CS 435 - Computational Photography

Assignment 3 - Canny Edge Detection

Introduction

In this assignment you will demonstrate your ability to implement the individual components of a Canny Edge Detector
In this assignment you will demonstrate your ability to:

- Create and apply smoothing and gradient kernels.
- Find edge candidates by thresholding based on gradient strength.
- Apply hysteresis.
- Apply non-maximum suppression.

Grading

Theory Questions	20pts
Gaussian Smoothing	20pts
Computing Gradients	20pts
Apply threshold to gradients	10pts
Hysterisis	10pts
Non-maximum supression	10pts
Apply pipeline to addition image	10pts
TOTAL	100pts

Table 1: Grading Rubric

Dataset

For this assignment we are going to implement the various stages of a Canny Edge Detector and apply them to a few images, observing the results along the way.

I have provided a single sample image, circles1.gif, for us to observe the effect of stages of the edge detector. In the end you will also be asked to take a photo of a piece of paper, and to apply your edge detector to it (we'll be using this image for a later program!).

1 (15pts) Theory Questions

1. (5pts) Apply a 3×3 mean filter to the following 2D matrix. You may assume that the filter is only applied to areas of the data that have a full 9 samples to process. Feel free to use Matlab to help you compute this, however, realize that you may be asked to do this without a calculator on an exam.

$$I = \begin{bmatrix} 7 & 7 & 6 & 3 & 3 & 4 & 2 & 2 \\ 3 & 7 & 2 & 6 & 4 & 4 & 5 & 7 \\ 5 & 4 & 7 & 5 & 1 & 1 & 2 & 2 \\ 2 & 1 & 3 & 4 & 1 & 3 & 5 & 6 \\ 6 & 2 & 2 & 7 & 4 & 2 & 5 & 4 \\ 2 & 2 & 2 & 3 & 6 & 6 & 6 & 7 \\ 4 & 6 & 5 & 6 & 7 & 3 & 4 & 1 \\ 5 & 2 & 4 & 6 & 1 & 4 & 1 & 4 \end{bmatrix}$$

$$I' = \begin{bmatrix} 5.3333 & 5.2222 & 4.1111 & 3.4444 & 2.8889 & 3.2222 \\ 3.7778 & 4.3333 & 3.6667 & 3.2222 & 2.8889 & 3.8889 \\ 3.5556 & 3.8889 & 3.7778 & 3.1111 & 2.6667 & 3.3333 \\ 2.4444 & 2.8889 & 3.5556 & 4.0000 & 4.2222 & 4.8889 \\ 3.4444 & 3.8889 & 4.6667 & 4.8889 & 4.7778 & 4.2222 \\ 3.5556 & 4.0000 & 4.4444 & 4.6667 & 4.2222 & 4.0000 \end{bmatrix}$$

2. (5pts) What is the kernel function for a 5×5 Gaussian function with $\sigma = 1$? Show the filter as-is, and then discretize it so that all values are integers, that the minimum value is 1, then normalize it so that all its elements sum to 1. Feel free to use Matlab to help you compute this, however, realize that you may be asked to do this without a calculator on an exam.

$$W = \begin{bmatrix} .0183 & .0821 & .1353 & .0821 & .0183 \\ .0821 & .3679 & .6065 & .3679 & .0821 \\ .1353 & .6065 & 1.000 & .6065 & .1353 \\ .0821 & .3679 & .6065 & .3679 & .0821 \\ .0183 & .0821 & .1353 & .0821 & .0183 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1.0000 & 4.4817 & 7.3891 & 4.4817 & 1.0000 \\ 4.4817 & 20.0855 & 33.1155 & 20.0855 & 4.4817 \\ 7.3891 & 33.1155 & 54.5982 & 33.1155 & 7.3891 \\ 4.4817 & 20.0855 & 33.1155 & 20.0855 & 4.4817 \\ 1.0000 & 4.4817 & 7.3891 & 4.4817 & 1.0000 \end{bmatrix} \Longrightarrow$$

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 20 & 33 & 20 & 4 \\ 7 & 33 & 55 & 33 & 7 \\ 4 & 20 & 33 & 20 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} 0.0588 & 0.2353 & 0.4118 & 0.2353 & 0.0588 \\ 0.0494 & 0.2469 & 0.4074 & 0.2469 & 0.0494 \\ 0.0519 & 0.2444 & 0.4074 & 0.2444 & 0.0519 \\ 0.0494 & 0.2469 & 0.4074 & 0.2469 & 0.0494 \\ 0.0588 & 0.2353 & 0.4118 & 0.2353 & 0.0588 \end{bmatrix}$$

3. (5pts) What is the derivative of a 3×3 mean kernel with respect to x and y?

$$W = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \frac{\partial}{\partial x} = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}; \frac{\partial}{\partial y} = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\frac{\partial W}{\partial x} = \frac{\partial}{\partial x} * W = \frac{1}{18}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\frac{\partial W}{\partial y} = \frac{\partial}{\partial y} * W = \frac{1}{18}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

4. (5pts) Given the following 2D kernels, what is the magnitude and direction of the gradient at the center pixel in I? Feel free to use Matlab to help you compute this, however, realize that you may be asked to do this without a calculator on an exam.

$$\frac{\partial}{\partial x} = \begin{bmatrix} -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & 0 & \frac{1}{3} \end{bmatrix}, \frac{\partial}{\partial y} = \begin{bmatrix} -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$I = \begin{bmatrix} 7 & 7 & 6 \\ 3 & 7 & 2 \\ 5 & 4 & 7 \end{bmatrix}$$

$$G_x = \frac{\partial I_{center}}{\partial x} = \frac{7}{3} + 0 - 2 + 1 + 0 - \frac{2}{3} + \frac{5}{3} + 0 - \frac{7}{3} = 0$$
$$G_y = \frac{\partial I_{center}}{\partial y} = \frac{7}{3} + \frac{7}{3} + 2 + 0 + 0 + 0 - \frac{5}{3} - \frac{4}{3} - \frac{7}{3} = \frac{4}{3}$$
$$\|G\| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| = \frac{4}{3}$$
$$\theta = \arctan \frac{G_y}{G_x} = 90°$$

