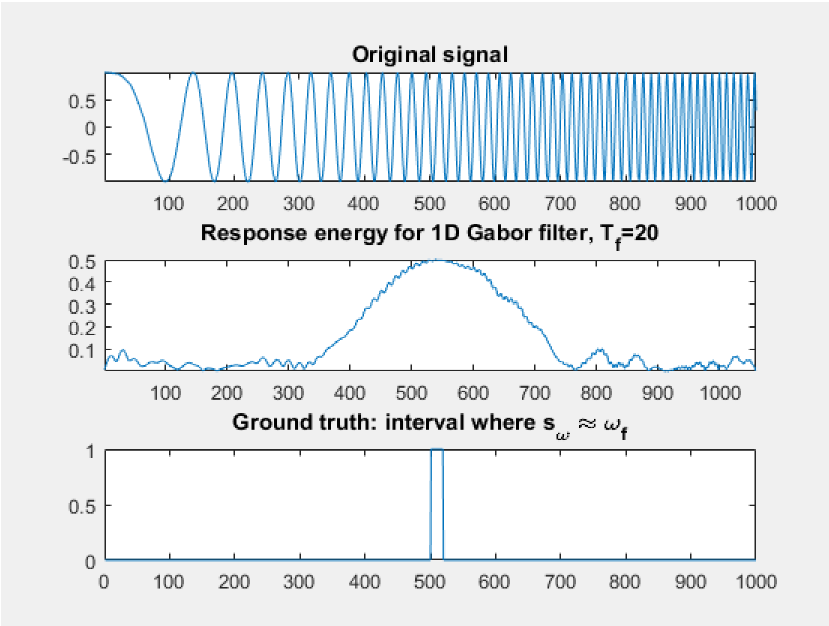
CIS580 HW3

Nikhil Krishnan

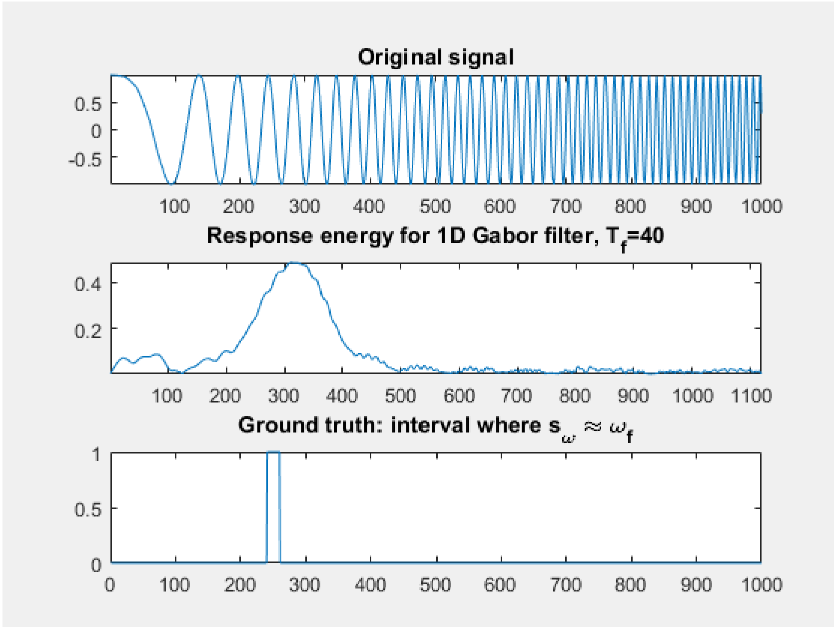
March 2018

1 Gabor Filters

1.1 Gabor Filter in 1d



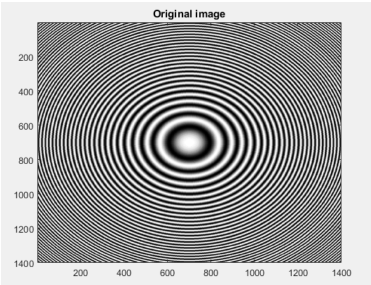
Signal for $T_f = 20$



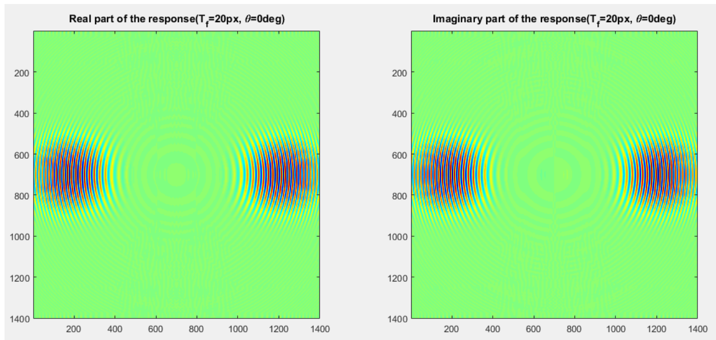
Signal for $T_f = 40$

The plot is correct for both $T_f = 20$ and $T_f = 40$. Interestingly enough, when I plotted $T_f = 60$, the bottom plot appeared to equal 0 everywhere. This is because