ex1_204502926_305498099
Ofir Nesher & Yossi Gavriel

A.

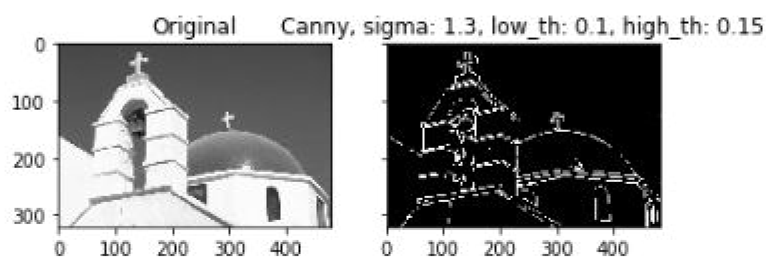
1. convolutionMask A1:
   [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]

2. convolutionMask A2:
```
[[ 0.  0.  0.  0.  0.]
 [ 0. -1.  1.  1.  0.]
 [ 0. -1. -1.  1.  0.]
 [ 0. -1. -1.  1.  0.]
 [ 0.  0.  0.  0.  0.]]
```

3. convolutionMask A3:
```
[[ 0.  0.  0.  0.  0.]
 [ 0. -1.  1.  1.  0.]
 [ 0. -1.  0.  1.  0.]
 [ 0. -1. -1.  1.  0.]
 [ 0.  0.  0.  0.  0.]]
```

B.

2. “display the image you chose, its edges, and the parameters you used”
We tested our canny implementation with the following parameters:
Image name: ‘Church.jpg’
Sigma: 1.3, 2, and 3.
Low threshold: 0.1, 0.15 and 0.2.
High threshold: 0.15, 0.3 and 0.5
We have chosen the following parameters which perform better results:

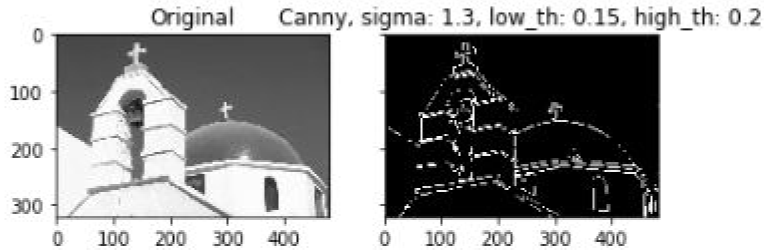


3.
Sigma: The larger the sigma, the larger the values in the Gaussian-filter which are further from the center, so more pixels in the image are considered when this filter is used as the convolution-mask on the image.
Hence, the larger the sigma, the more visible the blur on the image is!
(The mask size should be a function of σ)

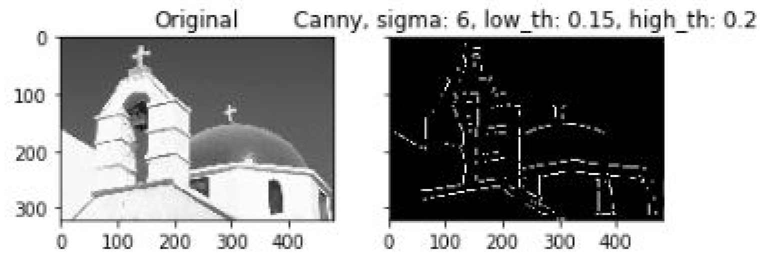
The detected edges also differ depending on the sigma's value:

- Large values result in the detection of stronger, "sharper" edges
- Small values - finer edges and more noise left in the convolved image

Example (same high and low thresholds but sigma is 1.3 or 6):

