Assignment 1

Authors:

Achal Shah

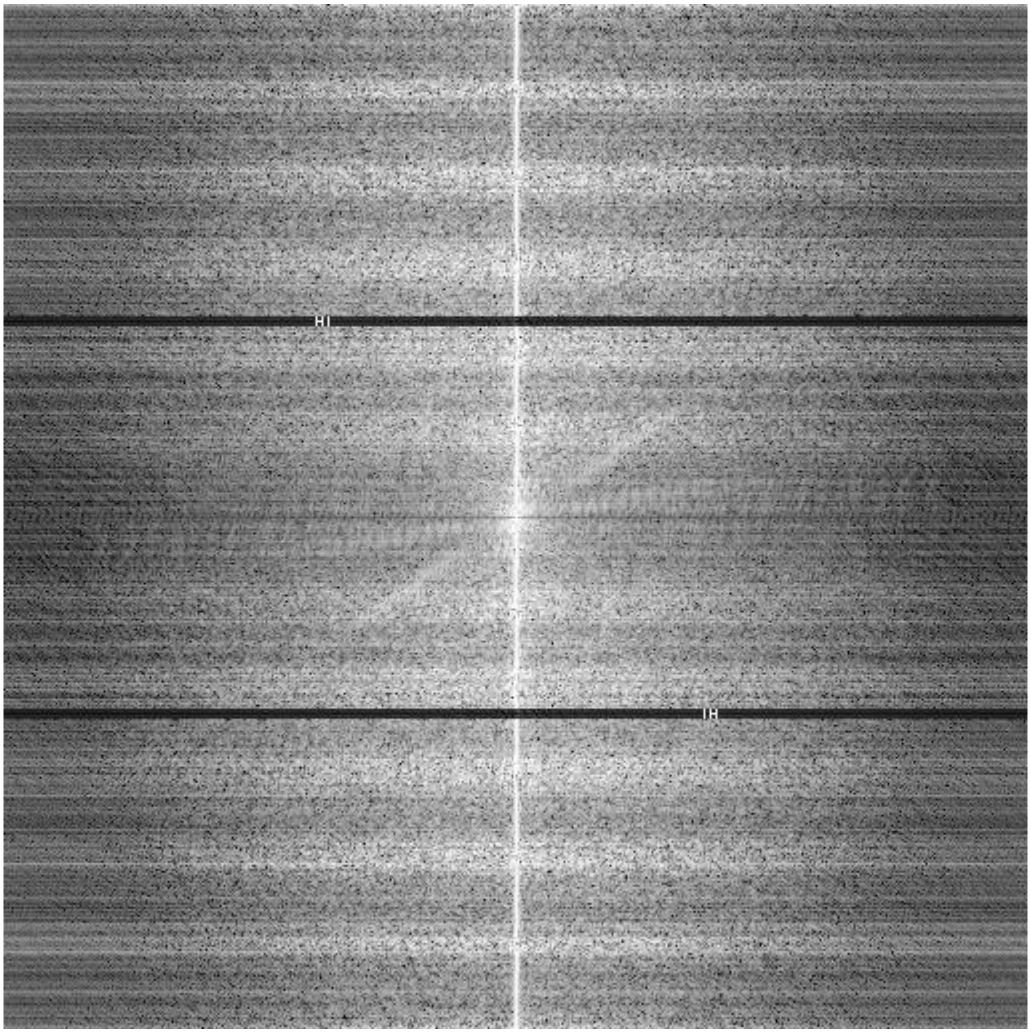
Sachin Sable

Kushal Kokhe

Lijiang Guo

Part 1: Fourier domain

1.1 Spectrogram of noise1.png



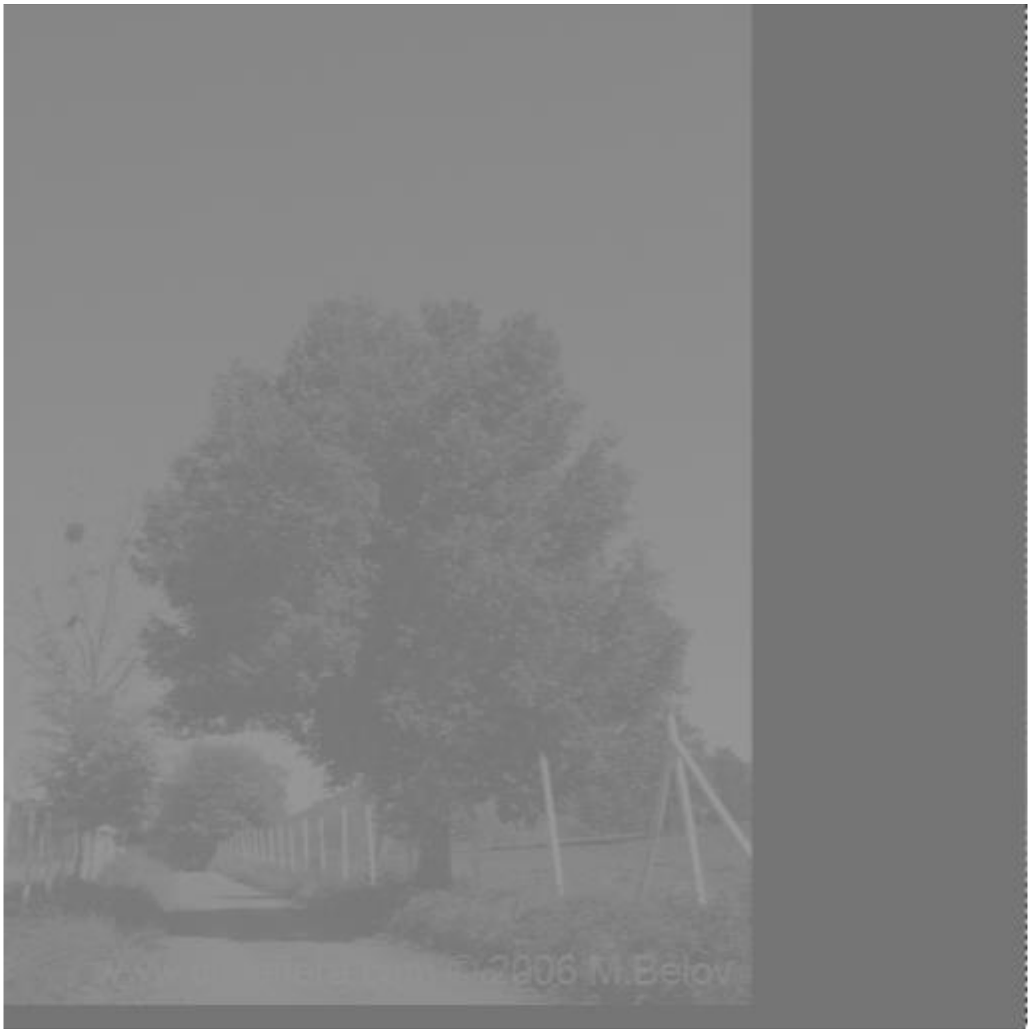
1. 2 Remove interference

As we can clearly see in the spectrogram of noise1.png image that there are HI and IH words in the spectrogram. So, intuitively we can clearly understand by removing these letters, noise should get removed.

