EE 473 - *Introduction to Digital Signal Processing*

Semester Project

Project Name:

Edge Detector Application for Mobile Devices

Project Members:

Sezai Burak Kantarcı - 2020701087

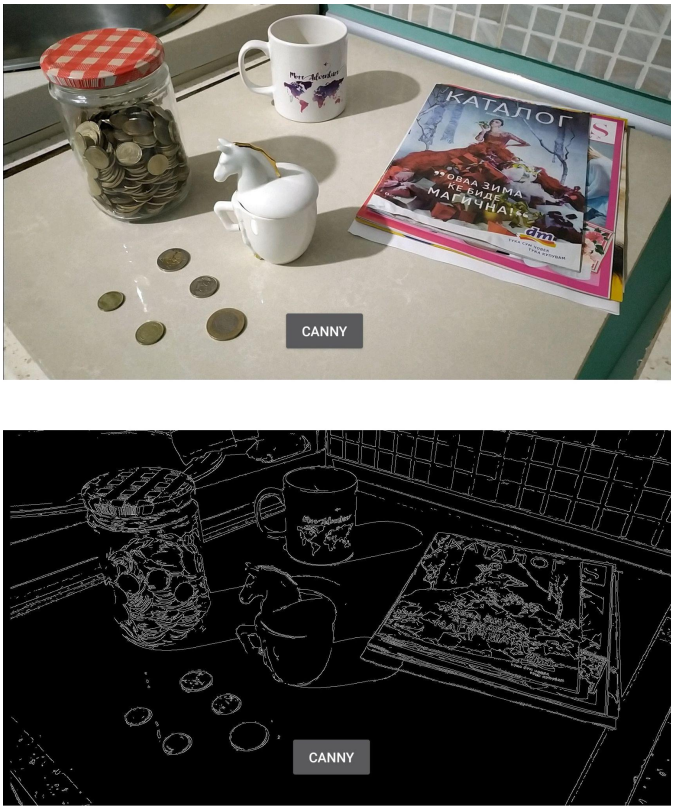

Figure 1: Original Image (Top) - Application Running (Bottom)

1. Introduction

The Photo Industry skyrocketed with how commonly smartphones have become. Instagram, Reddit and Pinterest are just a few of the platforms that make use of this massive market.

In this project, we had the goal of bringing an edge detector as a cool filter, so people can use it to have a different effect on their photos. The target audience was the regular users of the platforms given above.

Firstly we discuss some basics about Computer Vision in the paper.

We follow up with our methodology & how we approached the problem.

Finally, results and the overall success of the project in the given time frame is discussed, with a special light shed on the experience gained with the project.

2. Building Blocks

A digital image is a numerical representation of a 2D vector. The edge is a fundamental part of an digital image and contains a wealth of information that is important for obtaining the characteristics of that frame [1].

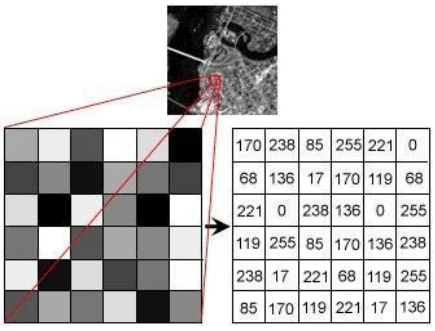

Figure 2: A Digital Image [8].

The research behind the edges started in the 1960's. Hubel and Wiesel recorded the neuron responses in a cat's brain, in order to observe how different parts of the brain react to different stimuli. Neurons in the cats were most stimulated when edges moved in specific directions [5].

Dr. Biederman studied how fast humans can identify objects. He found out that the rates of recognition were almost exactly the same for two groups; one given the full object picture and the other were only looking at its edges [6].

Moreover, Edge Detection reduces the amount of data and eliminates unwanted information, while retaining all the important attributes of an image.

A variety of computer vision areas are related with edge detection, such as segmentation, object detection, object recognition, tracking, unmanned vehicle routing and image-to-text analysis [4].

