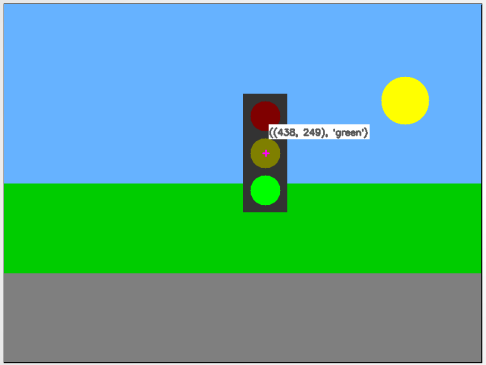
Computer Vision
Spring 2022
Problem Set #2

Ali Alrasheed
Ajar3@gatech.edu

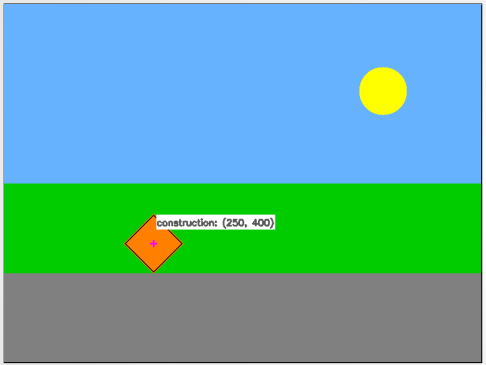
1a)Traffic Light Detection



ps2-1-a-1

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

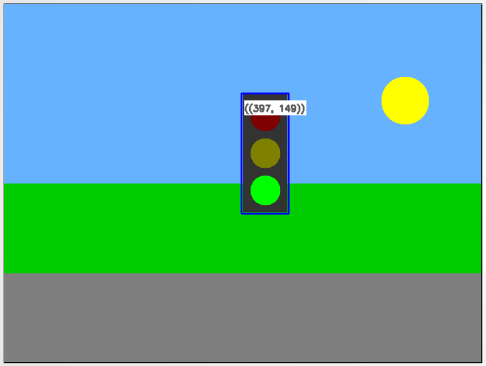
1b)Traffic Sign Detection - Construction