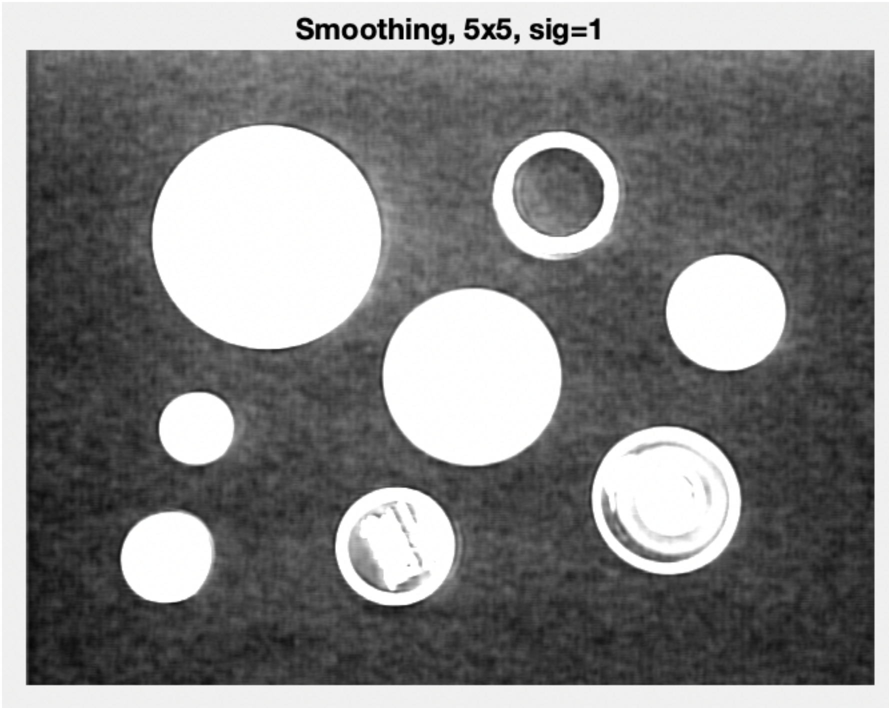
2 Gaussian Smoothing

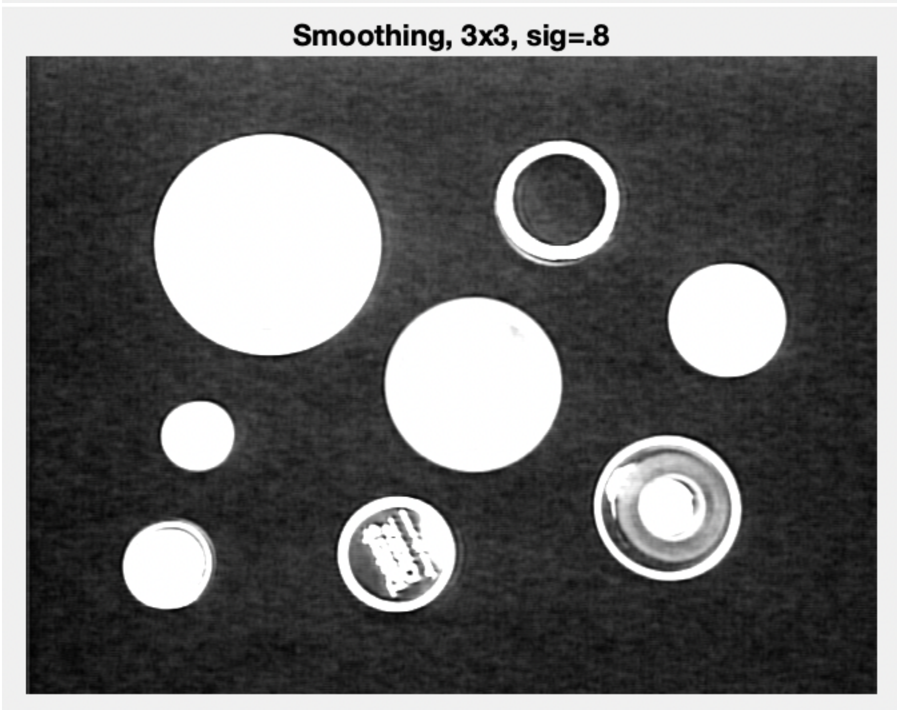
The first step in the Canny Edge Detector is to apply a Gaussian smoothing kernel to your image. Our edge detection will be done in grayscale, so first convert your image to grayscale, if necessary.

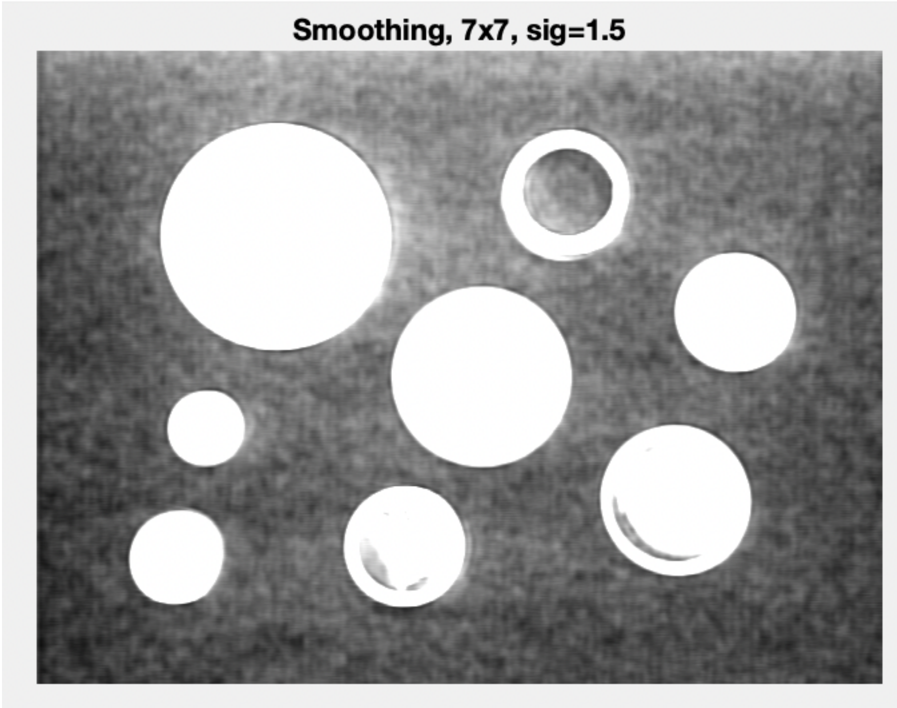
Next, given an odd filter size, N and a Gaussian variance parameter, σ, compute the $N \times N$ Gaussian smoothing kernel and apply it to your image to generate a smoothed new image. Just apply the kernel to the "inside" of the image, that is, areas where there is a large enough neighborhood to apply your kernel.