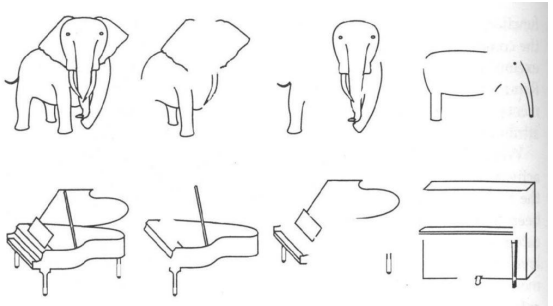

Figure 3: Recognizing Objects [7].

Sobel [9], Prewitt [10] and Canny [11] Algorithms are the most common ones in order to solve edge detection problem. Following up, we narrow this circle and focus on the Canny Edge Detection and the production of this project.

3. Methodology

In the project, the Canny Edge Detection Algorithm was implemented through OpenCV library. Originally, it was developed by John F. Canny in 1986 [1]. Although it is quite old, this method is still used by researchers as one of the standard edge detection methods.

3.1. Canny Edge Detection

Canny takes an original RGB image and outputs an edge detected version of it. It can be broken down to four algorithmic steps.

- Noise Reduction - Smoothing

Finding edges is susceptible to noise in the image; so the first step is to remove it in the image with a 5x5 Gaussian filter.

OpenCV Canny() function, uses a 5*5 GaussianBlur(). It has the effect of reducing the image's high-frequency components, as a low pass filter. An example kernel is given in Figure 4.

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figure 4: A Gaussian Kernel with $\sigma = 1.4$ [8].

- Finding Gradients

Based on grayscale intensities, the Canny algorithm locates edges where the intensity changes most. This is done by determining gradients in the image. Edge Gradient G and it's angle is given as:

$$G = \sqrt{G_x^2 + G_Y^2} \qquad (1)$$

$$\theta = arctan(\frac{G_Y}{G_X}) \qquad (2)$$

where G_X and G_Y are the gradients in The x and y directions respectively. θ can be examined in Figure 5.

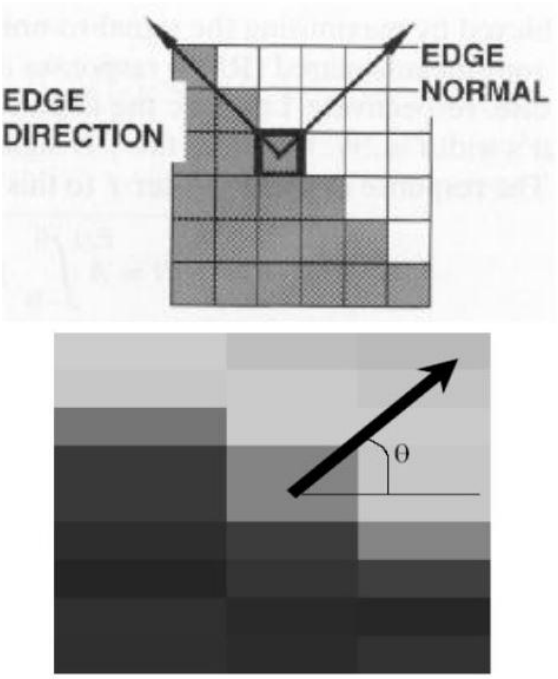

Figure 5: Edge Gradient (Top) and the Angle (Bottom) [13].

- Non-Maximum Suppression

To remove any pixels that may not belong to the edge, the image is scanned to determine magnitude and direction of gradients.

For every pixel, the neighborhood of the pixel is checked to see if it is a local maximum.

