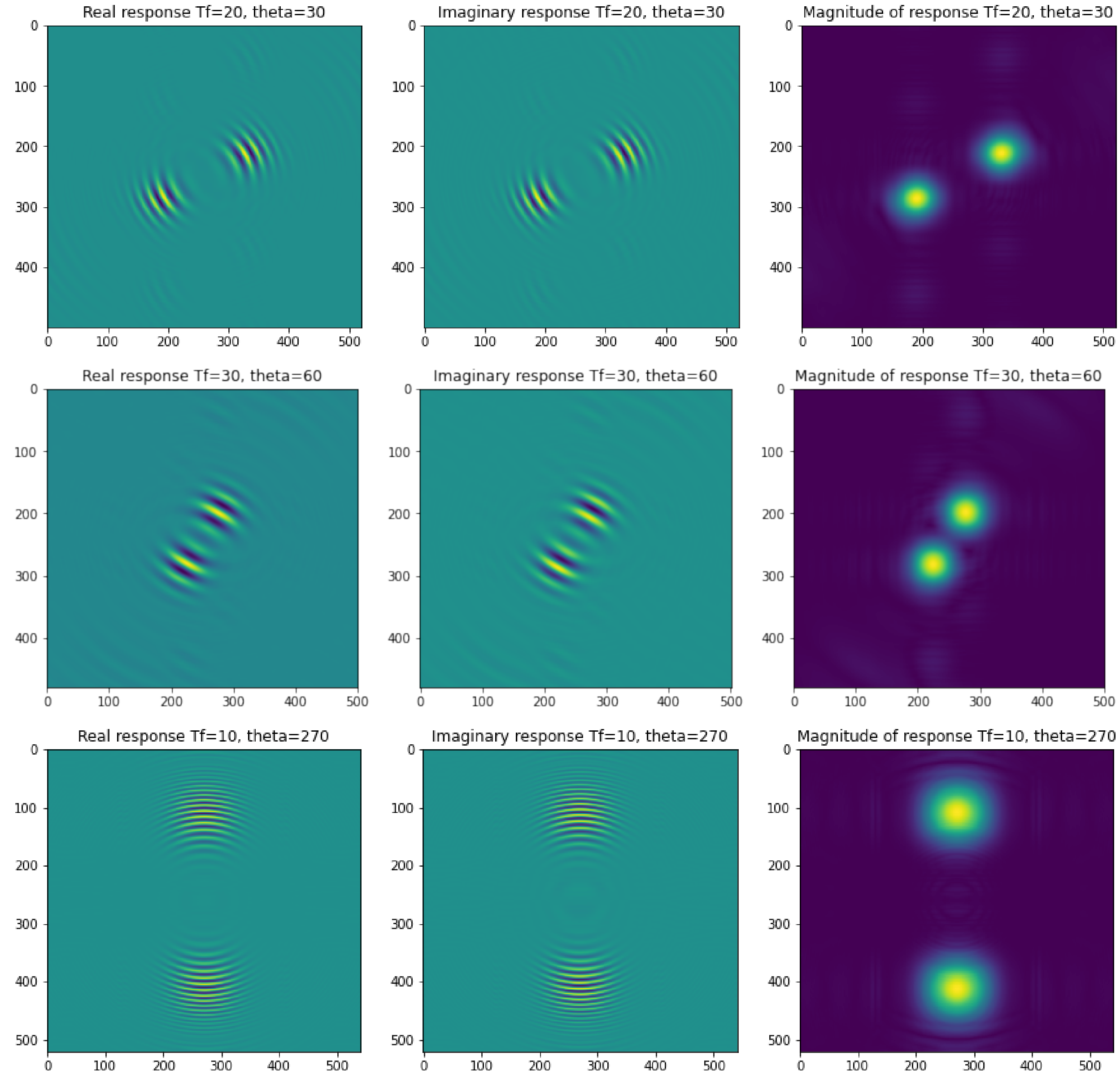


Figure 6: Original Image



(a) $\sigma = 0.1$

Figure 7: Gabor 2D Plot with different parameters

As the spatial period increases, the dots move away from the center. The angle we are using as offset determines the orientation of the image. We also notice that the magnitude of the response is inversely proportional to the distance from the center.