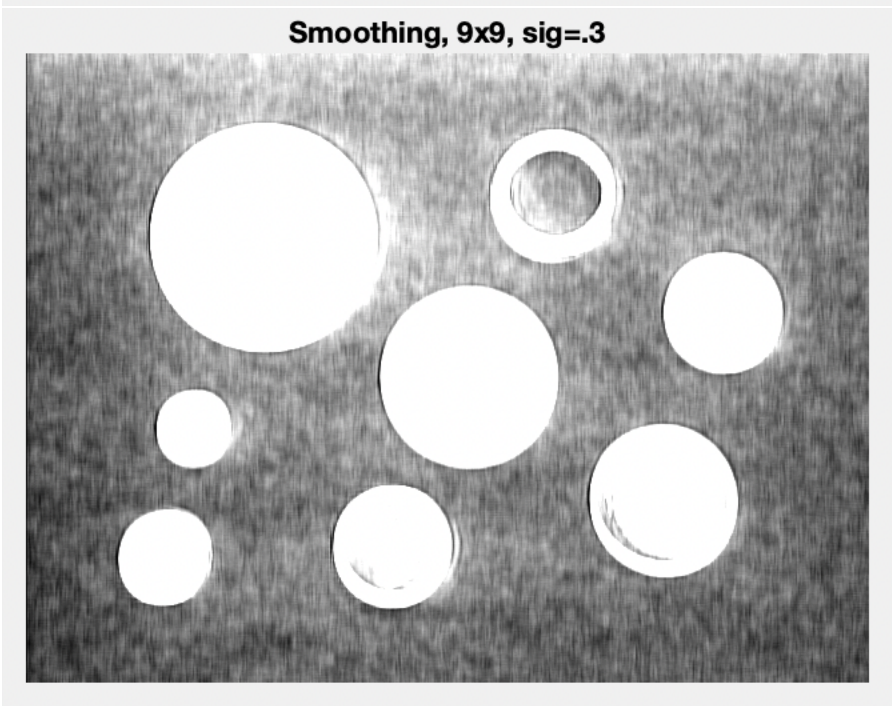
Show the original grayscale image and then the results for at least 4 different combinations of (N, σ)

Note: For this part you **may not** use Matlab's *conv2* function. I would like you to at least once implement convolution yourself.


Smoothing, 5x5, sig=1


Smoothing, 3x3, sig=.8





$N = 5, \sigma = 1$
$N = 3, \sigma = .8$
$N = 7, \sigma = 1.5$
$N = 9, \sigma = .3$

3 Gradients

Next, we'll compute the gradients on our image.

Generate three images using the original image (not smoothed):

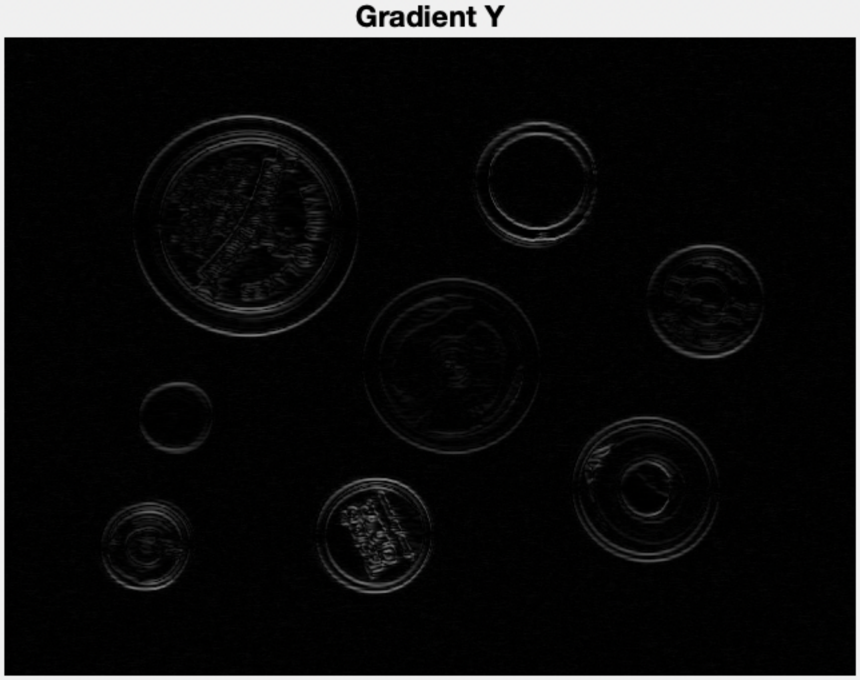
1. One which has the intensity with respect to the change in x
2. One that has the intensity with respect to the change in y
3. One that has the overall magnitude of the combined gradients.

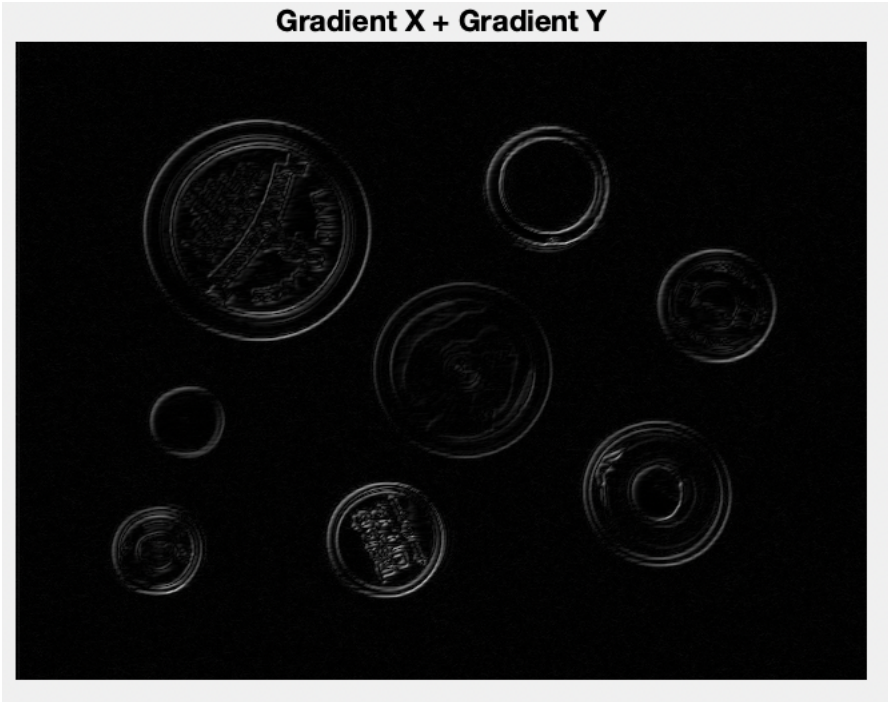
Doing this on your original image you'll likely see "noise", so next try first applying a smoothing filter (like you developed in the previous part) to remove noise prior to extracting gradients. You can choose the parameters of the kernel as you see fit. Show these images as well (6 total images).

