Assignment 1

Q1a

convolving f with g: $(I*F)[i,j] = \sum_{k,l} I[i-k, j-l]F[k,l]$

$k,l \in [-1,1]$

$$(I*F)[0,0] = \sum_{k,l} I[0-k, 0-l]F[k,l]$$
$$= I[1,1]F[-1,-1] + I[1,0]F[-1,0] + I[1,-1]F[-1,1]$$
$$+ I[0,1]F[0,-1] + I[0,0]F[0,0] + I[0,-1]F[0,1]$$
$$+ I[-1,1]F[1,-1] + I[-1,0]F[1,0] + I[-1,-1]F[1,1]$$
$$= (-2)(-1) + (1)(-1) + 0$$
$$+ 0 + (-1)(1) + 0$$
$$+ 0 + 0 + 0 = +2 - 1 - 1 = 0$$

$$(I*F)[0,1] = I[1,2]F[-1,-1] + I[1,1]F[-1,0] + I[1,0]F[-1,1]$$
$$+ I[0,2]F[0,-1] + I[0,1]F[0,0] + I[0,0]F[0,1]$$
$$+ I[-1,2]F[1,-1] + I[-1,1]F[1,0] + I[-1,0]F[1,1]$$
$$= (1)(-1) + (-2)(-1) + (1)(-1)$$
$$+ (2)(1) + 0 + (-1)(1)$$
$$+ 0 + 0 + 0 = \cancel{-1} + \cancel{2} \cancel{-1} + 2 - 1 = 1$$

$$(I*F)[0,2] = I[1,3]F[-1,-1] + I[1,2]F[-1,0] + I[1,1]F[-1,1]$$
$$+ I[0,3]F[0,-1] + I[0,2]F[0,0] + I[0,1]F[0,1]$$
$$+ I[-1,3]F[1,-1] + I[-1,2]F[1,0] + I[-1,1]F[1,1]$$
$$= 0 + (1)(-1) + (-2)(-1)$$
$$+ 0 + (2)(1) + 0$$
$$+ 0 + 0 + 0 = -1 + 2 + 2 = 3$$

$$(I*F)[1,0] = \sum_{k,l} I[0-k, 0-l]F[k,l]$$

$$= I[2,1]F[-1,-1] + I[2,0]F[-1,0] + I[2,-1]F[-1,1]$$
$$+ I[1,1]F[0,-1] + I[1,0]F[0,0] + I[1,-1]F[0,1]$$
$$+ I[0,1]F[1,-1] + I[0,0]F[1,0] + I[0,-1]F[1,1]$$
$$= 0 + 0 + 0$$
$$+ (-2)(1) + (1)(1) + 0$$
$$+ 0 + (-1)(0) + 0 = -2+1 = -1$$

$$(I*F)[1,1] = \sum_{k,l} I[0-k, 0-l]F[k,l]$$

$$= I[2,2]F[-1,-1] + I[2,1]F[-1,0] + I[2,0]F[-1,1]$$
$$+ I[1,2]F[0,-1] + I[1,1]F[0,0] + I[1,0]F[0,1]$$
$$+ I[0,2]F[1,-1] + I[0,1]F[1,0] + I[0,0]F[1,1] \Rightarrow 0 \text{ for this row}$$
$$= 0 + 0 + 0$$
$$+ (1)(1) + (-2)(1) + (1)(1) = 1-2+1 = 0$$
$$+ 0 + 0 + 0$$

$$(I*F)[1,2] = \sum_{k,l} I[0-k, 0-l]F[k,l]$$

$$= I[2,3]F[-1,-1] + I[2,2]F[-1,0] + I[2,1]F[-1,1]$$
$$+ I[1,3]F[0,-1] + I[1,2]F[0,0] + I[1,1]F[0,1]$$
$$+ I[0,3]F[1,-1] + I[0,2]F[1,0] + I[0,1]F[1,1]$$
$$= 0 + 0 + 0$$
$$+ 0 + (1)(1) + (-2)(1)$$
$$+ 0 + 0 + 0 = 1-2 = -1$$

$$(I*F) = \begin{bmatrix} 0 & 1 & 3 \\ -1 & 0 & -1 \end{bmatrix}$$

1b

1D conv: $(I * F_n)[i] = \sum_k I[i-k] F_n[k]$

$(I * F_1)[0,0] = \sum_k I[0-k, 0] F_1[k]$
$= I[1,0] F_1[-1] + I[0,0] F_1[0] + I[-1,0] F_1[1]$
$= (1)(-1) + (-1)(1) + 0 = -2$

$(I * F_1)[1,0] = I[2,0] F_1[-1] + I[1,0] F_1[0] + I[0,0] F_1[1]$
$= 0 + (1)(1) + (-1)(0) = 1$

$(I * F_1)[0,1] = I[1,1] F_1[-1] + I[0,1] F_1[0] + I[-1,1] F_1[1]$
$= (-2)(-1) + 0 + 0 = 2$

$(I * F_1)[1,1] = I[2,1] F_1[-1] + I[1,1] F_1[0] + I[0,1] F_1[1]$
$= 0 + (-2)(1) + 0 = -2$

$(I * F_1)[0,2] = I[1,2] F_1[-1] + I[0,2] F_1[0] + I[-1,2] F_1[1]$
$= (1)(-1) + (2)(1) + 0 = -1 + 2 = 1$

$(I * F_1)[1,2] = I[2,2] F_1[-1] + I[1,2] F_1[0] + I[0,2] F_1[1]$
$= 0 + (1)(1) + (2)(0) = 1$

$$I * F_1 = \begin{bmatrix} -2 & 2 & 1 \\ 1 & -2 & 1 \end{bmatrix} = I'$$

1D conv: $(I * F_n)[i] = \sum_k I[i-k] F_n[k]$

$(I' * F_2)[0,0] = \sum_\ell I'[0, j-\ell] F_2[\ell]$
$= I'[0,1] F_2[-1] + I'[0,0] F_2[0] + I'[0,-1] F_2[1]$
$= (2) + (-2) = 0$

$(I' * F_2)[0,1] = I'[0,2] + I'[0,1] + I'[0,0] = 1 + 2 - 2 = 1$
$(I' * F_2)[0,2] = I'[0,3] + I'[0,2] + I'[0,1] = 0 + 1 + 2 = 3$
$(I' * F_2)[1,0] = I'[1,1] + I'[1,0] + I'[1,-1] = -2 + 1 = -1$
$(I' * F_2)[1,1] = I'[1,2] + I'[1,1] + I'[1,0] = 1 - 2 + 1 = 0$
$(I' * F_2)[1,2] = I'[1,3] + I'[1,2] + I'[1,1] = 0 + 1 - 2 = -1$

$$(I * F_1) * F_2 = \begin{bmatrix} 0 & 1 & 3 \\ -1 & 0 & -1 \end{bmatrix}$$

1c

1D conv: $(I * F_n)[i] = \sum_k I[i-k] F_n[k]$

$$(I*F_n)[i,j] = \sum_k I[i-k, j] F_n[k]$$

Prove for $F = F_1 F_2$, $I * F = (I * F_1) * F_2$. Note $F[k, l] = F_1[k] F_2[l]$ by definition. Also note that $(I * F_1) * F_2 = \sum_l \left(\left(\sum_k I[i-k, j] F_1[k]\right)[j-l]\right) F_2[l]$ by our 1D convolution definition. Then we have:

$$(I*F)[i,j] = \sum_k \sum_l I[i-k, j-l] F[k, l]$$

$$= \sum_k \sum_l I[i-k, j-l] (F_1[k])(F_2[l])$$

$$= \sum_l F_2[l] \underbrace{\sum_k I[i-k, j-l] F_1[k]}_{=(I * F_1)[i, j-l]}$$

$= (I * F_1)[i, j-l]$, for a given l. We will next take the summation

$$= \sum_l F_2[l] \left[(I * F_1)[i, j-l]\right]$$

$$= (I * F_1) * F_2.$$ ✓

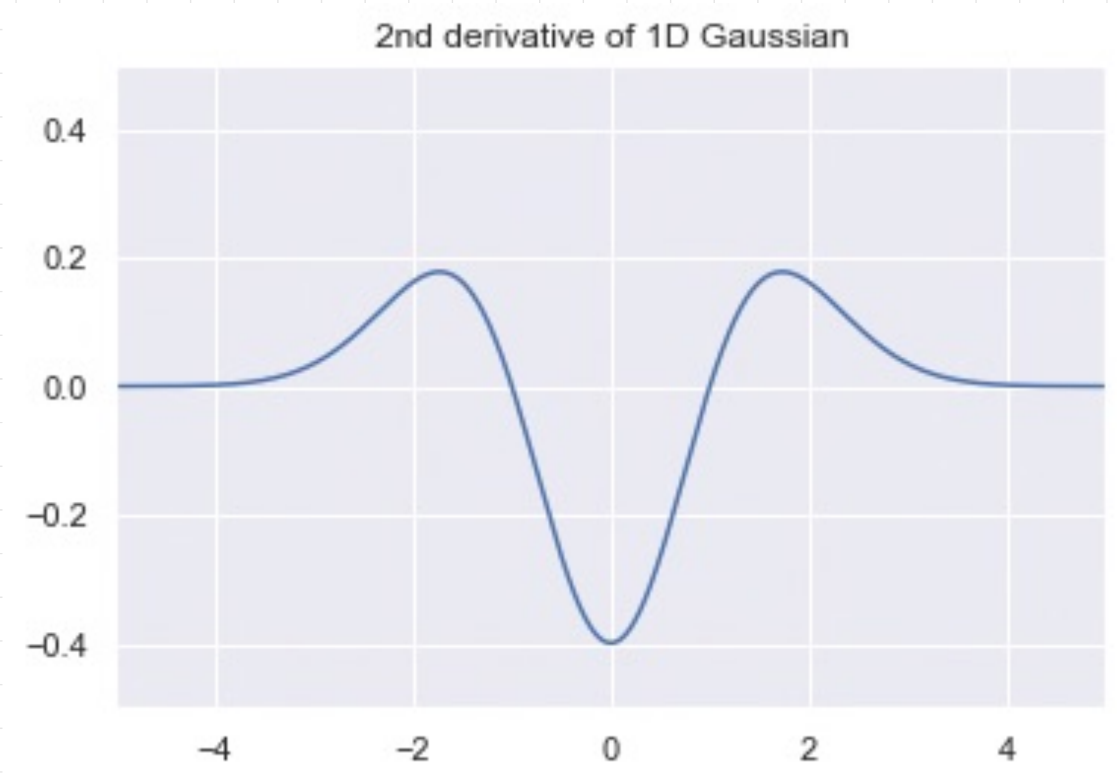
2a) $g_\sigma(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$

$$\frac{d}{dx} g_\sigma(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\left(\frac{-2x}{2\sigma^2}\right) = -\frac{x}{\sqrt{2\pi}\,\sigma^3} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

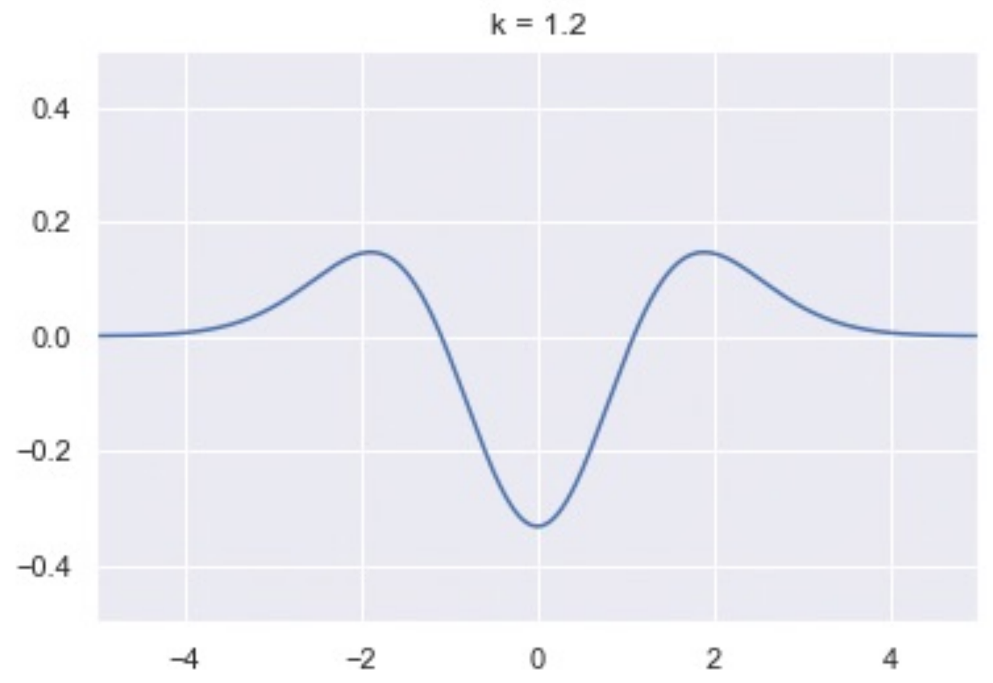
$$\frac{d^2}{dx^2} g_\sigma(x) = \frac{-1}{\sqrt{2\pi}\,\sigma^3} \exp\left(-\frac{x^2}{2\sigma^2}\right) - \frac{x}{\sqrt{2\pi}\,\sigma^3} \cdot \left(\exp\left(-\frac{x^2}{2\sigma^2}\right) \frac{-\cancel{2}x}{\cancel{2}\sigma^2}\right)$$

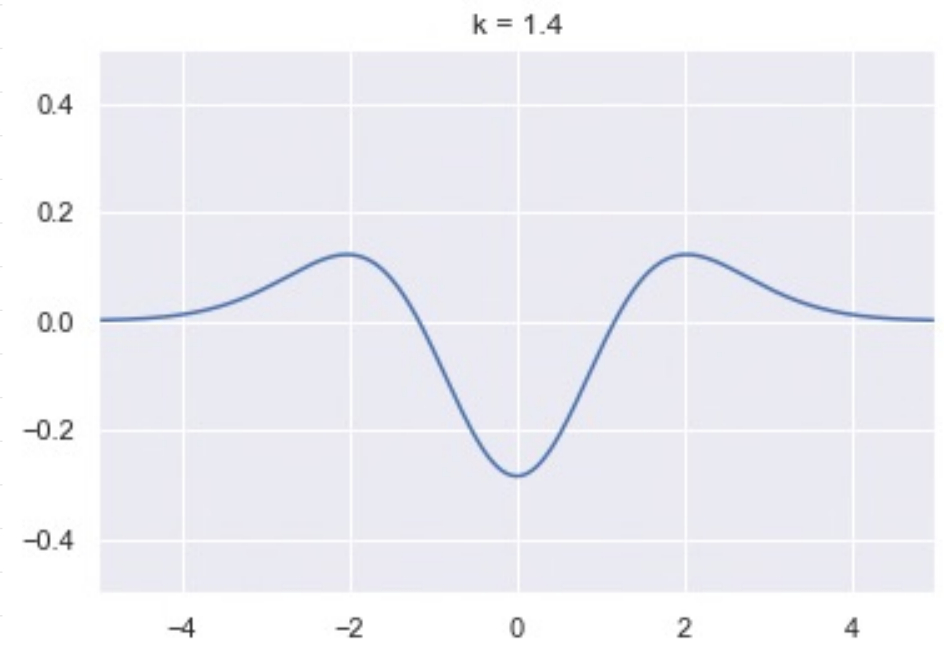
$$= \exp\left(\frac{-x^2}{2\sigma^2}\right)\left[\frac{x^2}{\sqrt{2\pi}\,\sigma^5} - \frac{1}{\sqrt{2\pi}\,\sigma^3}\right]$$

