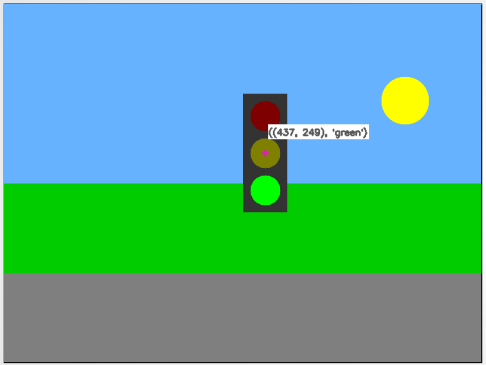
Computer Vision
Fall 2021
Problem Set #2

Ian Dover
idover3@gatech.edu

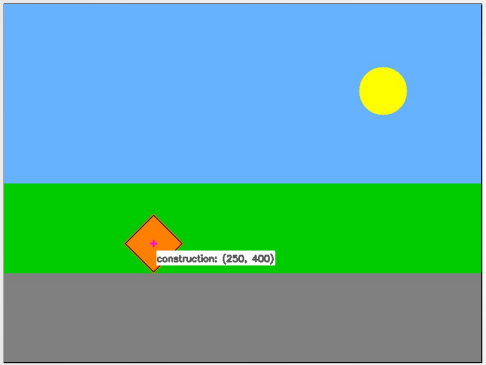
1a)Traffic Light Detection



ps2-1-a-1

Coordinates and State:
(437, 249), color: green

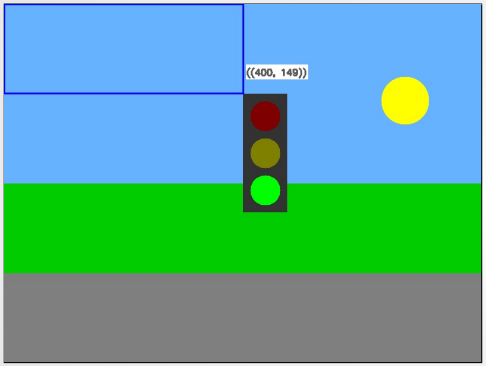
1b)Traffic Sign Detection - Construction



ps2-1-b-1

Coordinates:
(250, 400), construction

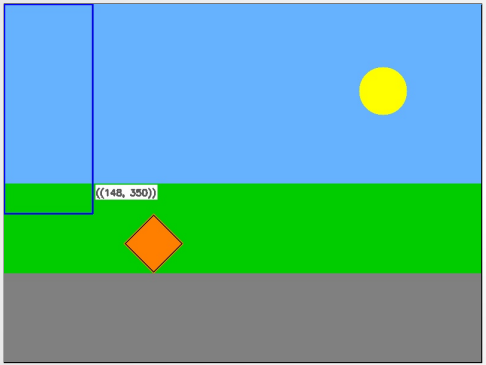
2a)Template Matching - TL



ps2-2-a-1

Coordinates:
(400, 149)
Note: The coordinates I returned in the function represent the top, right corner of the template match. I was unsure how to get the box to surround the light.

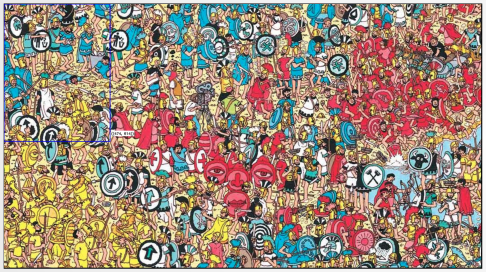
2b)Template Matching - Construction



ps2-2-b-1

Coordinates:
(148, 350)
Note: The coordinates I returned in the function represent the top, right corner of the template match. I was unsure how to get the box to surround the construction sign.

2c)Template Matching - Finding Waldo



ps2-2-c-1

Coordinates:
(474, 614)
Disclaimer: The box is in dark blue which makes it hard to see here.
Note: The coordinates I returned in the function represent the top, right corner of the template match. I was unsure how to get the box to surround Waldo. The bottom, right corner of the box is the position of the top, left corner of the template match.