this frequency does not fall into the ground-truth frequencies. If the range of frequencies was expanded from 0.99w < s < 1.01w, the signal would appear.

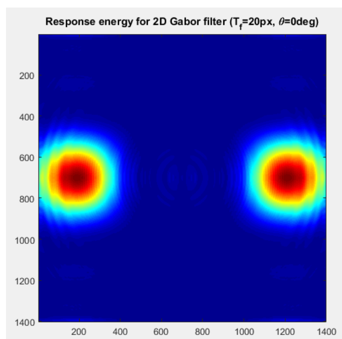
1.2 Gabor Filter in 2d



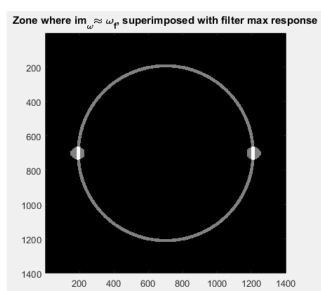
Original Image



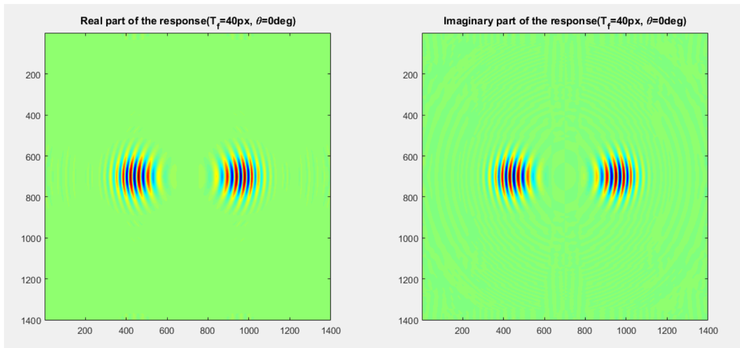
Real and Imaginary Responses for $T_f = 20$, theta $= 0$



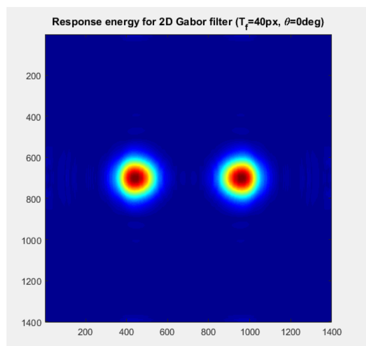
Response Energy for $T_f = 20$, theta $= 0$