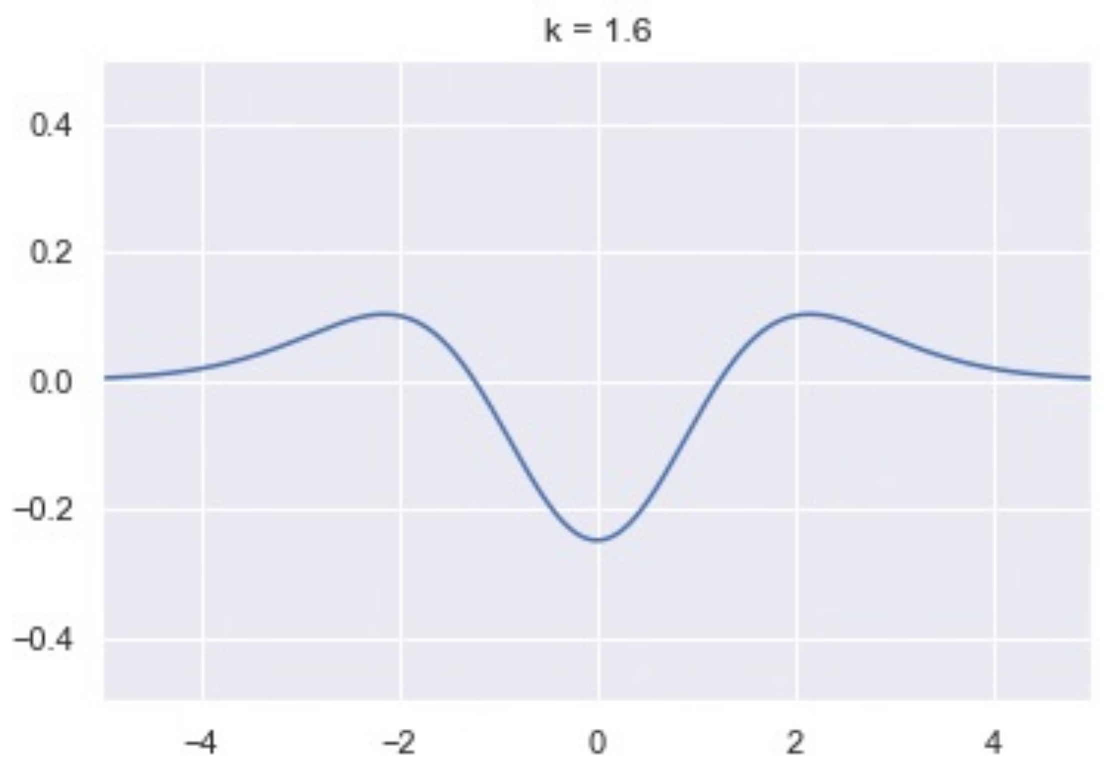
$$= \exp\left(-\frac{x^2}{2\sigma^2}\right)\left[\frac{x^2 - \sigma^2}{\sqrt{2\pi}\,\sigma^5}\right]$$