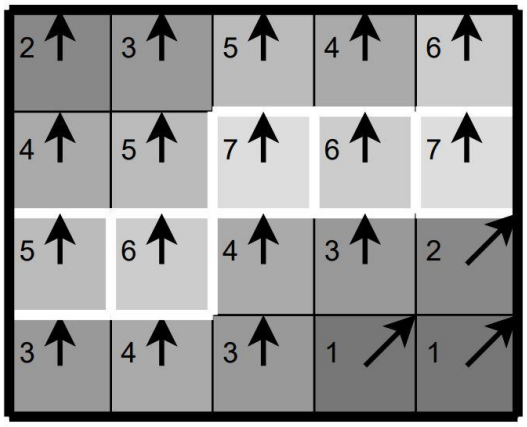

Figure 6: Non Maximum Suppression among all pixels [12].

- Hysteresis Thresholding

During this stage, we determine which edges are really edges and which are not with the help of two thresholds.

- A pixel is considered an edge pixel if its gradient value is higher than the upper threshold.

- A pixel is rejected if its gradient value is less than the lower threshold.

- When the gradient value of a pixel is between the lower and upper thresholds, it will only be accepted if it is connected to a pixel that is above the upper threshold.

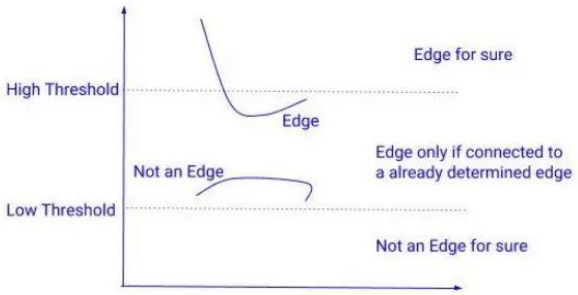

Figure 7: Hysteresis Thresholding [14].

3.2. Android Studio

In the making of the application for Android Operating System, Android Studio IDE is used since it's free and it has a global community.

In order to simply apply Canny for an image taken from a smartphone camera, OpenCV Android Library was used.

The pseudo code is given down below.

< Initialize the Application >

< Open the camera and start the frame stream in full screen>

< If the Canny Button is pressed>

<Take each frame and turn it into Grayscale>

<Take the grayscale frame and use it as an input to the Canny function>

<Return Edge Detected Frame>

<Return Frame>

All the code for Canny as well as Sobel Edge Detectors is made available at: github.com/kantarcise/edge-detection

4. Results

The *MVP* (Minimum Viable Product) for the project was given in the proposal as an application that is able to make use of the mobile phones camera and show Canny Edge Detection, real time. So that was achieved.

However, other filters like Holistically Nested Edge Detection and Sephia Filters in Android were not completed in the given time frame.

The difficulties in the project mostly came from working alone, which was expected.

In the future, the application can be broadened with detailed research on product-market fit. Moreover, some features can be added with the feedback of the users.

Also, the general metrics for an edge detector performance was not evaluated. These metrics can be given as:

- Probability of false edges.
- Probability of missing edges.
- Error in estimation of the edge angle.

In order to compensate for this situation, another Edge detector called “Sobel” was implemented in Python in order to compare the results.

Some of the results from the application are given down below.

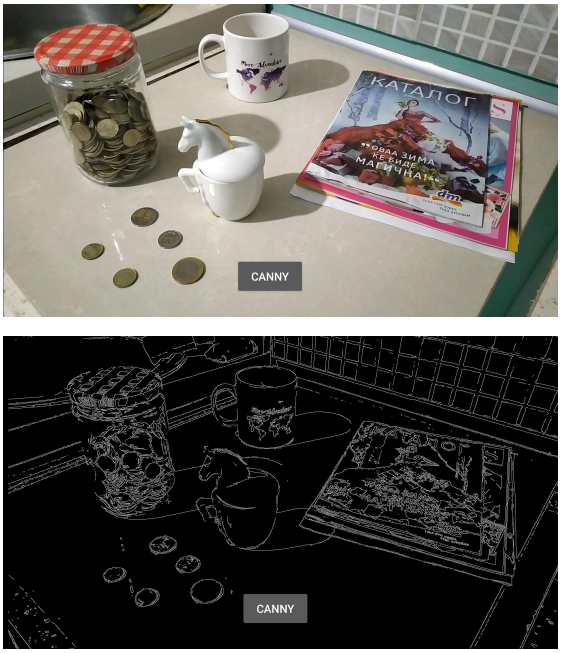

Figure 8: First Example; Original Image (Top), Canny (Middle), Sobel (Bottom).