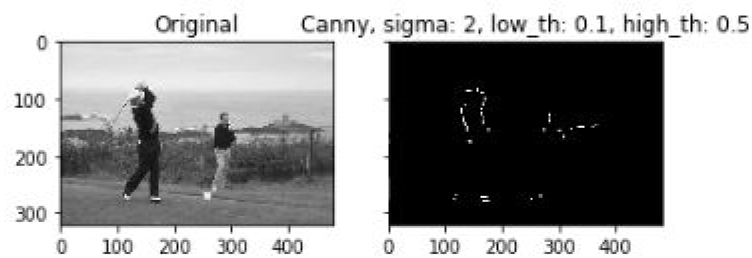


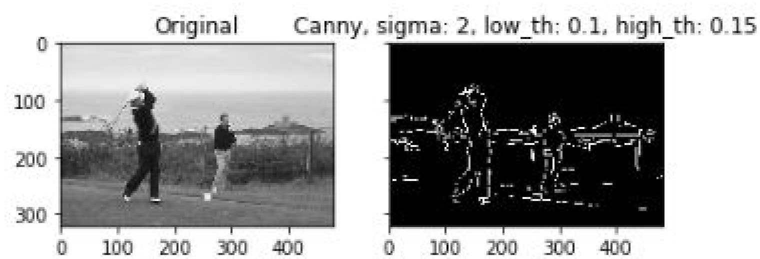
Higher sigma:



Higher threshold: Used to filter-out "non-strong"/"non-sharp" from the final resulting image. In general, the higher this value is - the fewer components (details) will remain in the resulting image (A threshold set too high can miss important information, though - False negatives).

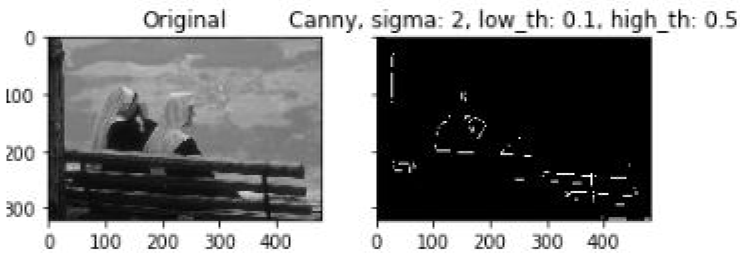
<u>Example</u>: Same sigma and low-threshold but the high-threshold is 0.15 or 0.5

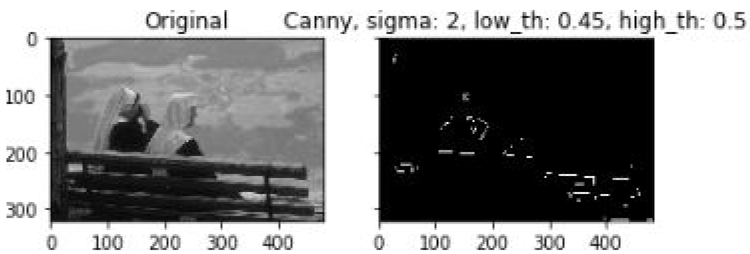




Lower threshold: Used to mark "weak" edges in the given image.

Usually, a weak edge pixel caused by true edges will be **connected** to a strong edge's pixel, while noise components ("fake" edges) are unconnected.
A threshold too low will falsely identify irrelevant information (noise) as important edges in the image, hence, more false-positives relevant details.
Example: Same sigma and high-threshold but the low-threshold is 0.1 or 0.45:





C.

1. a. The choice of Canny's parameters that will cause P to be high is when the thresholds' values are **large**.
   This will result in fewer details overall in the final resulting image, and thus fewer elements in E (only the really strong pixels that were detected as edges will remain).
   Small |E| → intersection with GT == ~ |E| → larger $P = \frac{|intersection|}{|E|}$

   b. The choice of Canny's parameters that will cause R to be high is when the thresholds' values are **small**.
   This will result in more details overall in the final resulting image, and thus more elements in E.
   Large |E| → intersection with GT == ~|GT| → larger $R = \frac{|intersection|}{|GT|}$

- The larger the sigma (with the same thresholds) - the lower the values we get for P & R.

2. Function in ex1.py

3. Function in ex1.py

4. We considered the following values for Canny's 3 parameters:

Sigma values = [1.1, 2, 4]
Low threshold values = [0.1, 0.15, 0.2]
High threshold values = [0.25, 0.3, 0.6]

The best values for each parameter and their results:

Img: **Church_GT**.bmp, sigma: 1.1, low_th: 0.1, high_th: 0.25 -
P: 0.3359960973230075, **R**: 0.5503945659774249, **F**: 0.41726618705035967

Img: **Golf_GT**.bmp, sigma: 1.1, low_th: 0.1, high_th: 0.25 -
P: 0.36788792017336086, **R**: 0.5560210221646736, **F**: 0.44279995147397794

Img: **Nuns_GT**.bmp, sigma: 1.1, low_th: 0.1, high_th: 0.25 -
P: 0.3114554936588835, **R**: 0.591935574702934, **F**: 0.40815454810212604