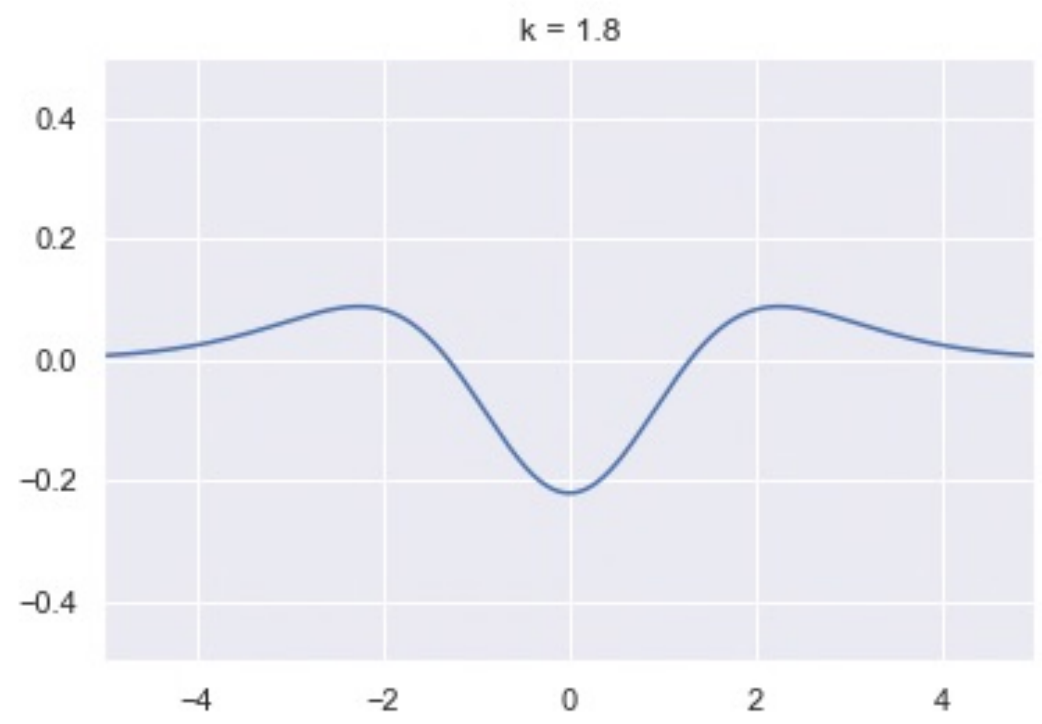


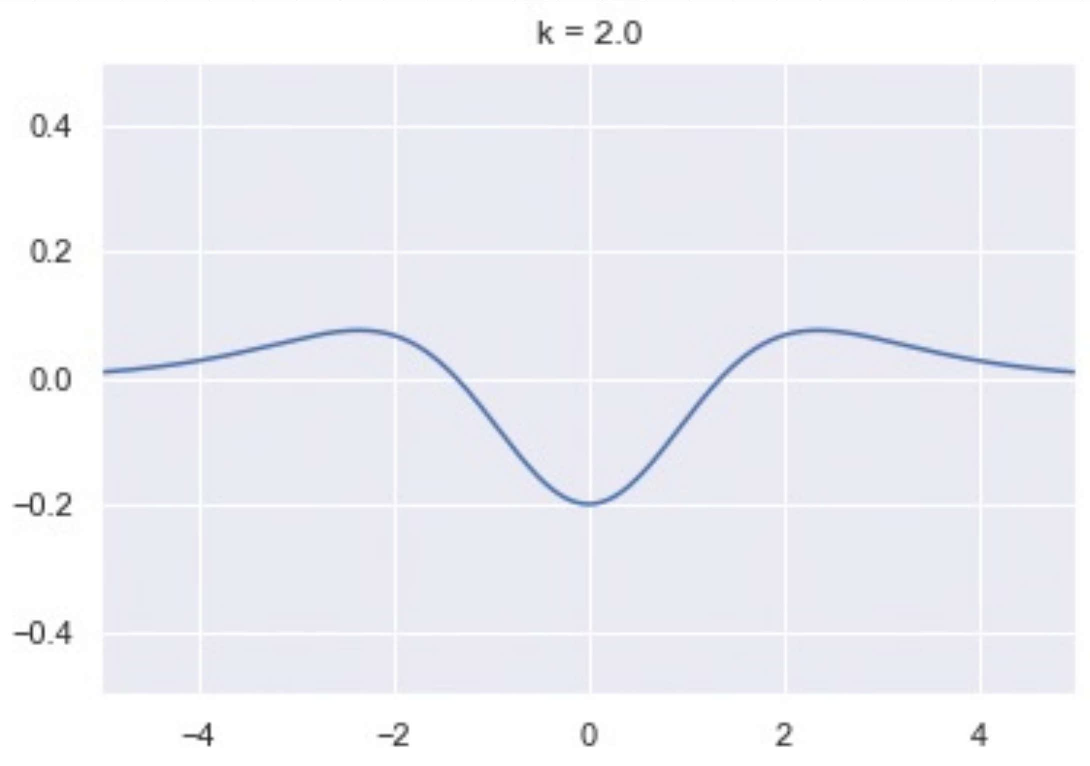
2b) The value of k that gives the best approximation to the 2nd derivative WRT x is k=1.2.











2c) The difference of Gaussian, like the Laplacian, will respond maximally to circular structures. Particularly, since the DoG approximates the Laplacian, it will also respond approximally maximally to circles whose radii are roughly $r = \sigma\sqrt{2}$. This is because those circles' edges will roughly line up with the zero's of the DoG.