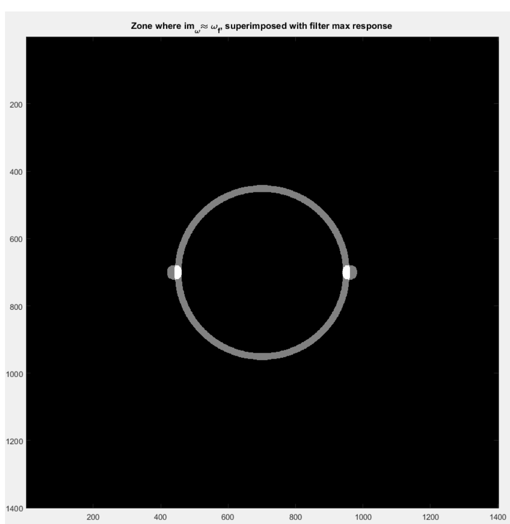
Filter Response for $T_f = 20$, theta $= 0$



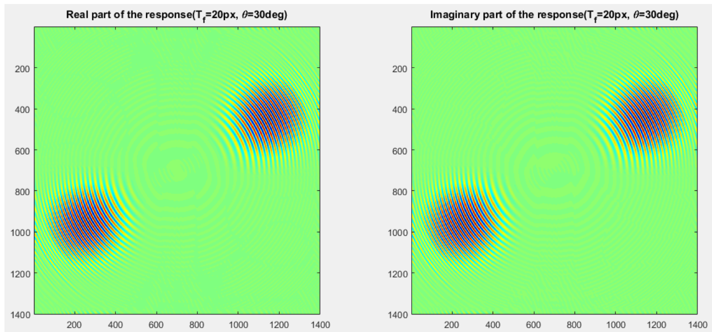
Real and Imaginary Responses for $T_f = 40$, theta $= 0$



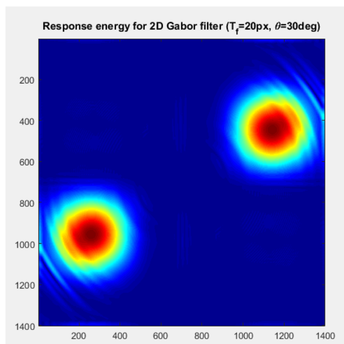
Response Energy for $T_f = 40$, theta $= 0$



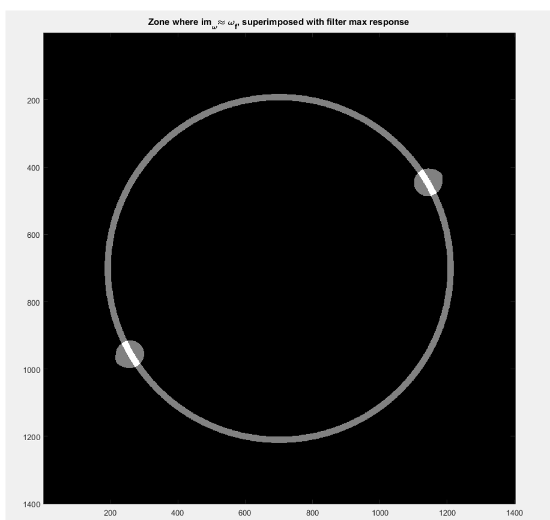
Filter Response for $T_f = 40$, theta $= 0$



Real and Imaginary Responses for $T_f = 20$, theta = 30



Response energy for $T_f = 30$, theta = 30

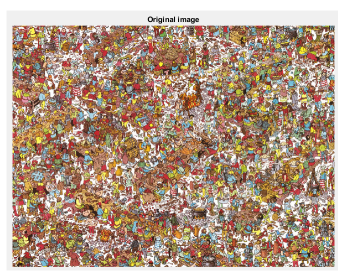


Filter Response for $T_f = 30$, theta = 30

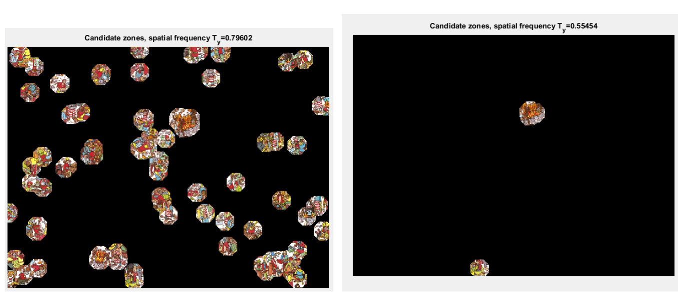
For all tested values of T_f, and both values of theta, the ground truth matches the maximum response. When T_f increases, the response energy, and thus the filter response, moves closer together. As theta increases, the responses rotate counter-clockwise with the magnitude of theta. This is because the frequency decreases as spatial period increases.

