Programming Assignment - 1 (Submission)

Computer Vision CAP - 5415, Fall 2023

Name: Rajarshi Bhattacharya **Student ID**: 5568759 **Email**: ra075955@ucf.edu

Section A: Canny Edge Filtering Mechanism

1. Grayscale Image is read and normalized as a 2-D Floating Point matrix between [0,1]. Let's denote the image as I.
2. A 1-D Gaussian Mask is designed considering size 'n' where n is an odd integer. The variance 'σ' is kept variable. Let's call the mask G.
3. G' is computed as the derivative of the Gaussian mask.
4. Convolution of I with G along the x-direction and y-direction yields I_x and I_y,respectively (*written as Ix and Iy in code*).
5. Convolution of I_x with G' along x-direction yields I_x' (*written as Ixx in code*). Similarly, the y-counterpart for this step yields I_y'(written as Iyy in code).
6. We now polarize the obtained maps I_x' and I_y'. We define $M = \text{sqrt}(I_x'^2 + I_y'^2)$, where sqrt() is the square root function. We also define $\theta = \arctan(I_y' , I_x')$. M is normalized upon computation of the square-root.
7. Non-maximum suppression is performed on the pixels. We discretize θ to nearest multiples of 45 degrees and suppress the pixels whose neighborhood pixels do not have maximal values in the orientation of the pixel. This operation localizes the edges better as the edges appear non-localized in M. Let's call the output M_{nms} (*M_nms in code*).
8. Hysteresis thresholding is performed on M_{nms}. We choose 2 thresholds and split the range of the matrix in 3 distinct zones, low, mid and high. High pixels are set to 255, low pixels are zeroed out and mid values are set to some user defined intensity value. The output is M_{thresh} (*M_thresh in code*).
9. Finally, we connect the edge clusters together using the Connected Components function from OpenCV as allowed by the instruction sheet. The final output is M_{cc} (*M_cc in code*).

Section B: Code Description

Extensive documentation has been provided and the code has also been provided on Github. There are two files: `utils.py` and `main.py`. Images used for evaluation are provided in `/assets` directory and all outputs provided are in `/outputs` directory. 'Utils' contains the individual components described in Section A and 'main' forms the output console. Library requirements for the code base has been provided in `reqs.txt`.

Section C: Experimentation

Parameters:

- Run Name: test_1_1
- Input Image: `assets/119082.jpg`
- Mask Size: 5
- σ: 0.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

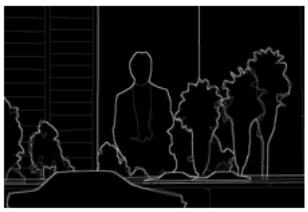
Image

Ix

Iy

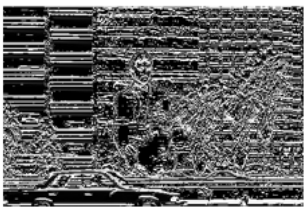
M

M_thresh



Ground Truth

Ixx



Iyy

M_nms

M_cc

Parameters:

- Run Name: test_1_2
- Input Image: `assets/119082.jpg`
- Mask Size: 5
- σ: 1.0
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

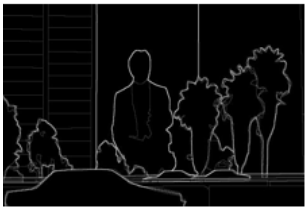
Image

Ix

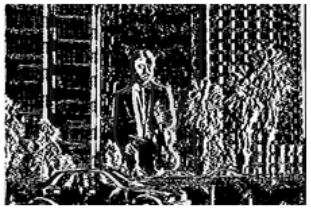
Iy

M

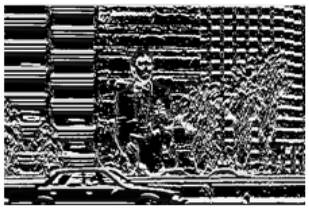
M_thresh



Ground Truth



Ixx



Iyy

M_nms

M_cc

Parameters:

- Run Name: test_1_3
- Input Image: `assets/119082.jpg`
- Mask Size: 5
- σ: 1.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

Image

Ix

Iy

M

M_thresh

Ground Truth

Ixx

Iyy

M_nms

M_cc

Parameters:

- Run Name: test_2_1
- Input Image: `assets/58060.jpg`
- Mask Size: 5
- σ: 0.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

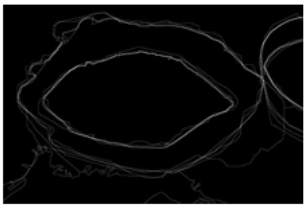
Image

Ix

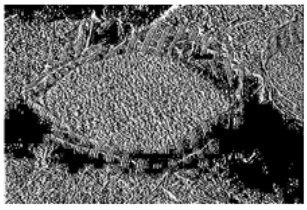
Iy

M

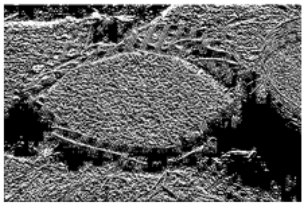
M_thresh



Ground Truth

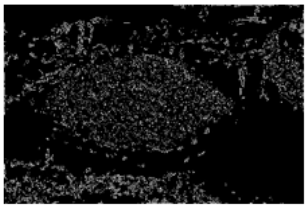


Ixx



Iyy

M_nms



M_cc

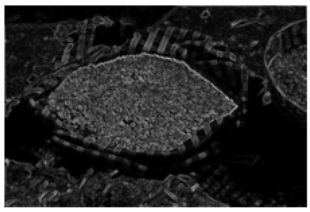
Parameters:

- Run Name: test_2_2
- Input Image: `assets/58060.jpg`
- Mask Size: 5
- σ: 1.0
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

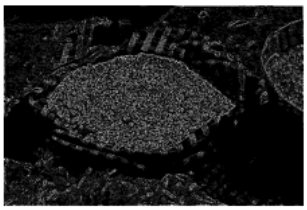
Image

Ix

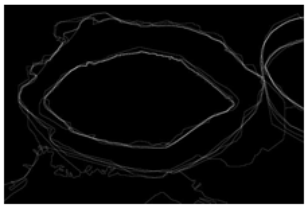
Iy



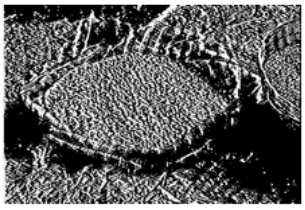
M



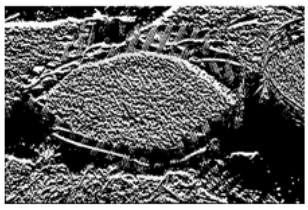
M_thresh



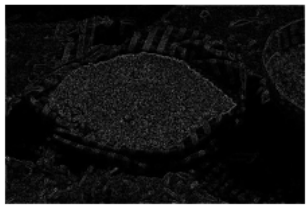
Ground Truth



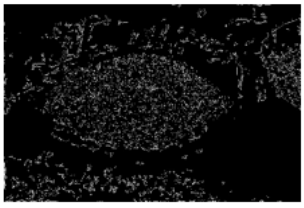
Ixx



Iyy



M_nms

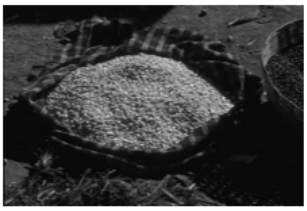


M_cc

Parameters:

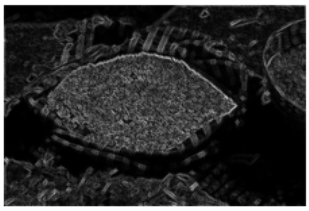
- Run Name: test_2_3
- Input Image: `assets/58060.jpg`
- Mask Size: 5
- σ: 1.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

Image

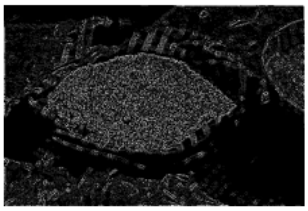


Ix

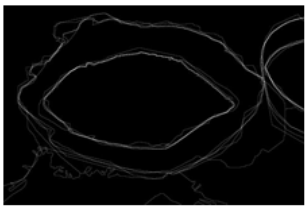
Iy



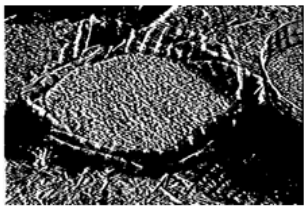
M



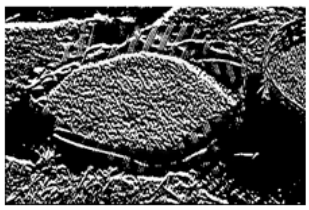
M_thresh



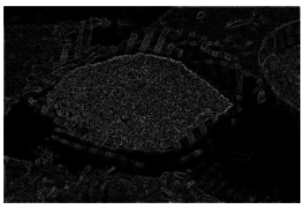
Ground Truth



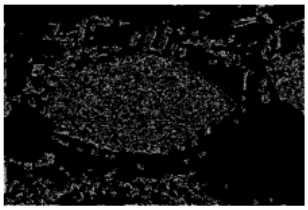
Ixx



Iyy



M_nms



M_cc

Parameters:

- Run Name: test_3_1
- Input Image: `assets/42049.jpg`
- Mask Size: 5
- σ: 0.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

Image

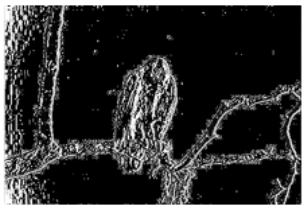
Ix

Iy

M

M_thresh

Ground Truth



Ixx

Iyy

M_nms

M_cc

Parameters:

- Run Name: test_3_2
- Input Image: `assets/42049.jpg`
- Mask Size: 5
- σ: 1.0
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

Image

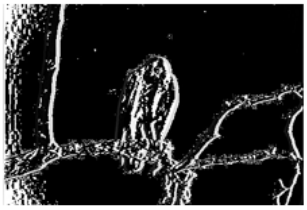
Ix

Iy

M

M_thresh

Ground Truth



Ixx

Iyy

M_nms

M_cc

Parameters:

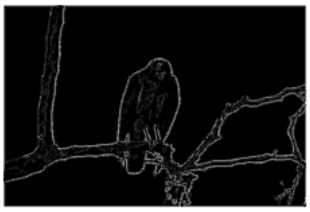
- Run Name: test_3_3
- Input Image: `assets/42049.jpg`
- Mask Size: 5
- σ: 1.5
- Lower Threshold: 0.1
- Higher Threshold: 0.35
- Middle Level: 0.4

Image

Ix

Iy

M



M_thresh

Ground Truth

Ixx

Iyy

M_nms

M_cc

Section D: Observations and Remarks

In general, it is observable that the variance of the Gaussian mask only makes the more coarse edge features prominent and suppresses the finer edge features. In `assets/58060.jpg` we observe a higher concentration of finer edge features which is being lost in the pipeline. However, `assets/42049.jpg` has more prominent and longer edges which are being detected better upon increasing variance. Thus, empirically, variance controls the size of features detected using the pipeline.