Question 3

Fit-Affine Panorama - Hand-defined Correspondences

Fit-Affine Panorama - Automatic Correspondences

Question 4

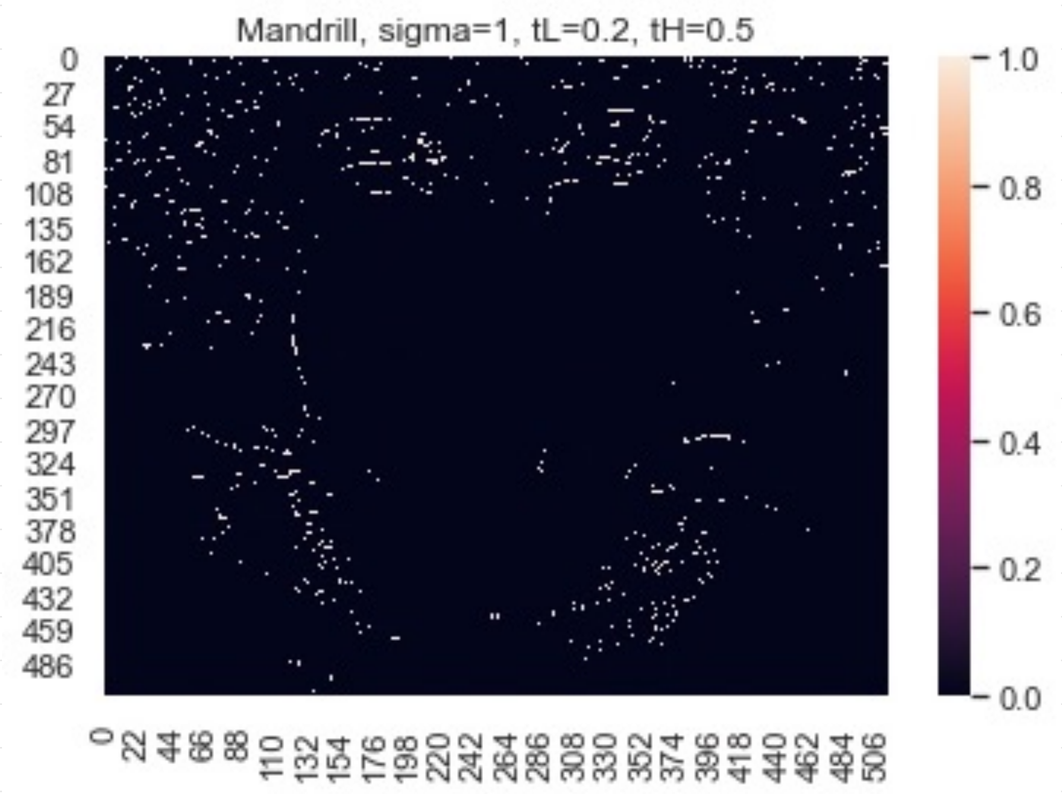
RANSAC Robust Panorama

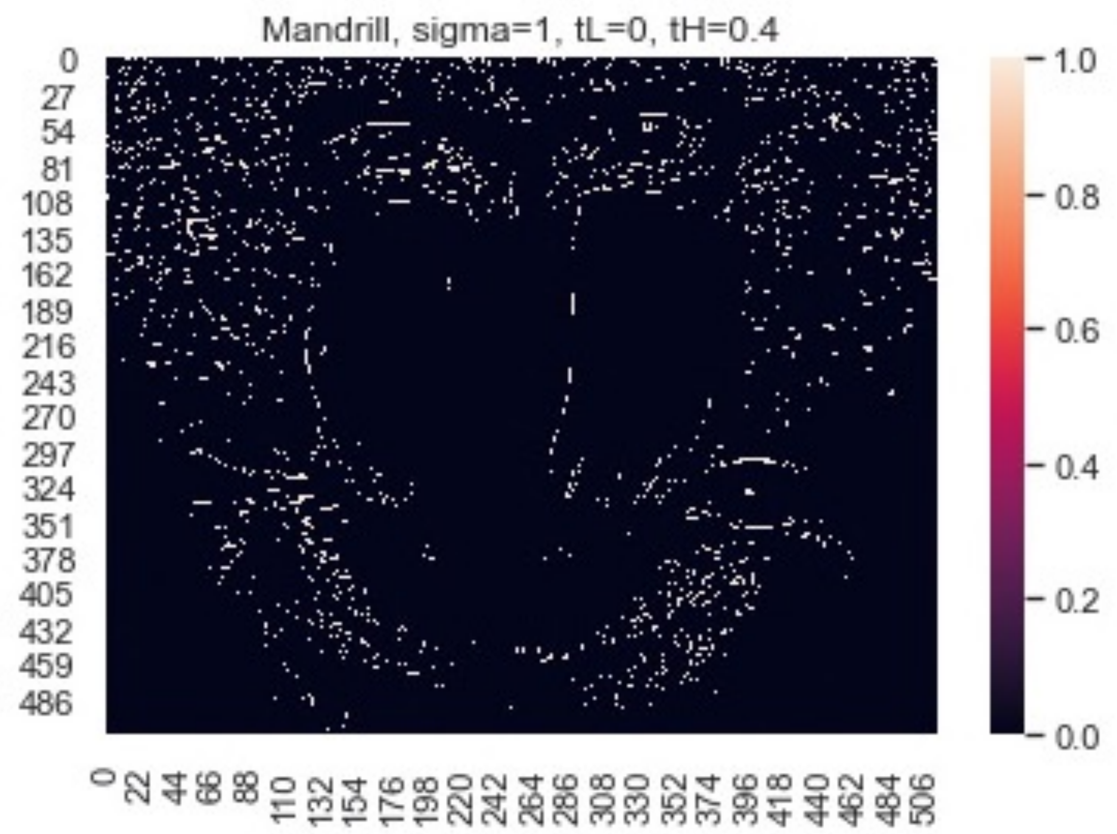
Question 5

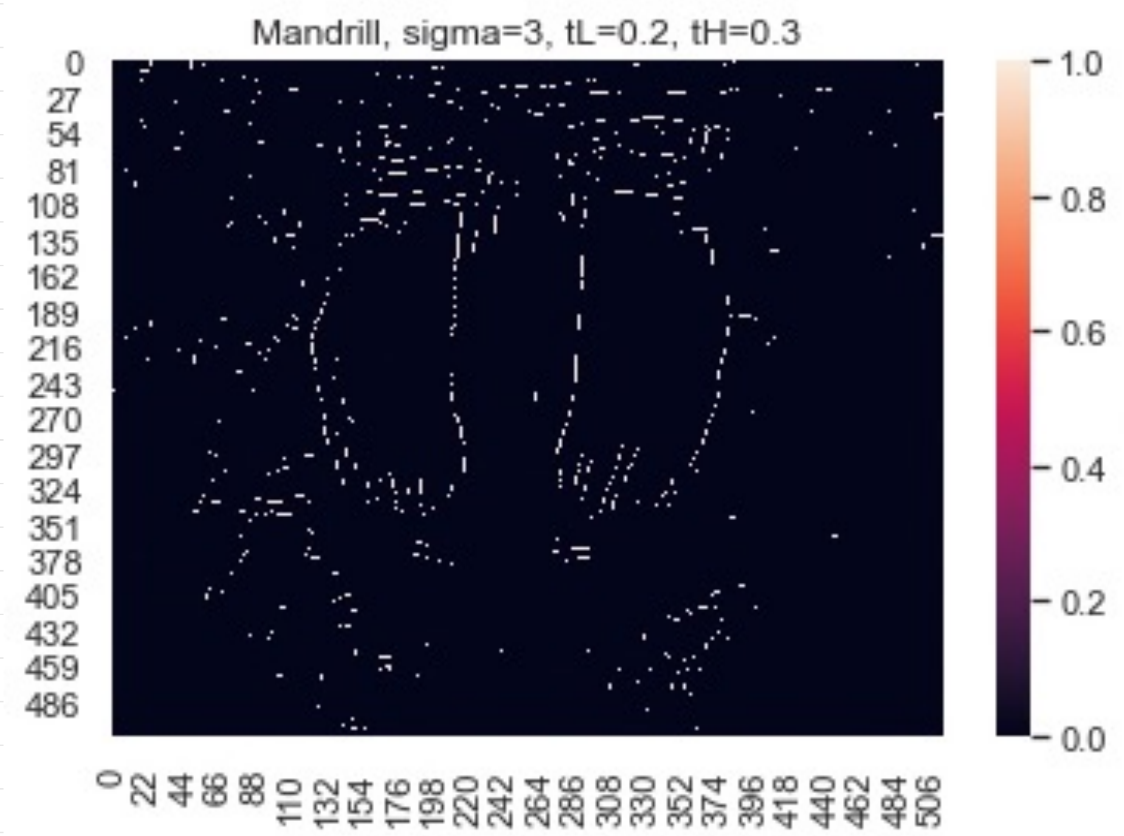
Without linear blend, the panorama contains artifacts like edges of the other image that were not properly resampled during image warping. The linear blending removes some of these artifacts, especially the edges of the image boundaries and the fountain overlap area, creating a better stitched result.

Question 6

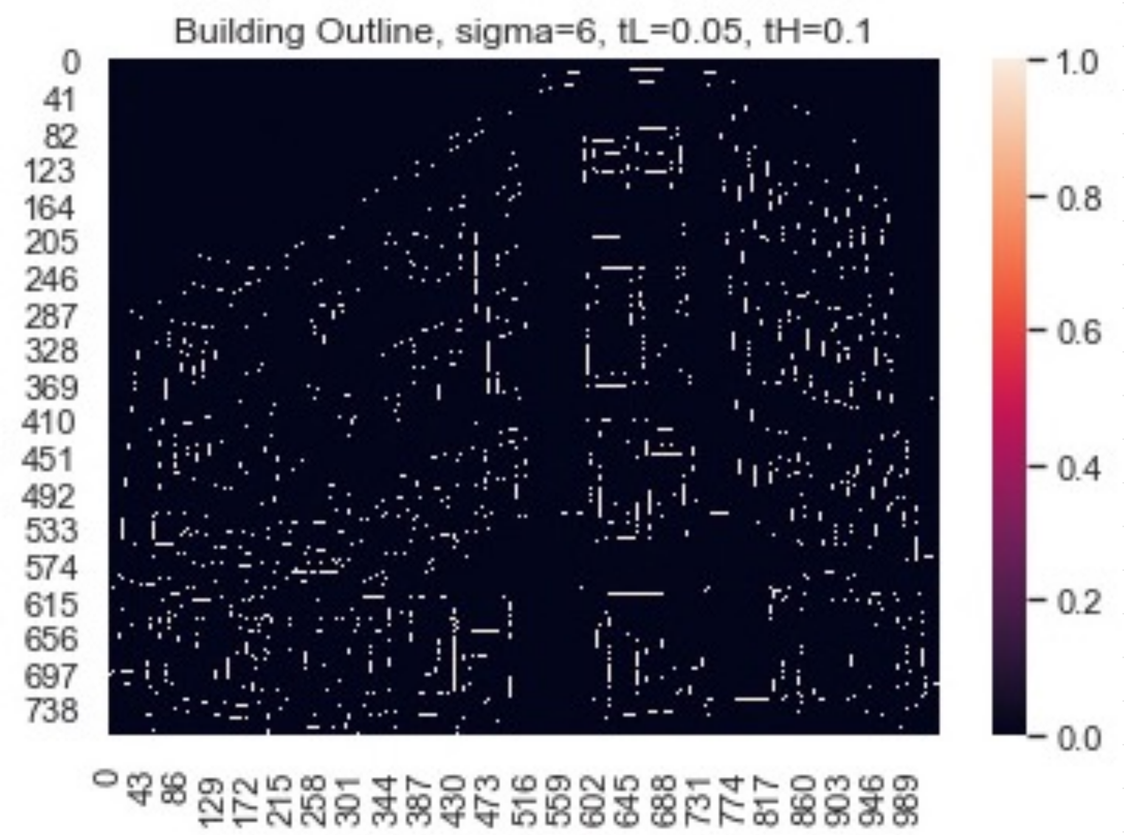
6a)