Note: Since you already demonstrated in the previous part your ability to perform convolution, for the remaining parts of the assignment you **may** use Matlab's *conv2* function.

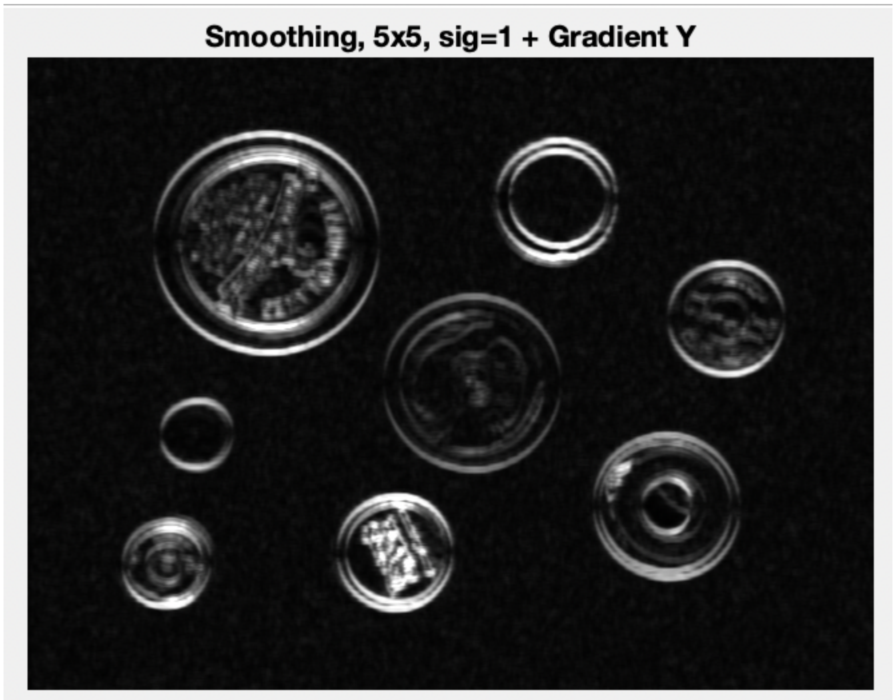
Gradient X

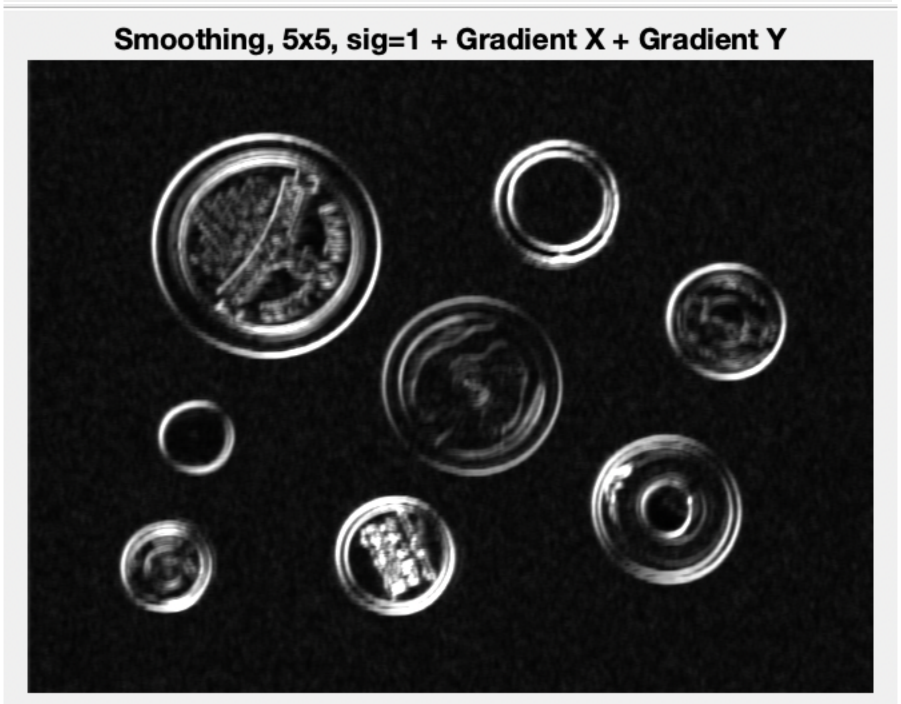
Gradient Y




Gradient X + Gradient Y


Smoothing, 5x5, sig=1 + Gradient X



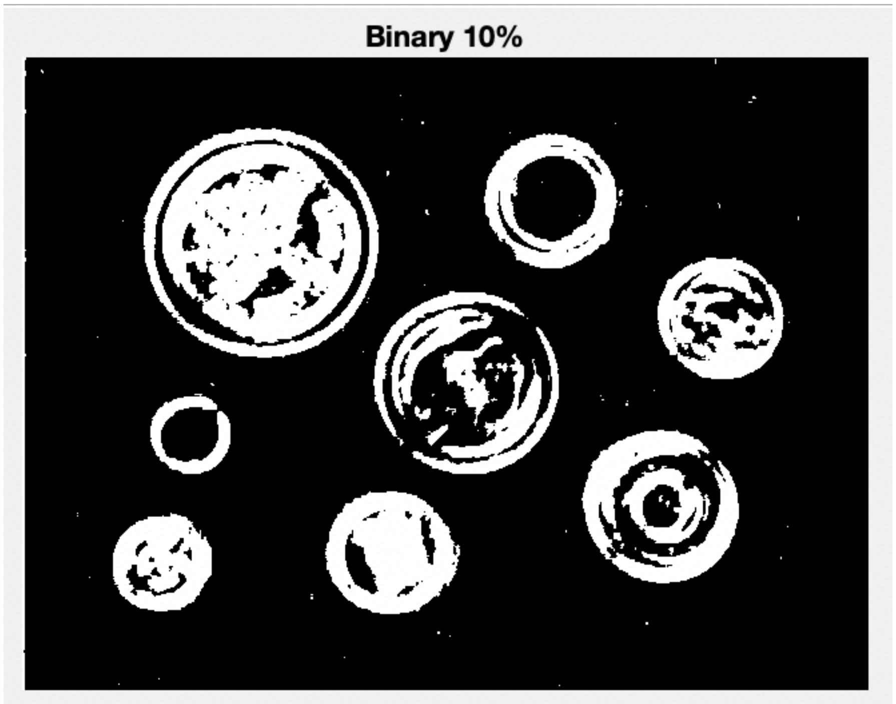


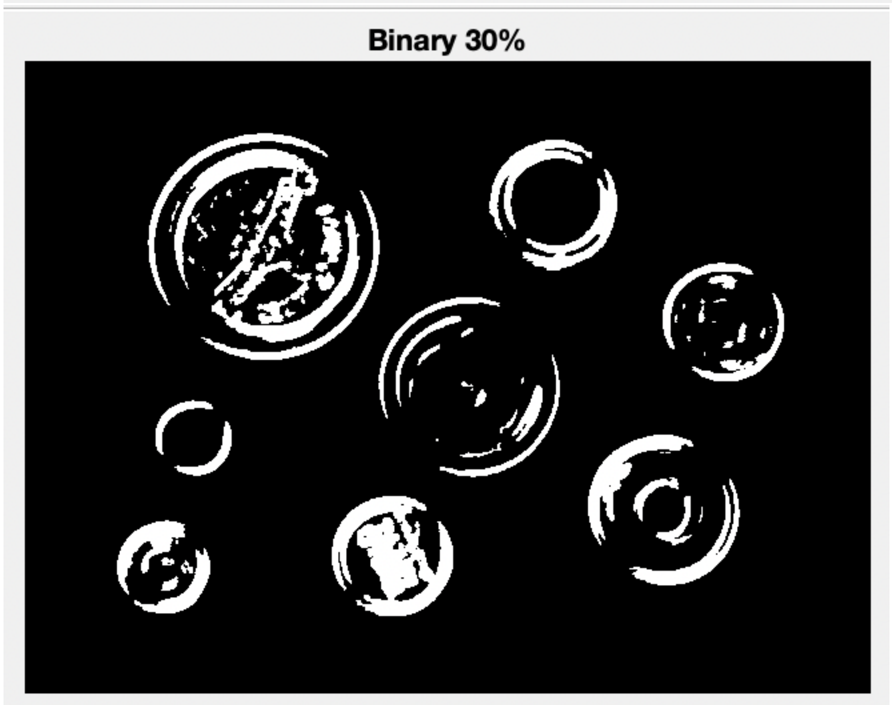
Parameters used: $N = 5, \sigma = 1$

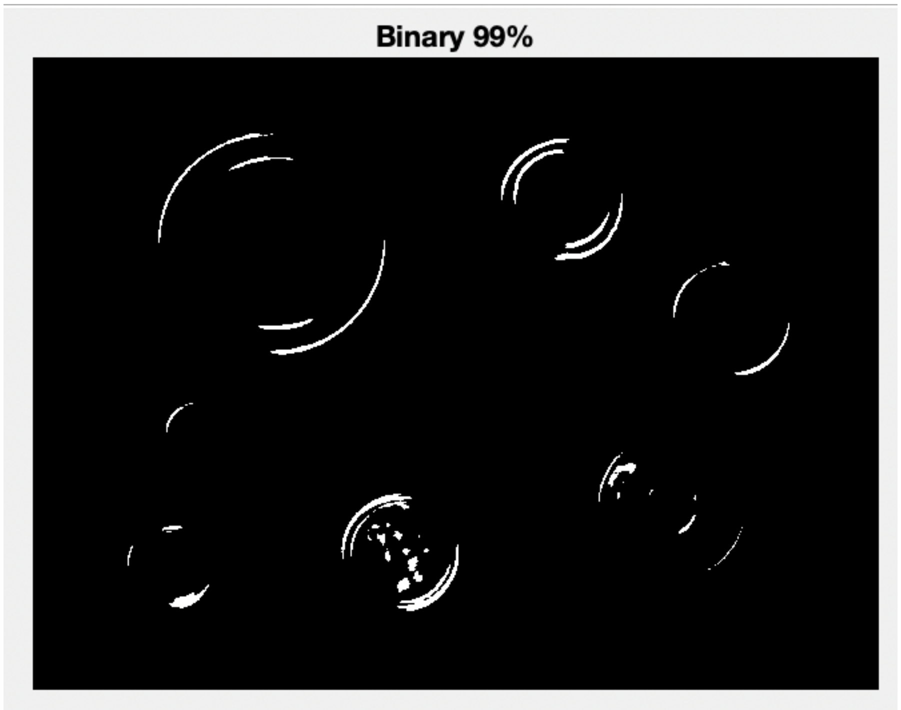
4 Threshold

On your gradient magnitude image, apply a threshold to obtain binary images. Experiment with a few different thresholds.