2.2 Finding Waldo

2.2.1 Creation of im_red

The image is created by supressing the green and blue color channels from the RGB image. This helps convert the image to a grayscale image. The choice of suppressing the greens and blues by subtracting their average intensity might be impacted by the fact that we wish to maintain the whitespace in the image, which is represented by RGB$(255, 255, 255)$. Hence, if we eliminate the greens and blues, we will lose the white.

```
im_red = image[:,:,0] - 0.5*(image[:,:,1] + image[:,:,2]);
im_red[im_red < np.mean(im_red[:])] = np.mean(im_red[:]);
im_red = im_red - np.mean(im_red[:]);
```

2.2.2 Finding Waldo

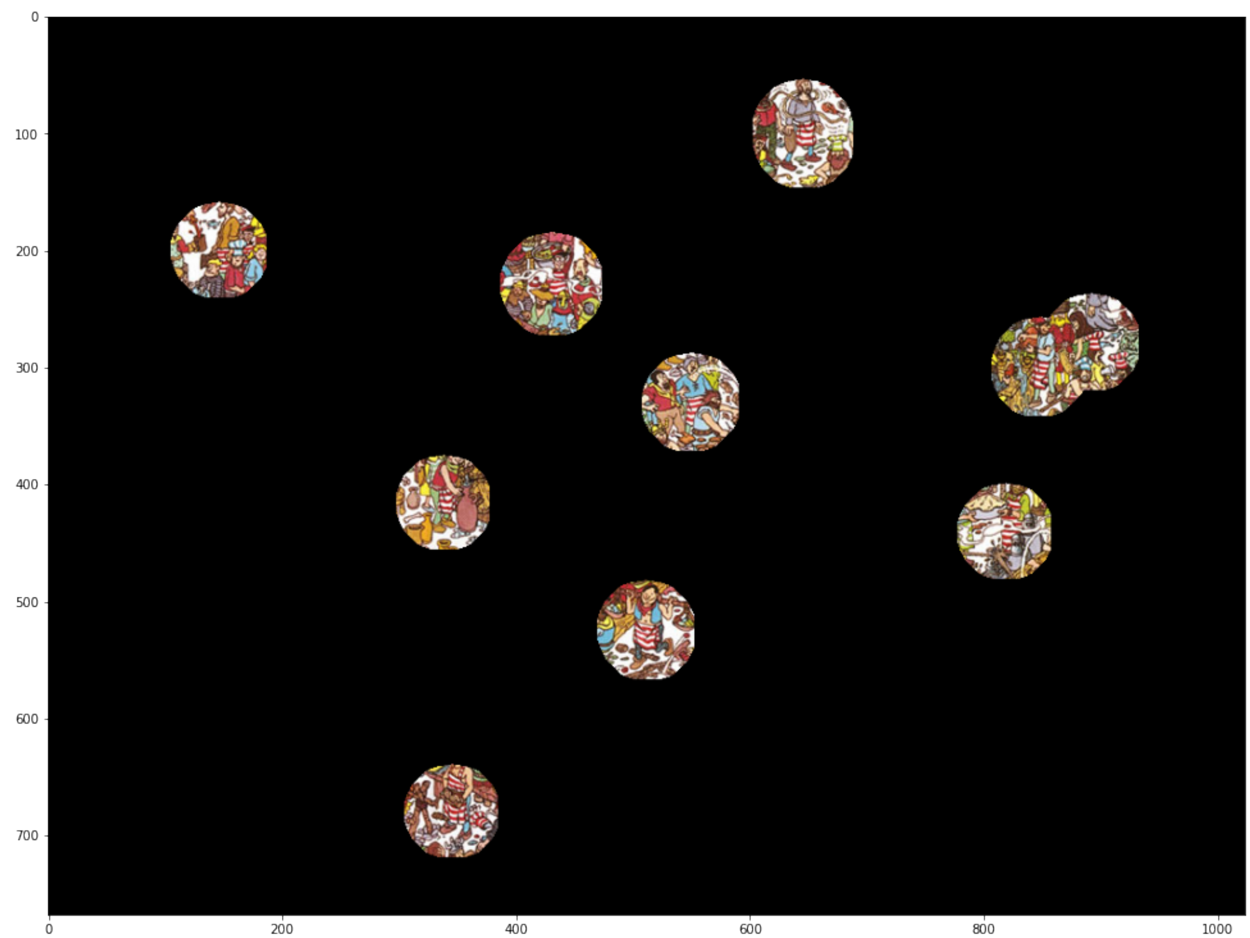


Figure 8: Masked Image with Waldo

Using the above masked image, we can find Waldo!