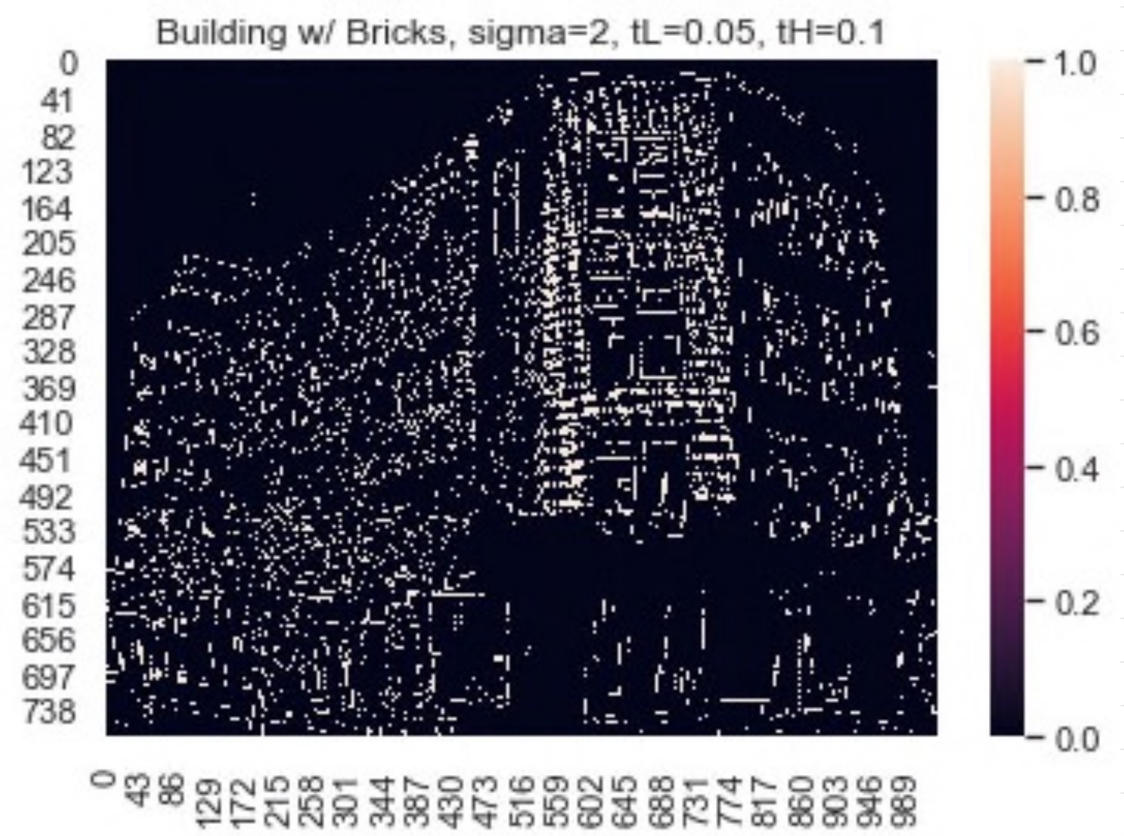




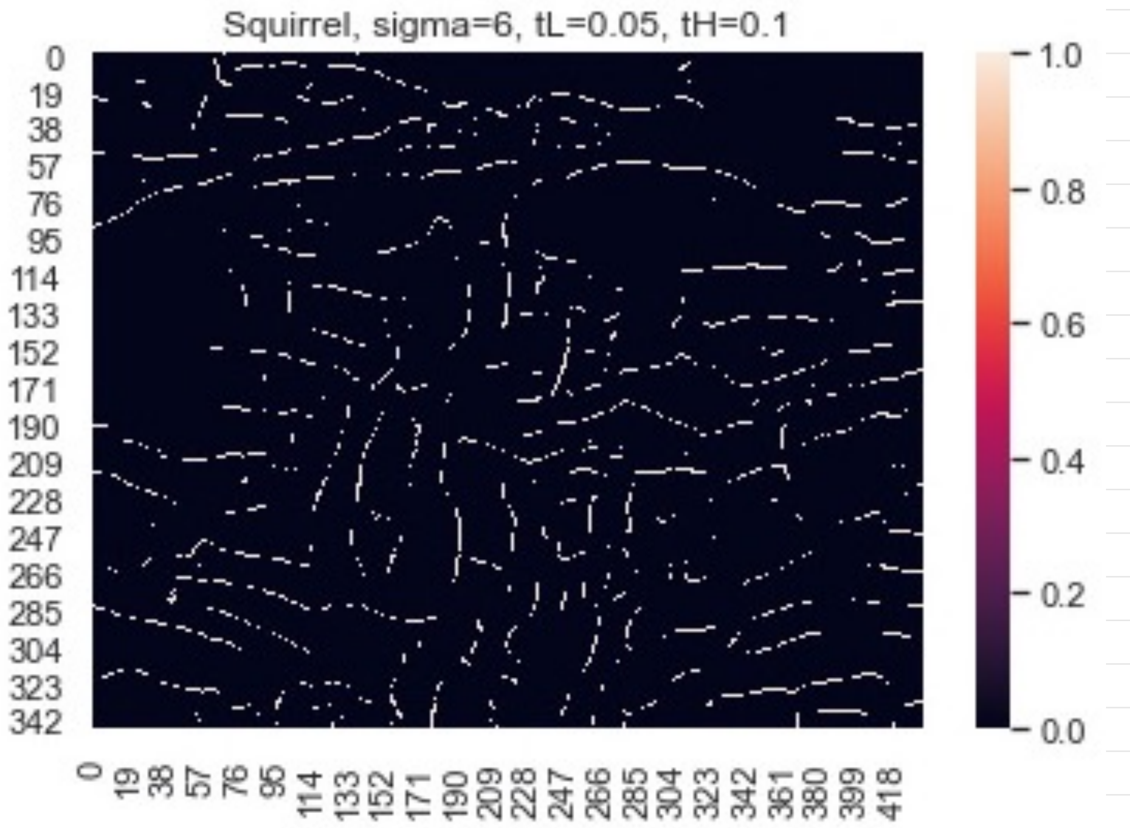


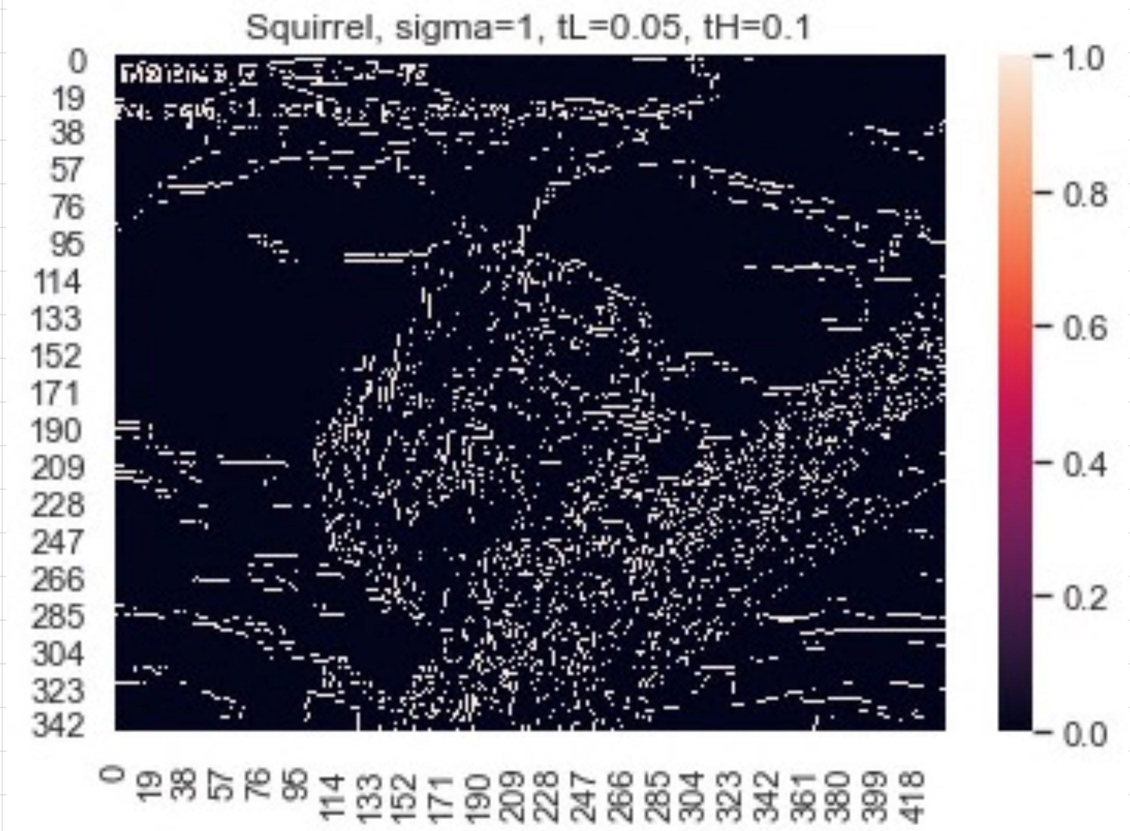
6b)