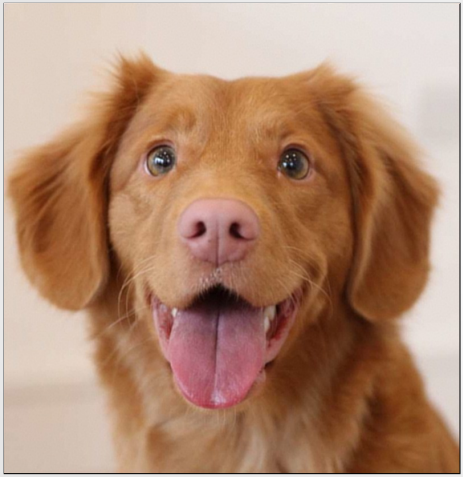
2d)Discussion

What are the disadvantages of using Hough based methods in finding Waldo? Can template matching be generalised to all images?

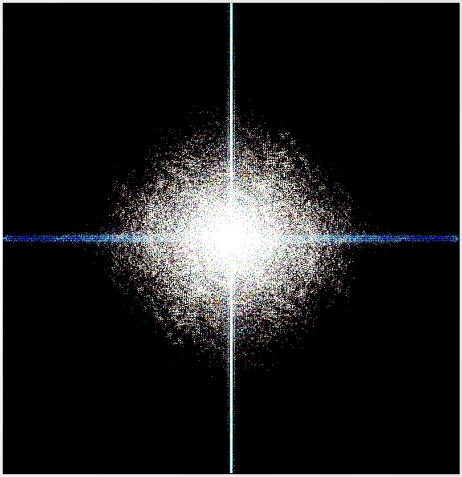
Explain Why/Why not. Which method consistently performed the best, why?

While the Hough Transform is a robust algorithm that is very good at handling noise, it is not good at handling shadows. If an image contains shadows, it will produce odd results. Additionally, the Hough Transform normally consists of a Canny operator step that grabs the edges of an image, and then the Hough Transform step is applied to that new edge image. In the Waldo image, there are too many edges, and this would make it very difficult for a Hough Transform to pick-up on Waldo. Moreover, Waldo is not polygonal, so using a preset shape in the Hough Transform would not be an option, and therefore, a Hough Transform searching for an irregular shape would have to be used; however, that irregular shape would not be known beforehand, making this analysis near impossible.
Template matching can not be generalized to all images, and there are multiple reasons why:
1. Sometimes objects are too close together
2. Sometimes the template simply isn't good enough and the person performing the analysis is not aware of what they should be matching on ahead of time.
3. Changes in image composition (shadows and backgrounds) can often throw off the template match because it would look for particular shades of colors. Additionally, the template may consist of a background inconsistent with the background of the image.
4. Finally, template matching would have a difficult time dealing with scaled images.
The normalized cross correlation method consistently performed the best. Normalized cross correlation is more robust at detecting changes in illumination, but sum of square differences is less computationally complex. Normalized cross correlation ensures that bright patches in the images are not treated favorably over dark patches.