ps2-1-b-1

Coordinates:
(250, 400), color: Not-applicable

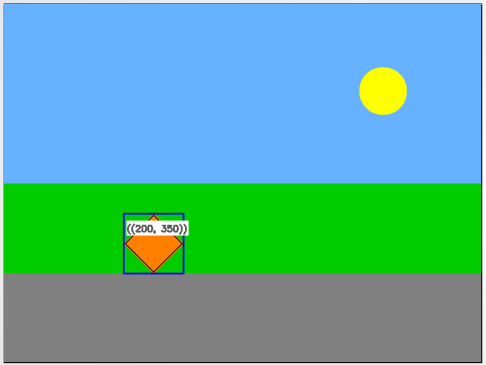
2a)Template Matching - TL



ps2-2-a-1

Coordinates:
(397, 149) – Top left corner

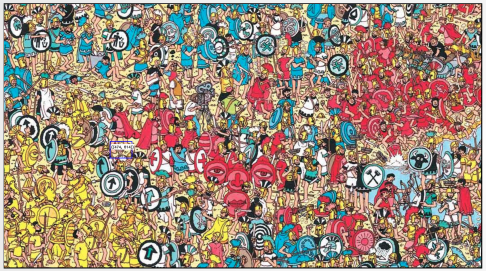
2b)Template Matching - Construction



ps2-2-b-1

Coordinates:
(200, 350) – Top left corner

2c)Template Matching - Finding Waldo



ps2-2-c-1

Coordinates:
(474, 614) – Top left corner

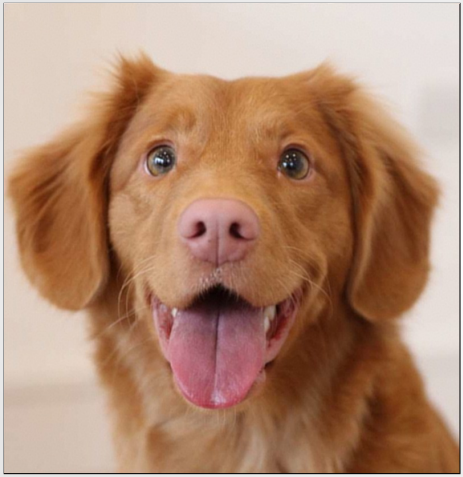
2d)Discussion

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

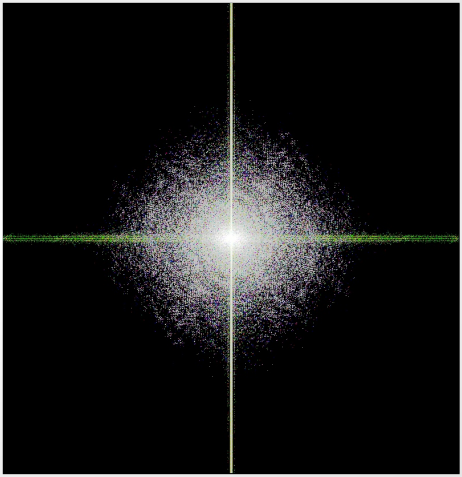
Explain Why/Why not. Which method consistently performed the best, why?

Hough transform can preform very well in finding a specific shape such as circles or lines since it relies on edges or thresholding. It can also deal with theses shapes at different rotation and sizes. However, it cannot find features inside that shape which is what make Hough transform not the best approach to find Waldo. In case of Waldo, we don't only need to find the correct shape but also what is in that shape (features inside the shape). This, however, can be done using the template matching techniques such NSSD and NCCOR.
The template matching techniques using SSD,NSSD,CCOR,NCCOR are not easy to generalize. This is because the best result is obtained when the size of the matched object/shape is the same in both the template image and matched image. Re-sizing the image may help but it is not easy to know the correct size that the inquiry image should be. In summary, for the best result using template matching techniques, the matched object needs to be the same size, orientation, and similar pixel intensity in both the template and the inquiry image. Due to this, template matching cannot easily generalized.
In my experiment, the only techniques that didn't work was CCOR, since it is not normalized, it finds the brightest part of the image. Furthermore, the rest of the techniques (NCCOR,SSD,NSSD) preforms equally well in the experiment. However, the NCCOR seems the best technique among the rest since it is less sensitive to the overall image brightness, even though the NSSD is less computational intensive, and require less time.