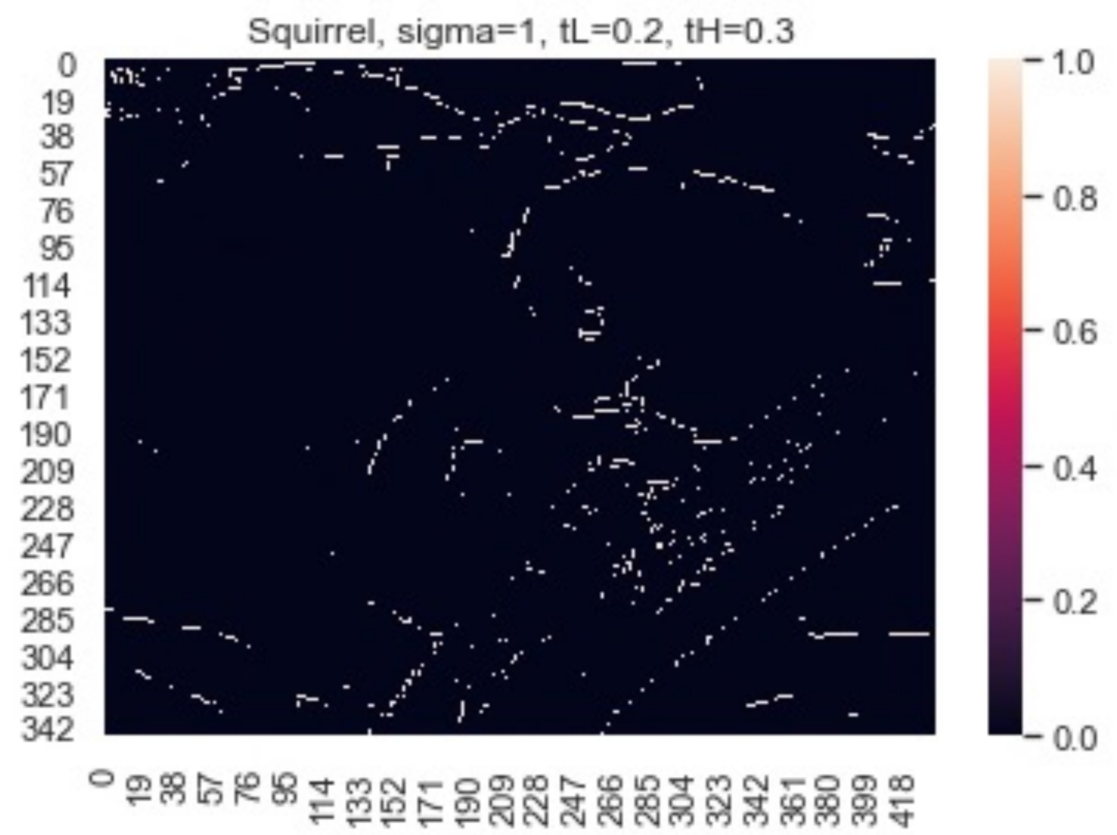

Building Outline, sigma=6, tL=0.05, tH=0.1
0
41
82
123
164
205
246
287
328
369
410
451
492
533
574
615
656
697
738
0
43
86
129
172
215
258
301
344
387
430
473
516
559
602
645
688
731
774
817
860
903
946
989
1.0
0.8
0.6
0.4
0.2
0.0


Building w/ Bricks, sigma=2, tL=0.05, tH=0.1
0
41
82
123
164
205
246
287
328
369
410
451
492
533
574
615
656
697
738
0
43
86
129
172
215
258
301
344
387
430
473
516
559
602
645
688
731
774
817
860
903
946
989
1.0
0.8
0.6
0.4
0.2
0.0

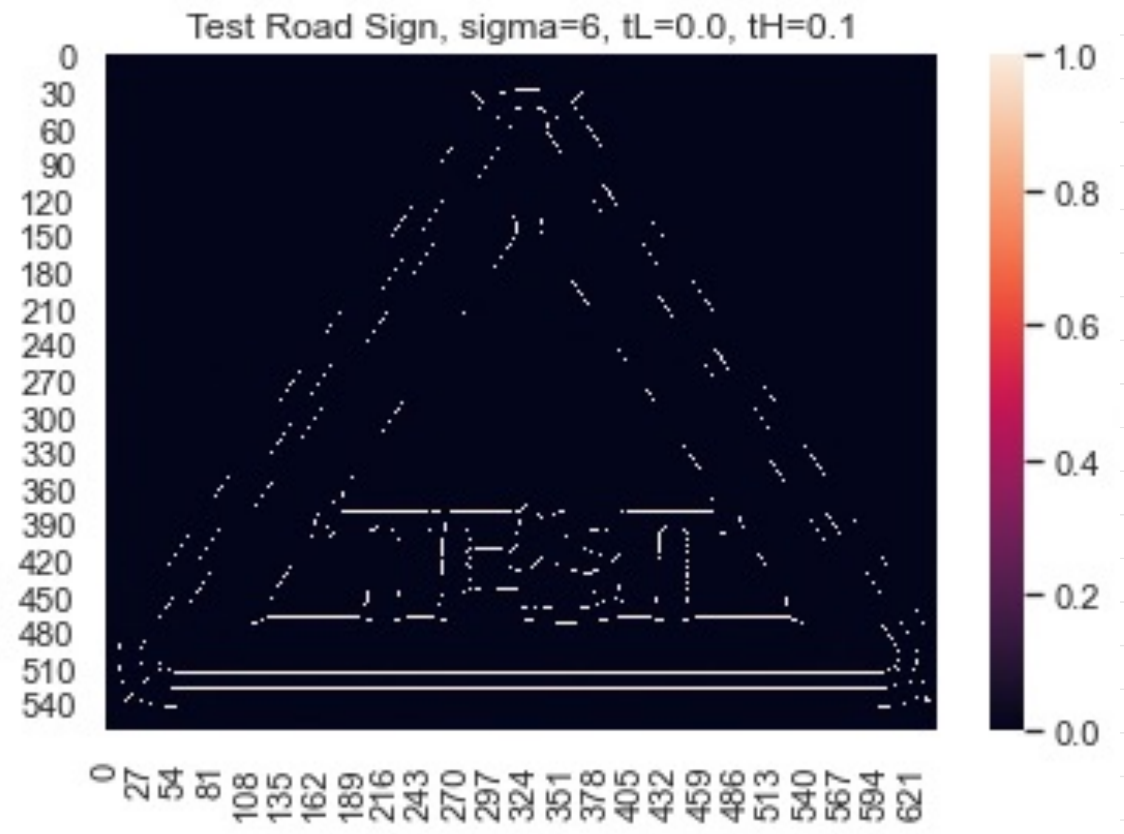
6c)