Show your output for a few different choices of your threshold.


Binary 10%

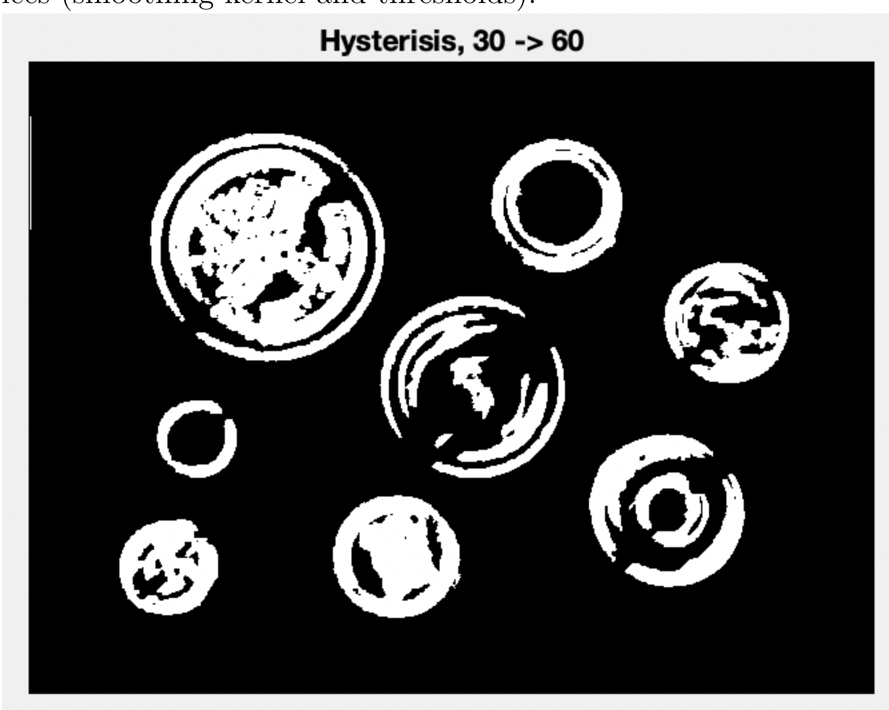

Binary 30%



Thresholds: $25.5, 76.5, 252.45$

5 Hysterisis

Now set a low and high threshold such that a pixel is an edge pixel if its gradient is greater than the high threshold, or if it is greater than the low threshold and borders (8-way) a pixel that is above the high threshold. Here you just need to provide one example output, but also report the parameter choices (smoothing kernel and thresholds).