1.3 Where's Waldo?

a) Where is Waldo is a script that uses image detection to find Waldo in a bustling image. The user selects an image of a pattern that resembles Waldo's shirt. The colors of this image are then weighted such that the blue and red are more dominant than green (as Waldo has red on his shirt and blue pants). After this, the image is converted to a gray color map for simpler analysis. The most dominant frequency, and thus most dominant spatial period of the image pattern that resembles Waldo's shirt is found using determineStripPeriod. This spatial period is used to create a Gabor filter that can then be convolved with the original weighted image to find the energy of the image compared to the striped pattern. The function 'imdilate' combined with the detection mask, which narrows down the points to energy values above a certain threshold, shows all matching patterns in the original image. As a result, the script can find any image that looks like stripes, and thus, locate Waldo.
b) As stated above, the equation allows blue and red to be weighted much more than green, as Waldo's shirt is red and his pants are blue.

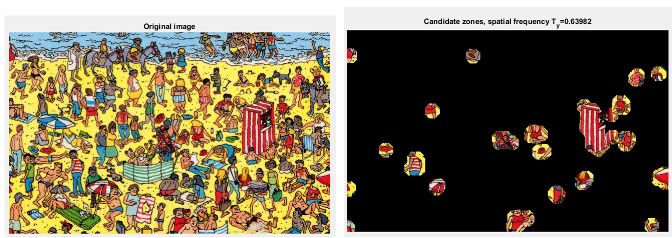


Original Waldo Image



0.80 threshold on the left, 0.90 threshold on the right

0.86 threshold

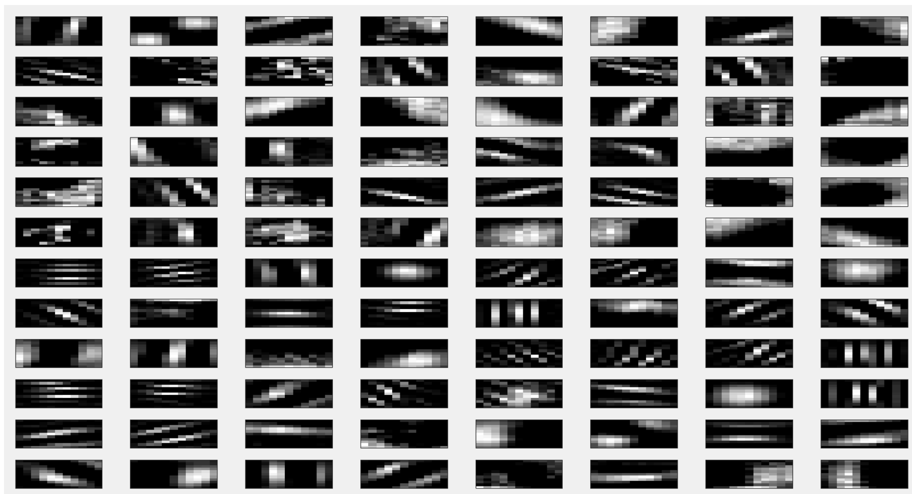


A second Waldo image (Proof of Concept)

When the threshold was 0.80 of the maximum, I felt that there were too many matching images, while at 0.90 the function was not consistently finding Waldo. I ended up settling on 0.86, which showed Waldo all the time, while minimizing the amount of matching images. I also tested another Waldo image as proof of concept, which worked perfectly.