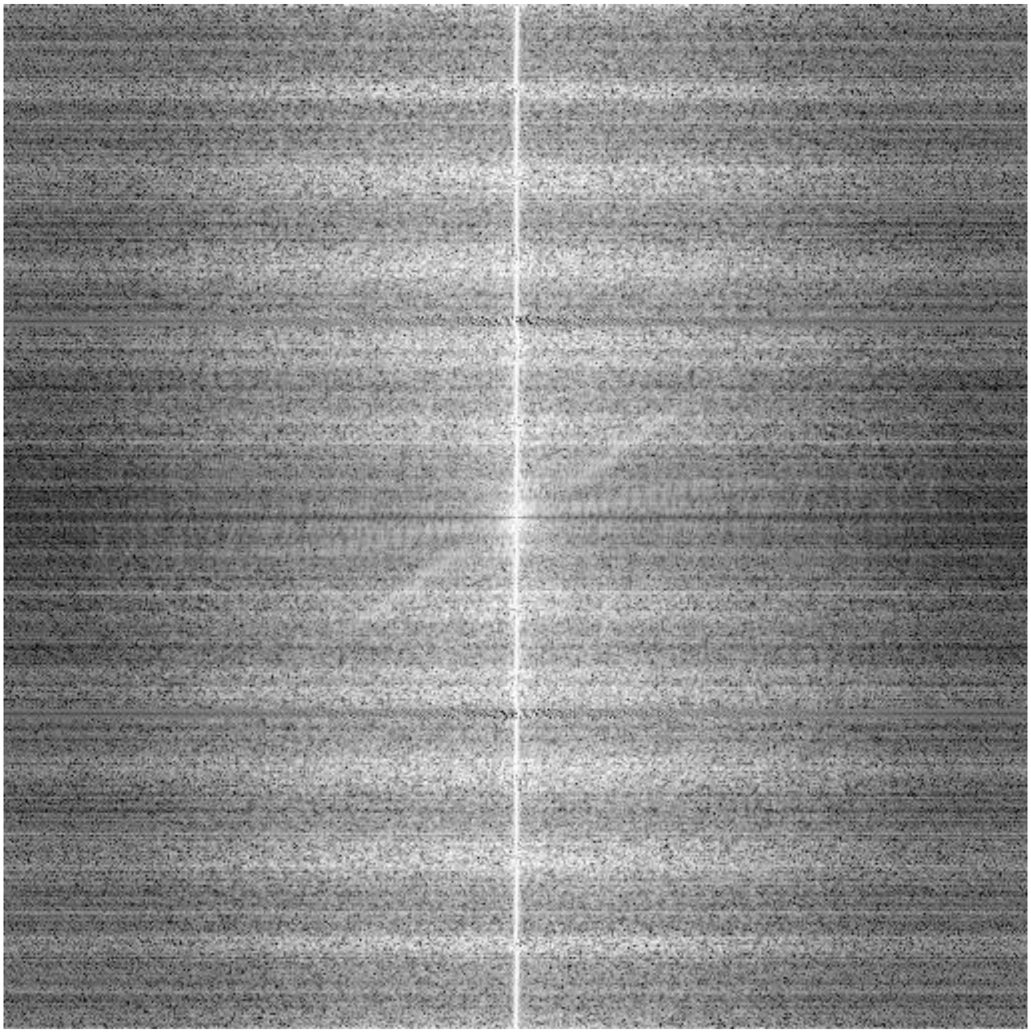
Following steps were performed to clean image:

1. Scan high frequency components on top-left portion of image and set them to 0
2. Scan high frequency components on bottom-right portion of image and set them to 0

Clean image:



Spectrogram of clean image



1.3 Watermarking

Magic parameters

N = 15, 20, 37, 60

l = 360

radius = 128

alpha = 5

correlation threshold = 0.20

Qualitative testing:

1. Same N used for different images
2. Different N used for same images
3. Different N used for different images

Quantitative testing:

- N ranges from 1 to 100 was used to evaluating watermarking process

Results

Image name	**N**	**Accuracy**	**N with false positives**	**Correlation for survived watermark**
clean.png	20	100%	None	0.515513
KameHouse.png	15	100%	None	0.369066
nature.png	37	100%	None	0.477078
mountain.png	60	100%	None	0.535025

More on results:

1. Watermarked images and spectrogram of watermarked images: master/part1/output/1.3
2. For quantitative report for different N, part1/results
3. Evaluation logs, part1/results/evalution_logs.log
4. Please refer readme file on github for instructions to run part 1 code