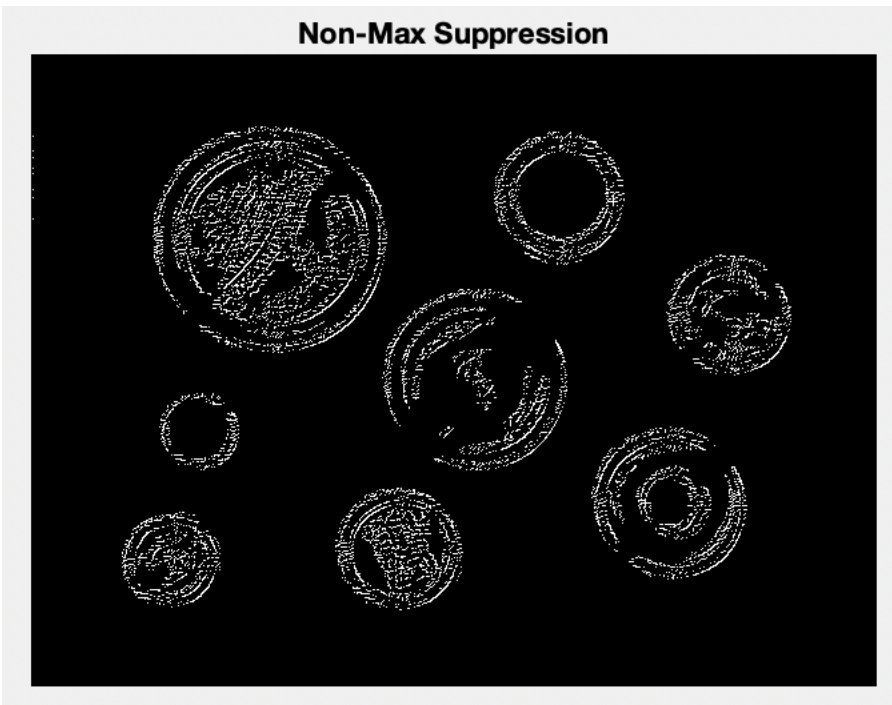
Low: 30, High: 60
Parameters Used: $N = 5, \sigma = 1$

6 Non-maximum supression

As our last step, let's apply *non-maximum supression* to the edge image you created via hysterisis.

Using the original gradient information, compute the angle of the gradient for each pixel. Then, for all the pixels that were flagged as potential edge pixels by the hysterisis process, compare the magnitude of the gradient of this pixel to its neighbors, as defined in the table below. If it is the largest of its neighbors, flag it as an edge pixel. Otherwise mark it as not an edge pixel.

Pixels to check	**Angle**
Left and right	$\theta \geq \frac{7\pi}{8}$ or $\theta < \frac{\pi}{8}$
Up-right and down-left	$\frac{\pi}{8} \leq \theta < \frac{3\pi}{8}$
Up and down	$\frac{3\pi}{8} \leq \theta < \frac{5\pi}{8}$
Up-left and down-right	$\frac{5\pi}{8} \leq \theta < \frac{7\pi}{8}$



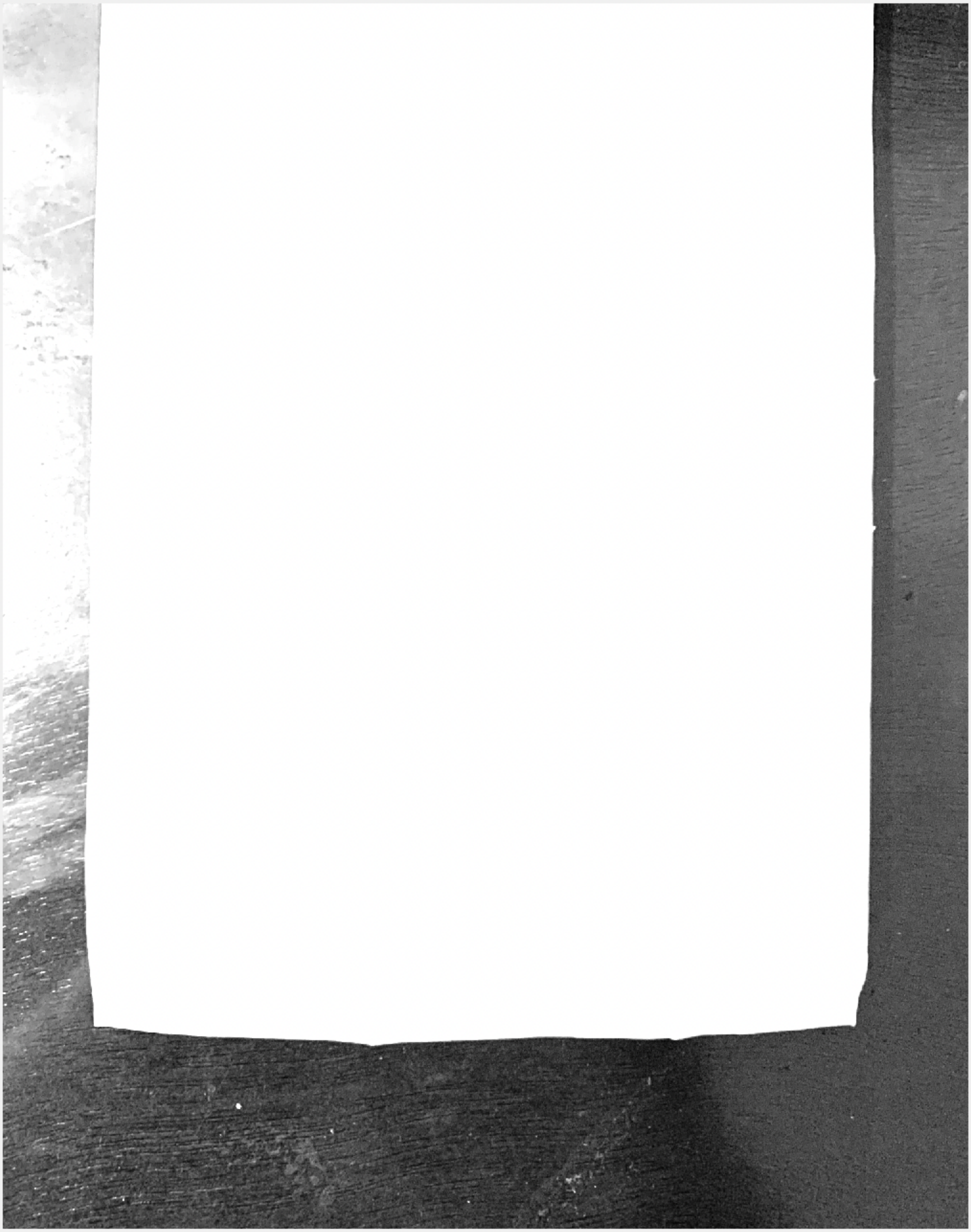
Low: 30, High: 60
Parameters Used: $N = 5, \sigma = 1$