4a1) Compression - Threshold 0.1

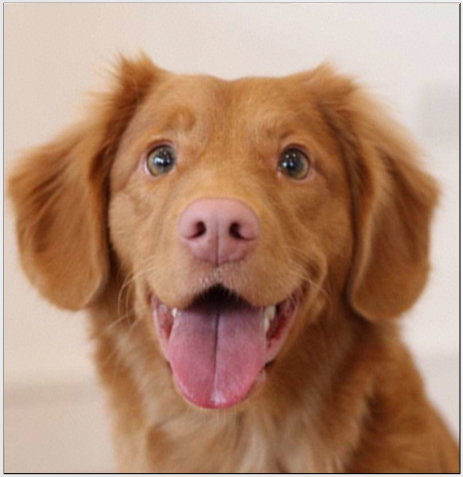


ps2-4-a-1 resulting image

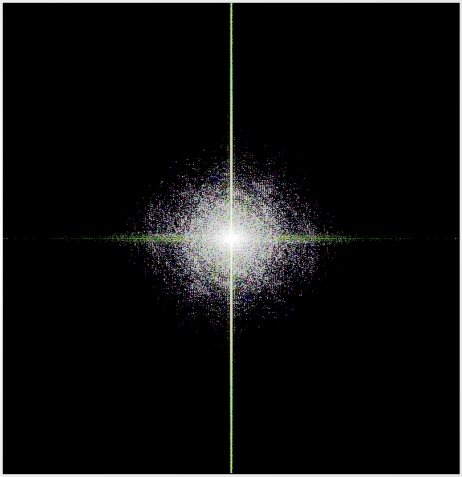


ps2-4-a-1 frequency domain

4a2) Compression - Threshold 0.05



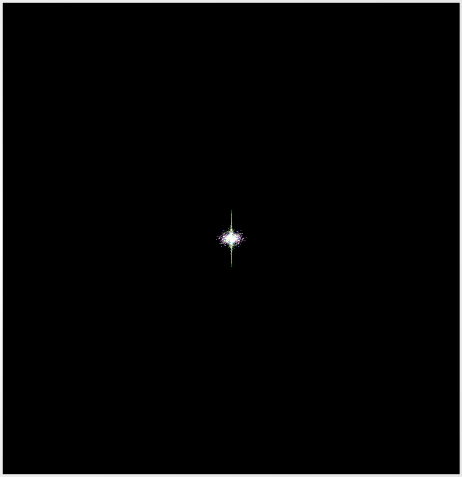
ps2-4-a-2 resulting image



ps2-4-a-2 frequency domain

4a3) Compression - Threshold 0.001

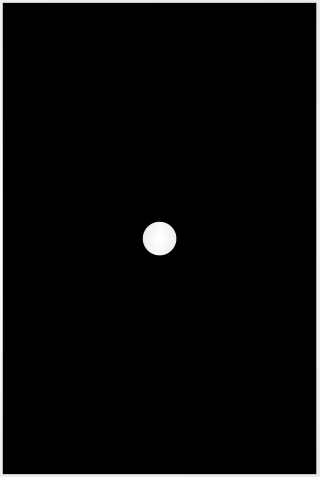
ps2-4-a-3 resulting image



ps2-4-a-3 frequency domain

5a1) Filtering - Radius 100

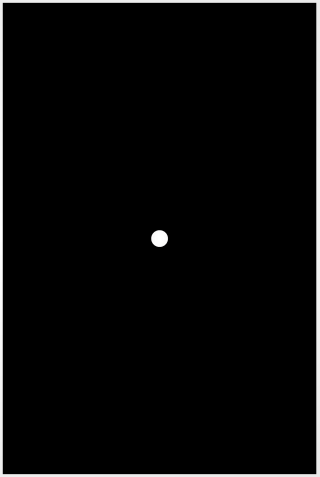
ps2-5-a-1 resulting image



ps2-5-a-1 frequency domain

5a2) Filtering - Radius 50

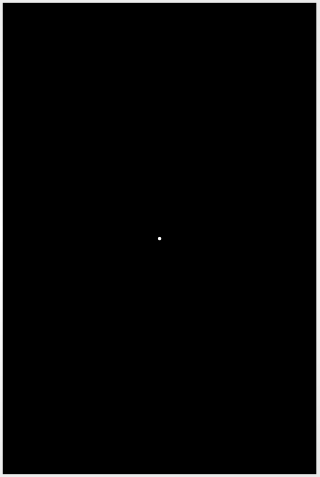
ps2-5-a-2 resulting image



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?

Compressing the image is done by converting it to frequency domain and keeping only the dominant frequency in the image [magnitude], regardless of their actual frequency (low or high frequency). By zeroing out all the weak frequencies in the image, the result image (using inverse FT) is compressed image that needs less data to be stored, but still preserve the overall quality of the image (depending on percentage of frequencies to keep).

However, low frequency filter only allows low frequency to pass, and it blocks all the high frequencies regardless of the their magnitude. The result image is usually a blurry/smoothed version of the original image as it can be seen from section 5. Aliasing could be also noticed, since the high frequency components are removed.

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 salt and pepper is characterized as a high-frequencies noise. Thus, a median filter works pretty well with salt and pepper noise. Furthermore, Low-filter approach can also effectively reduce or remove the noise in the image. However, depending on the cut-off frequency of the low filter, the resulting image could be blurry.
Gaussian and mean filter could also be able to reduce the salt and pepper noise, but they are not the best filter to use.