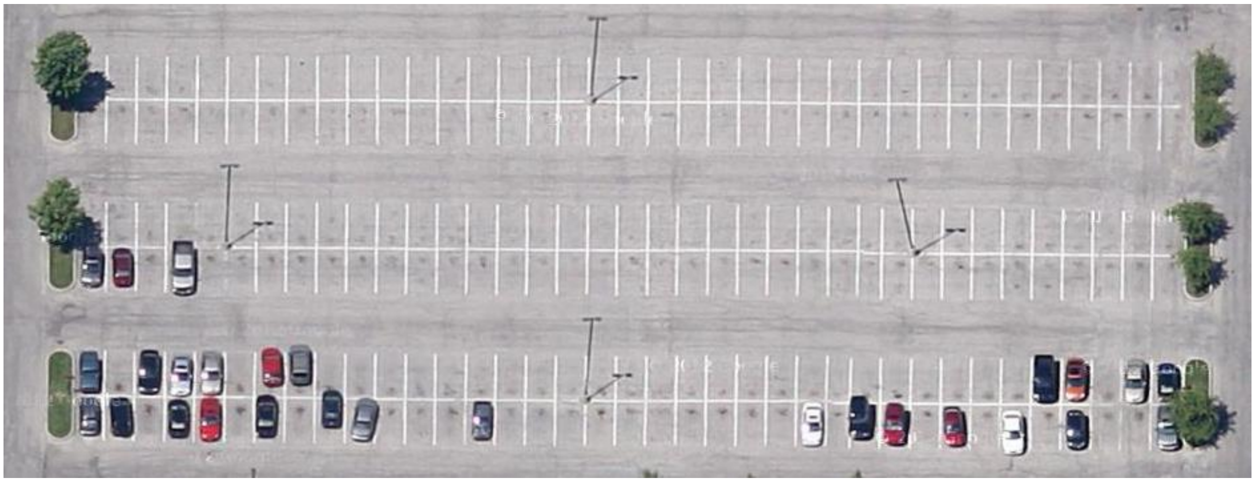
Issues faced:

1. If alpha is too low, not able to detect watermark
2. if alpha is high and radius is low then images got distorted

Part 2: Detecting objects

1. Image convolution:

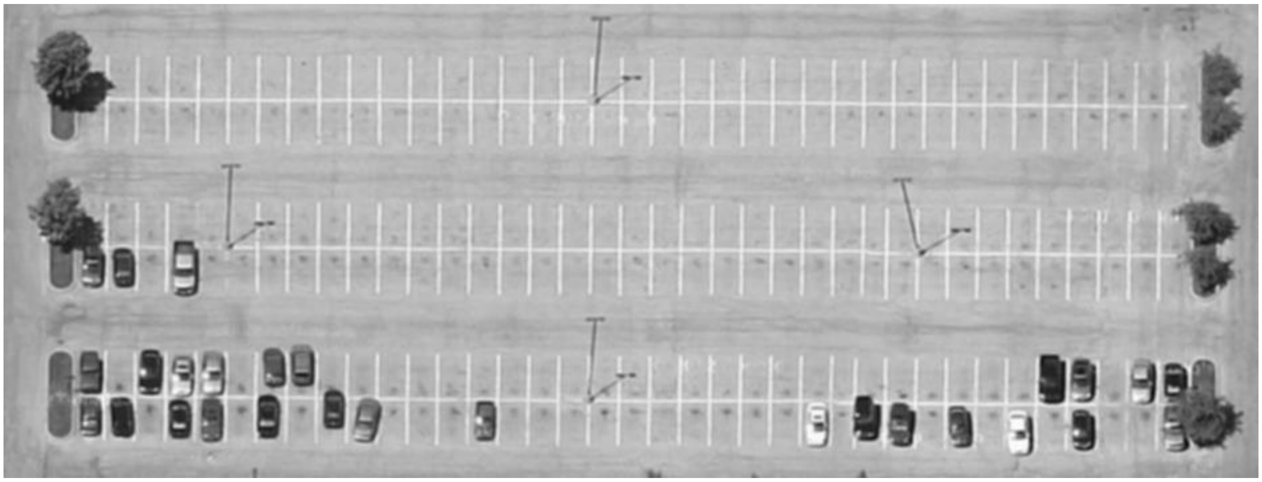
Input Image:



Output:

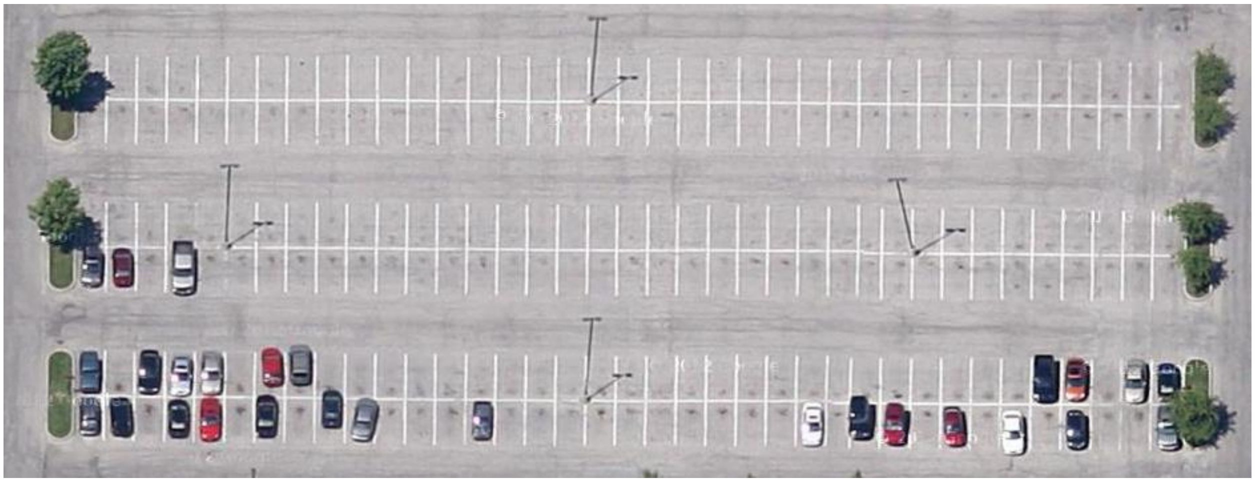
Convolved with a mean filter $\frac{1}{9}\,(1\;1\;1\;1\;1\;1\;1\;1\;1\;)$

Reflection boundary used to handle image boundaries.



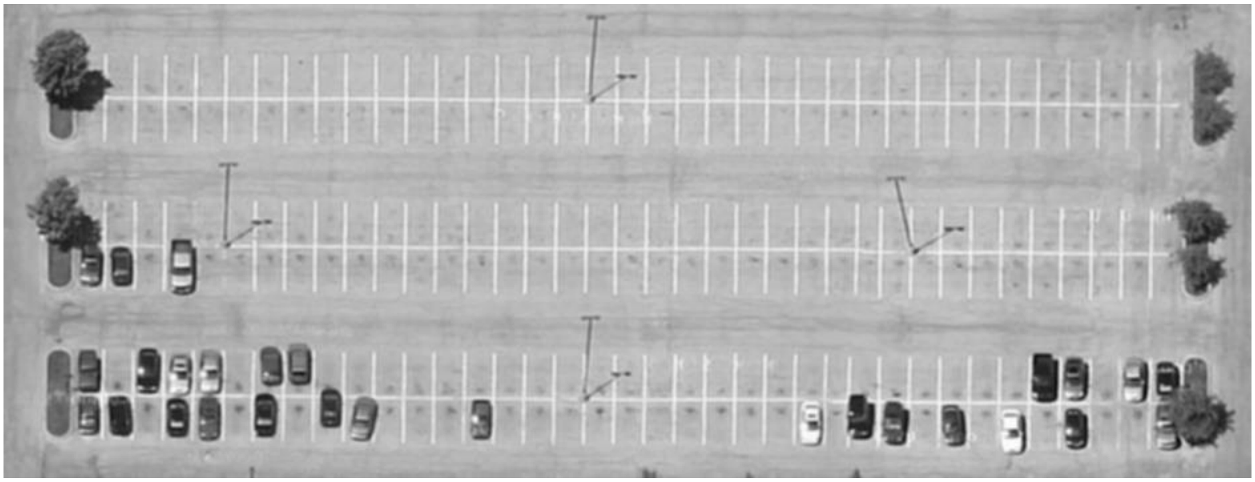
2. Separable Kernel:

Input image.



Output Image:

Separable kernel used : $[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}] * [\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$. Reflective Boundary method used for Image boundries.



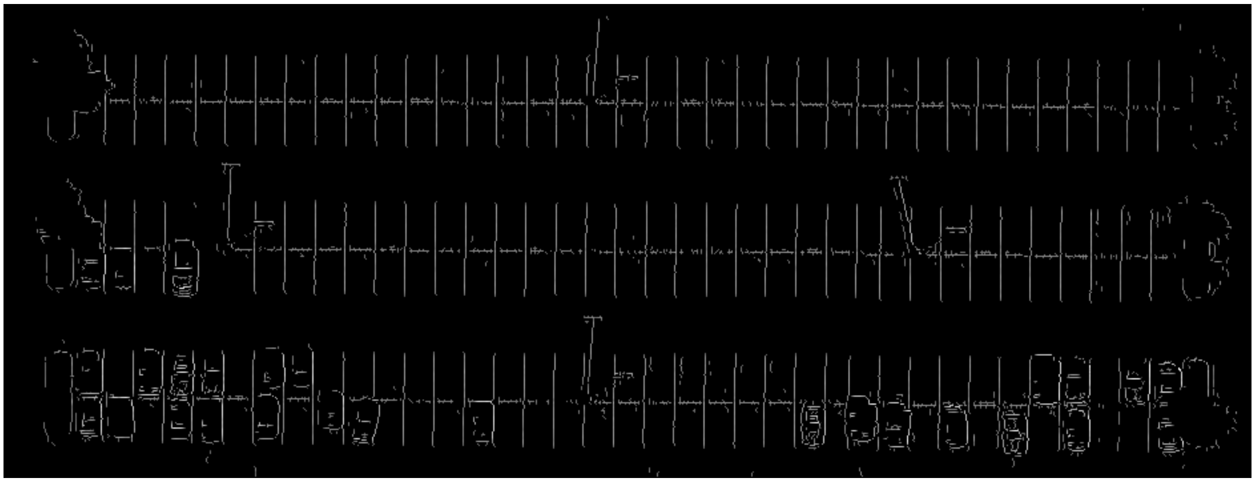
3. Sobel Operator:

$[-1\ 0\ +1\ -2\ 0\ +2\ -1\ 0\ +1]$ Gx $[+1\ +2\ +1\ 0\ 0\ 0\ -1\ -2\ -1]$ Gy

Edges detected in the Input Image SRSC.png :

Edges.png

The thresholding parameter in the Non-Maximum suppression step and lower and higher threshold value parameters during the hysteresis step can be altered as per the requirement for detecting edges.



4. Car detection:

Assumption: Width of the car between 15 to 30 pixels, Car is either parallel or perpendicular to x-y axis.

Future improvement: image can be rotated and zoomed and zoomed out to fit assumptions.

Works best when car is not very near to other car or big objects like tree.

following are the results for SRSC.png and Informatics.png

Also,

Output can be found in files as instructed:

1. edge.png
2. detected.png
3. detected.txt
4. binary_image.png (intermediate step for identifying car)