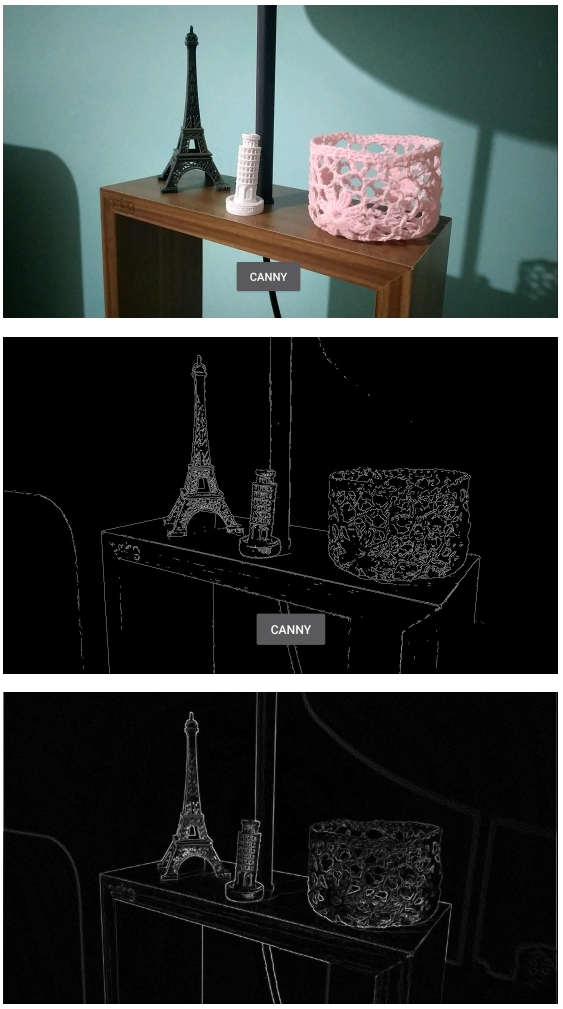

Figure 9: Second Example; Original Image (Top), Canny (Middle), Sobel (Bottom).

5. References

[1] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[2] Reddy, R. & Nagaraju, Chiluka & Reddy, I Raja Sekhar. (2016). Canny Scale Edge Detection. IJETT. 10.14445/22315381/IJETT-ICGTETM-N3/ICGTETM-P121.

[3] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1395-1403, doi: 10.1109/ICCV.2015.164.

[4] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1395-1403, doi: 10.1109/ICCV.2015.164.

[5] HUBEL, D H, and T N WIESEL. "Receptive fields of optic nerve fibres in the spider monkey." The Journal of physiology vol. 154,3 (1960): 572-80. doi:10.1113/jphysiol.1960.sp006596

[6] Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. Psychological Review, 94(2), 115–147. https://doi.org/10.1037/0033-295X.94.2.115

[7] What is an Image? : Stanford CS131 Lecture Notes

[8] Das, Goutam & Anwar, Nurul & Chowdhury, Shovan & Rahman, Kazi. (2016). Design and Fabrication of an Image Processing Based Autonomous Weapon. International Journal of Engineering Research. ISSN. 2319-68902347. 10.17950/ijer/v5s12/1212.

[9] N. Kanopoulos, N. Vasanthavada and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," in IEEE Journal of Solid-State Circuits, vol. 23, no. 2, pp. 358-367, April 1988, doi: 10.1109/4.996.

[10] Maini, R., & Sohal, J.S. (2006). Performance Evaluation of Prewitt Edge Detector for Noisy Images.

[11] Canny Edge Detector: Wiki

[12] Canny Edge Detection - March 23, 2009, link

[13] CS Toronto - Edge Detection - CSC320: Introduction to Visual Computing Michael Guerzhoy - link

[14] Deep Learning based Edge Detection in OpenCV - link