Squirrel, sigma=6, tL=0.05, tH=0.1

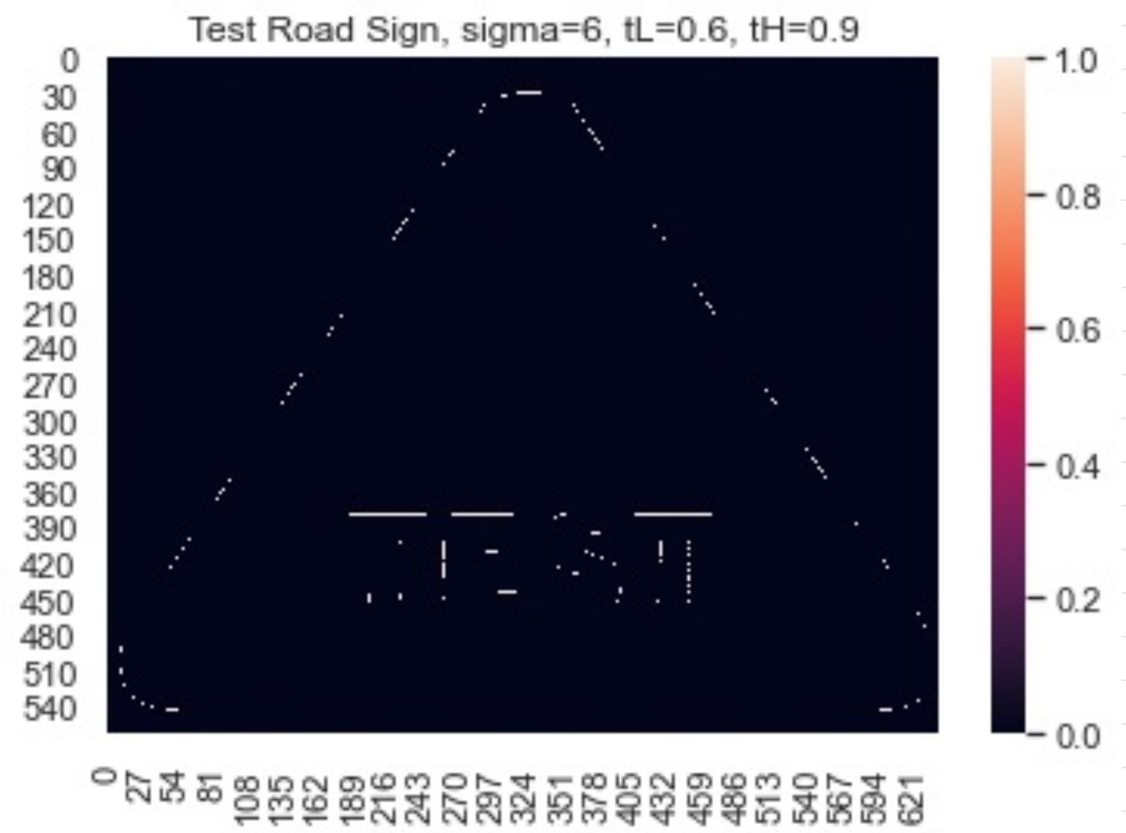

Squirrel, sigma=1, tL=0.05, tH=0.1


Squirrel, sigma=1, tL=0.2, tH=0.3

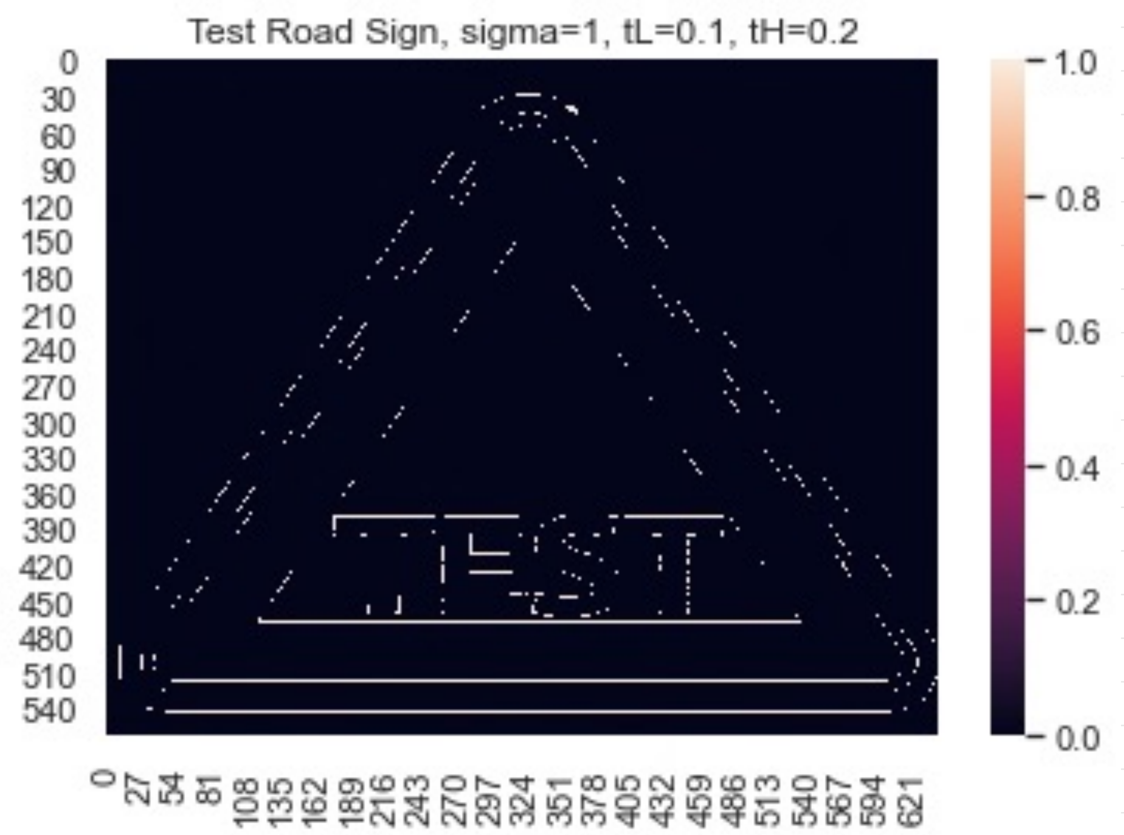
6c)



Test Road Sign, sigma=6, tL=0.0, tH=0.1
0
30
60
90
120
150
180
210
240
270
300
330
360
390
420
450
480
510
540
0
27
54
81
108
135
162
189
216
243
270
297
324
351
378
405
432
459
486
513
540
567
594
621
1.0
0.8
0.6
0.4
0.2
0.0



Test Road Sign, sigma=6, tL=0.6, tH=0.9
0
30
60
90
120
150
180
210
240
270
300
330
360
390
420
450
480
510
540
0
27
54
81
108
135
162
189
216
243
270
297
324
351
378
405
432
459
486
513
540
567
594
621
1.0
0.8
0.6
0.4
0.2
0.0



Test Road Sign, sigma=1, tL=0.1, tH=0.2
0
30
60
90
120
150
180
210
240
270
300
330
360
390
420
450
480
510
540
0
27
54
81
108
135
162
189
216
243
270
297
324
351
378
405
432
459
486
513
540
567
594
621
1.0
0.8
0.6
0.4
0.2
0.0

6d) We used heatmaps to visualize each step of the canny algorithm, and you can see this at the bottom of hw1.ipynb, where we also ran the canny.py code. One bug this helped fix was our conversion of the image to floating point. We noticed the edge orientation angles appeared incorrect in our heatmap of D, and traced this back to the floating point conversion issue. We also used these visuals to understand if our thresholding was working properly and to debug directional issues in the NMS step, where we could literally see the edges thin from the original F matrix. We also used heatmaps to verify that our F_x and F_y were behaving as expected, helping us to realize that our computation of these arrays was not sufficiently blurring in 2D. Heatmaps were also used to generate final visuals of the edgemaps for Questions 6a–c.