7 Test on another image

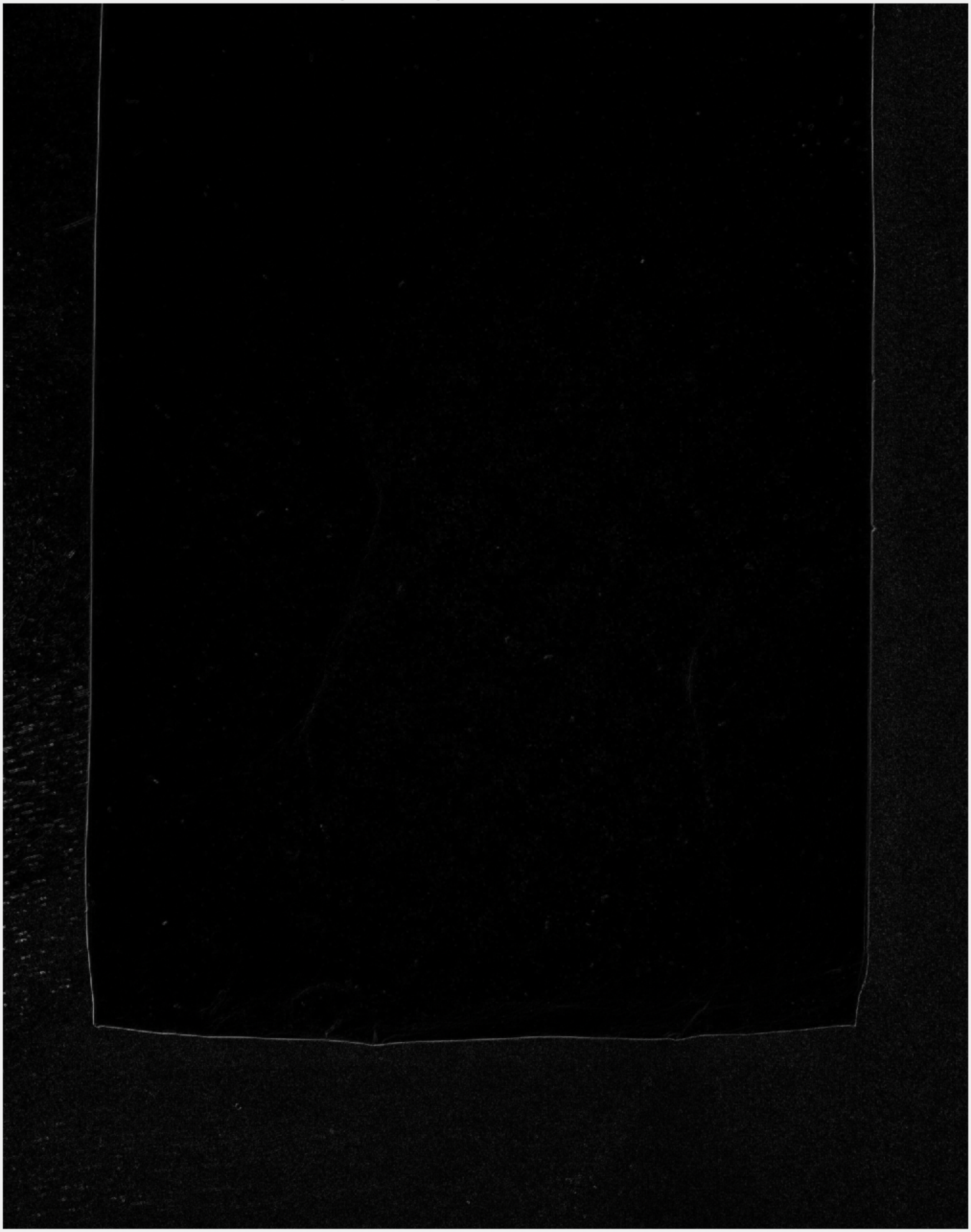
Now that you have all the stages of your Canny Edge Detector implemented, take a photo of a piece of paper on a dark background and apply your Canny Edge Detector to it. Show the result as it goes through each part of the pipeline.