4a1) Compression - Threshold 0.1

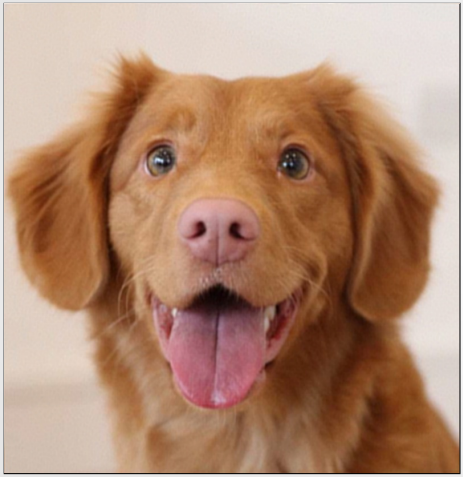


ps2-4-a-1 resulting image

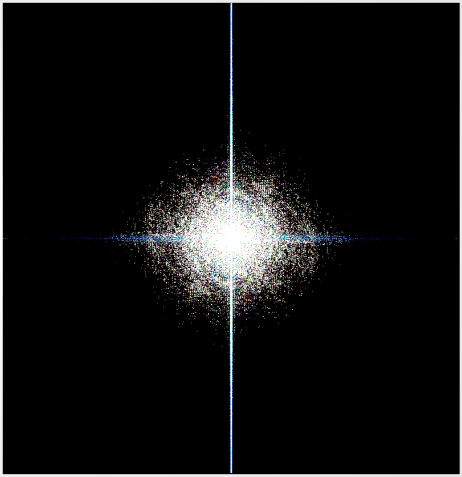


ps2-4-a-1 frequency domain

4a2) Compression - Threshold 0.05



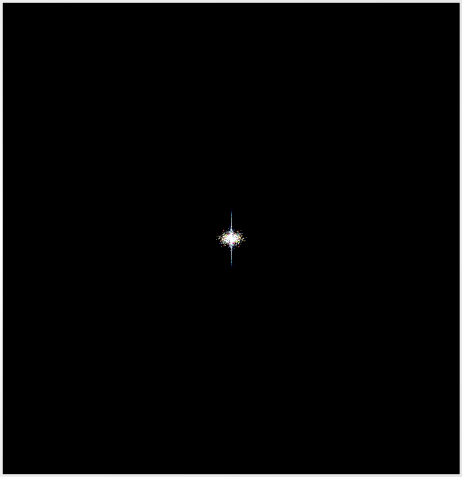
ps2-4-a-2 resulting image



ps2-4-a-2 frequency domain

4a3) Compression - Threshold 0.001

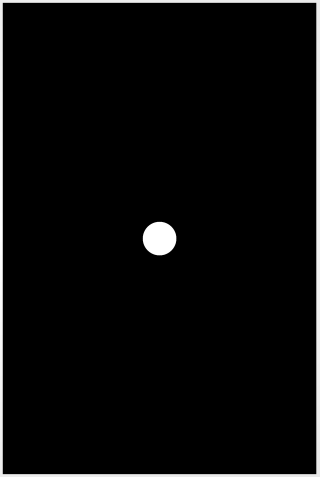
ps2-4-a-3 resulting image



ps2-4-a-3 frequency domain

5a1) Filtering - Radius 100

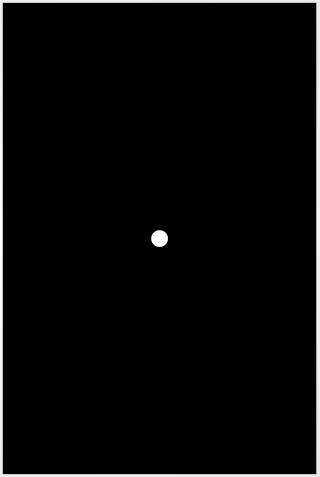
ps2-5-a-1 resulting image



ps2-5-a-1 frequency domain

5a2) Filtering - Radius 50

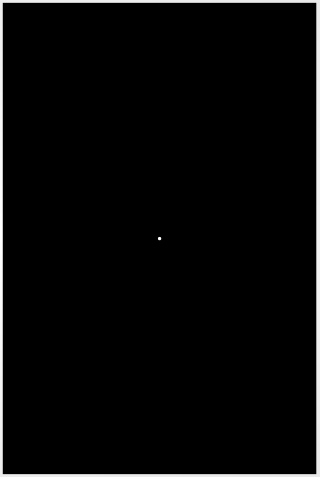
ps2-5-a-2 resulting image



ps2-5-a-2 frequency domain

5a3) Filtering - Radius 10

ps2-5-a-3 resulting image



ps2-5-a-3 frequency domain

5b) Discussion

What are the differences between compression and filtering? How does this change the resulting image?

The compression involved performing a magnitude threshold on the frequency image, but the filtering involved performs a low pass filter on the frequency image. These are not inherently the same. In compression, values with a high magnitude can theoretically exist anywhere in the image, but they generally reside close to the middle. By performing a magnitude threshold, you are removing data from an image with low energy, and this low energy data is typically not seen by the human visual system, allowing for unimportant data to be removed and compression to occur. Furthermore, low pass filtering as seen in the cat image removes high frequency values. This quickly damages the image as can be seen on the image with a radius of 50; the image has artifacts known as ringing, and this is a common problem for when an image is limited to particular frequency bands. Although, when the image compression is very high, you get a slight case of ringing because frequency bands are naturally removed from the image representation by random chance of not containing data with high energy; this results in the apparent horizontal lines on the compression image with threshold of 0.001 (1000x compression).

5c) Discussion

Given an image corrupted with salt and pepper pepper noise, what filtering method can effectively reduce/remove this noise? Also explain your choice of filtering method.

The best filtering method for correcting salt and pepper noise would be the median filter. The reason a median filter would be chosen for this purposes is because most other filtering methods involve some sort of averaging component (convolution, etc) that would corrupt the other data points with a blatant outlier data points (salt and pepper data). By using a median filter, these salt and pepper data points are effectively replaced with data in the normal range of values within the image, and no artifacts are left behind from the salt and pepper noise by an averaging operation.
The flaw with using a median filter approach is that median filters are known for breaking up edges, and this is due to the fact that edges tend to have sharp changes in values similar to salt and pepper noise. Therefore, a method for mitigating this effect would be necessary in developing a robust median filter algorithm for correcting salt and pepper noise.