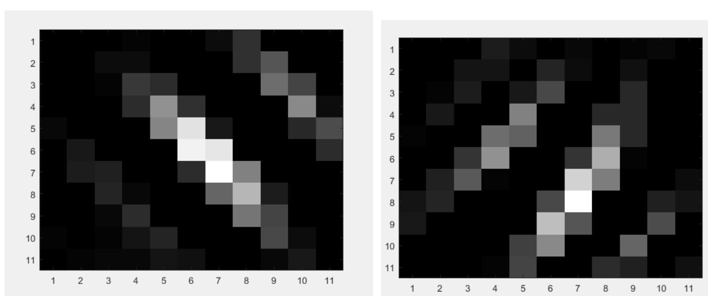
2 AlexNet

2.1 Weight Visualization

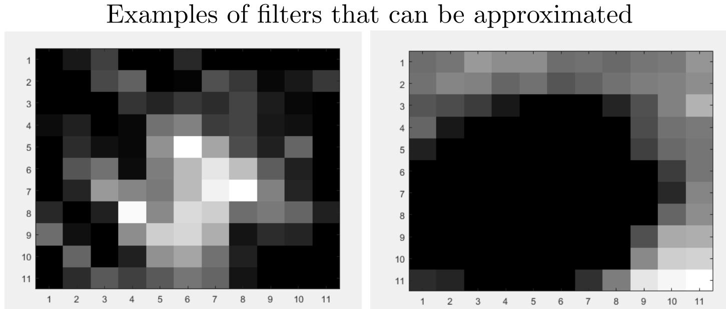


12x8 Grid of the AlexNet Filters

The AlexNet filters that can be approximated by Gabor filters are the ones that resemble periodic stripes, as this is what the general visualization of a Gabor filter looks like. Some screen shots of these images are below. The ones that do not resemble this pattern can not be approximated by Gabor filters.



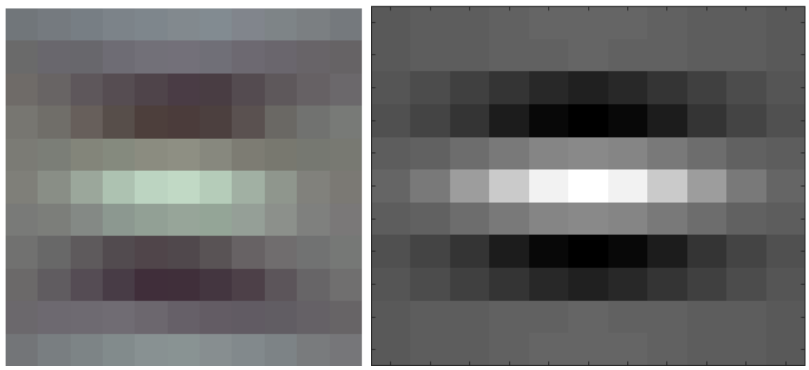
Examples of filters that can be approximated



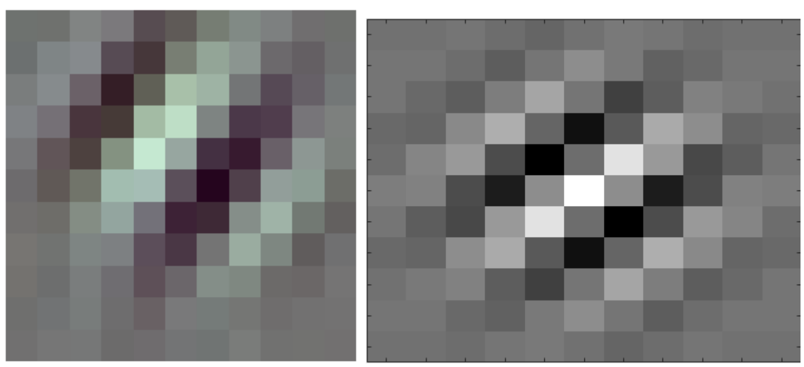
Examples of filters that can not be approximated

If we were to replace the first layer of AlexNet with a bank of 96 Gabor filters, we would have to learn 96 different Gabor filters with 5 inputs each, and thus, 480 different parameters.