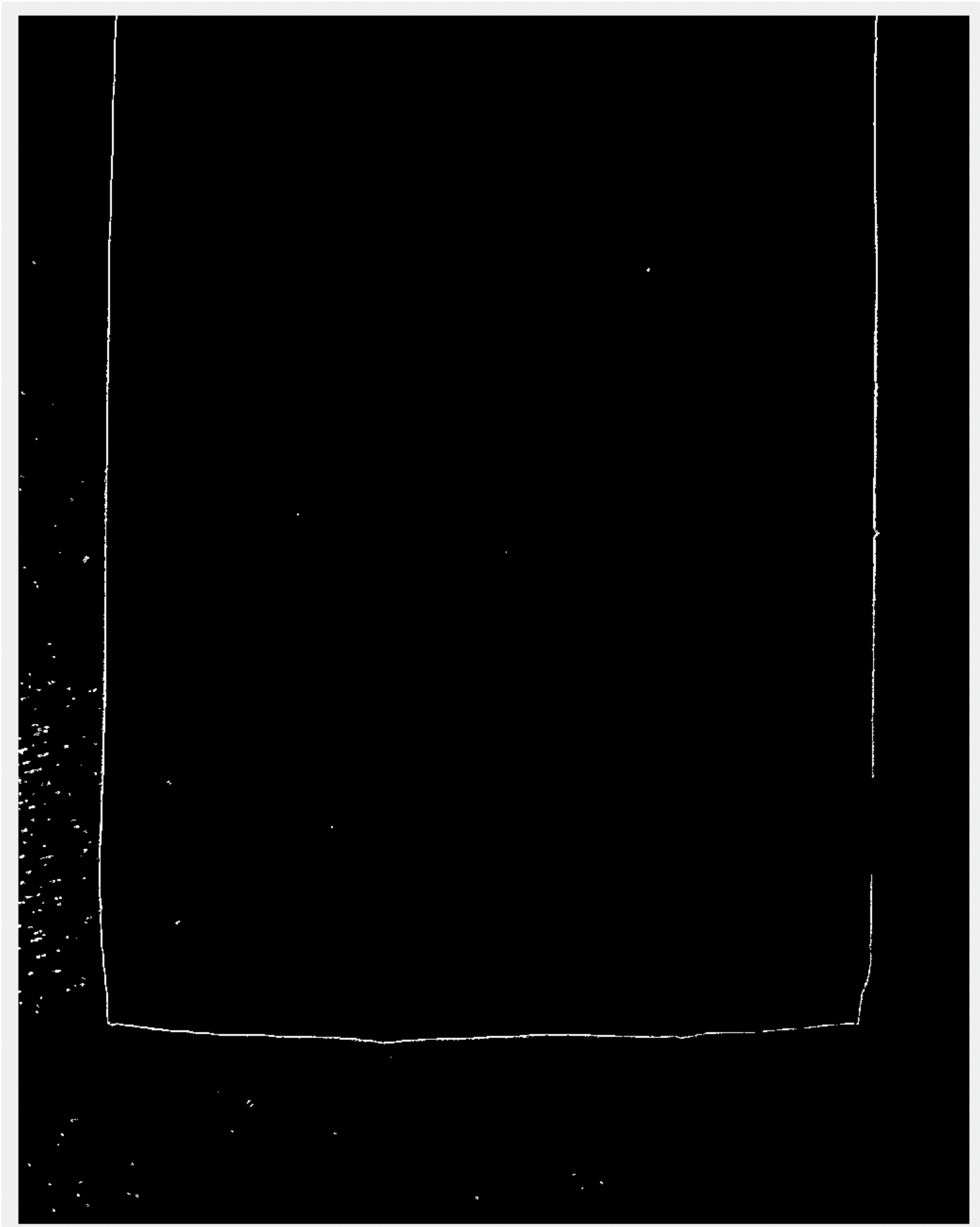
$N = 3, \sigma = .6$

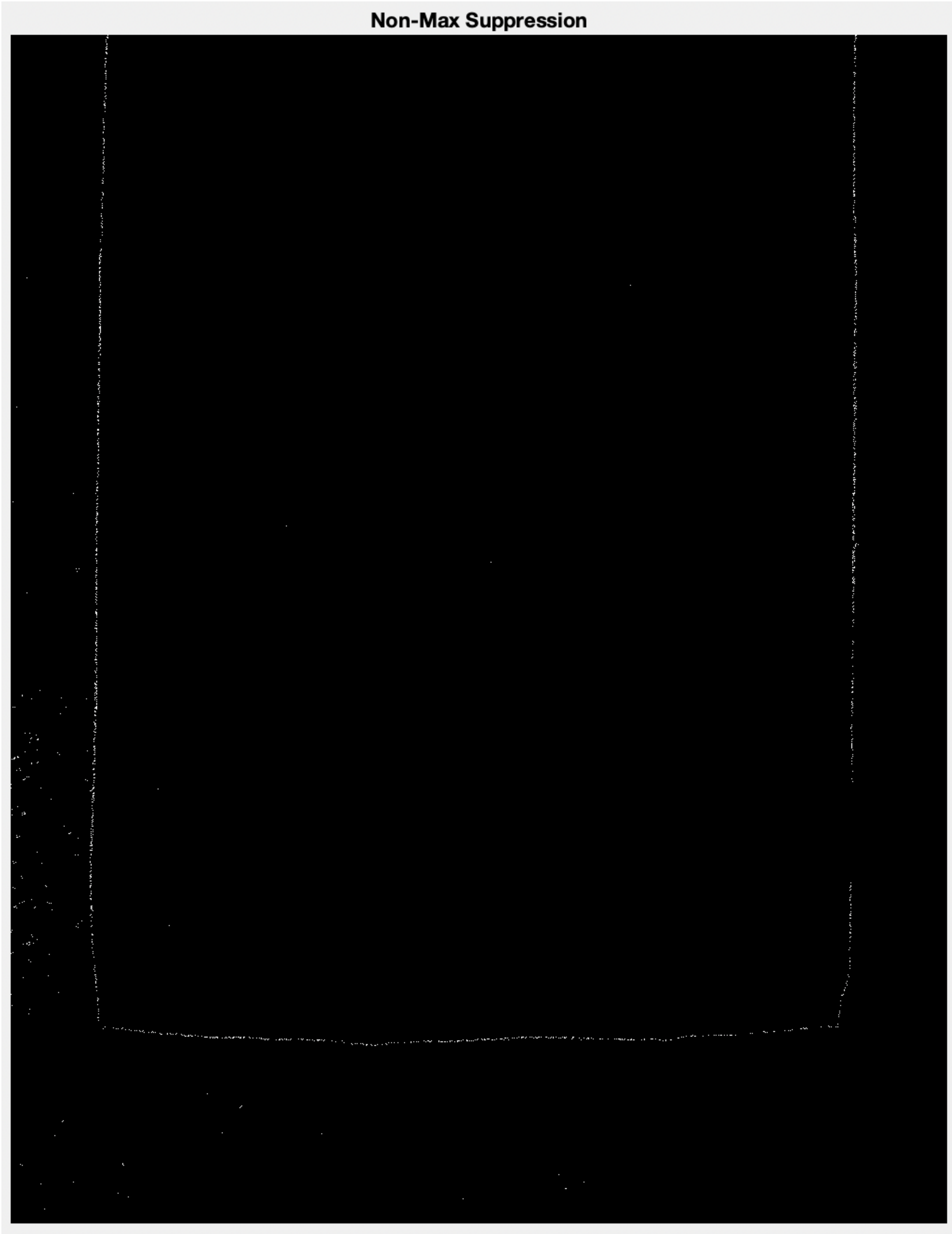
Original



Smoothing, NxN, sig=1 + Gradient X + Gradient Y






Non-Max Suppression

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF writeup that includes:
   (a) Your answer to the theory question(s).
   (b) For Part 2, 5 images. The original grayscale, then 4 combinations of the parameters (N, σ).
   (c) For Part 3, 6 total images. Three showing the partial gradients and magnitude of the gradient without pre-smoothing, and three after. In addition, report the parameters of the kernel you used for smoothing.
   (d) For Part 4, a few (at least three) edge images using different thresholds. Accompany the images with the chosen threshold values.
   (e) For Part 5, your edge image usng hysteresis. In your report provide the parameters of the smoothing kernel and the low and high threshold values for the hysterisis process.
   (f) For Part 6, your final edge image after applying smoothing, hysterisis, and non-maximum supression. Again, report all parameter choices.
   (g) For Part 7, 5 total image:
      i. Original image (either color or gray)
      ii. Smoothed image (along with choice of smoothing filter parameters).
      iii. Gradient magnititude image
      iv. Hysterisis edge image (along with choice of thresholds)
      v. Final edge image with non-maximum suppression applied.
2. A README text file (**not** Word or PDF) that explains:
   (a) Any unique features of your program (if applicable).
   (b) Any instructions on how to run your script to reproduce your results.
3. Your source file(s).
4. The chosen image(s) that you processed.