2.2 Approximating Filters with Gabors



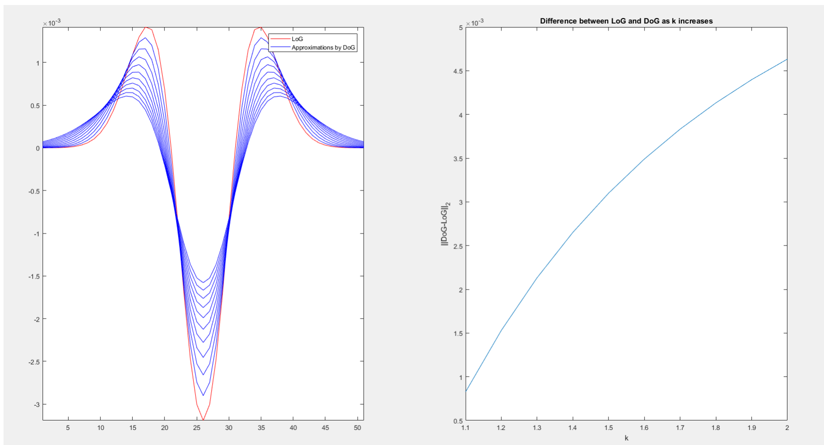
Left is Original, Right is Gabor Approximation
Parameters: T_f = 5, theta = 90, Sigma = [5 0; 0 5]



Left is Original, Right is Gabor Approximation
Parameters: T_f = 3, theta = -50, Sigma = [5 0; 0 5]

3 Scale Invariant Detection

3.1 Approximating a LoG by a DoG



b) I would expect the DoG to get closer to the LoG as k tends to 1, as k acts like a timestep. As a result, when k is lower, there are more iterations within the approximation, and thus, there is a more accurate result.

c) When k is constant, the scaling factor essentially just scales the difference. However, when looking at scale space, we just want to see the difference. The normalizing factor just makes the values smaller.

3.2 Detecting Sunflowers

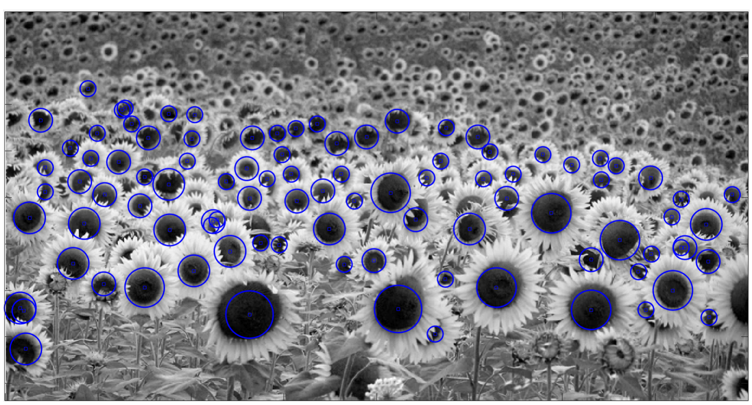
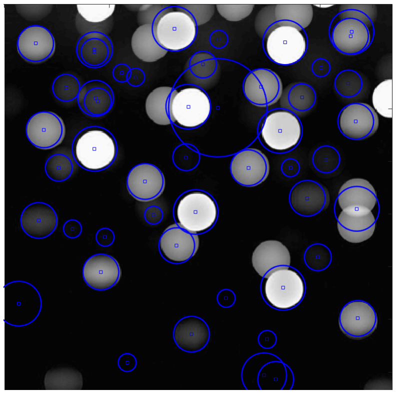


Image of flower detection



Detecting light circles

The two images are found above. I scaled the radius using the sigma values of each point. The light circles were created by filtering out maximum values and keeping minimums.