Final Project Report

Document Scanner

ECEN 447

Ting-Kai Huang

UIN: 826001835

Professor Righetti

Summary:

The purpose of this project is to create a document scanner program that can turn a photo of a document into a scanned version of it with the result only containing the document itself. To achieve this, I utilized several image-processing techniques using the OpenCV library in Python, including image smoothing, edge detection, image warping, bottom-hat, and thresholding technique. I will discuss the process of the program execution as well as display the result of each processed image using one image from the image set. Finally, I will display the original pictures and the resulting photos for the rest of the images in the image set.

The idea behind this project is to utilize the contour areas after applying the Canny Edge Detection technique. It is obvious that the document will have the largest contour area, so the program will find the largest contour area in the edge detection result which will be the document. However, we must make sure that the Canny Edge Detection technique can produce the contours for the document properly, so that the program can isolate the document. To make sure that the program can produce the desired contours for the document, I set up a while loop for the program to loop through the entire process.

The reason that I set up a while loop is that the Canny Edge Detection technique will start at a threshold of 200 as the parameter, and if it doesn't work, the program will subtract 25 from the threshold and try processing the image again. It will continue doing this until the program can correctly detect a complete closed contour for the document. After finding the contours for the document, the program will then extract the four corner points of the document and warp it so that the result only contains the document itself. After that, the program will apply the bottom-hat technique and the thresholding technique before producing the final scanned result of the document.

Program Execution Process:

See the following page.

Step 1:

The program reads and displays the image. (Figure 1)

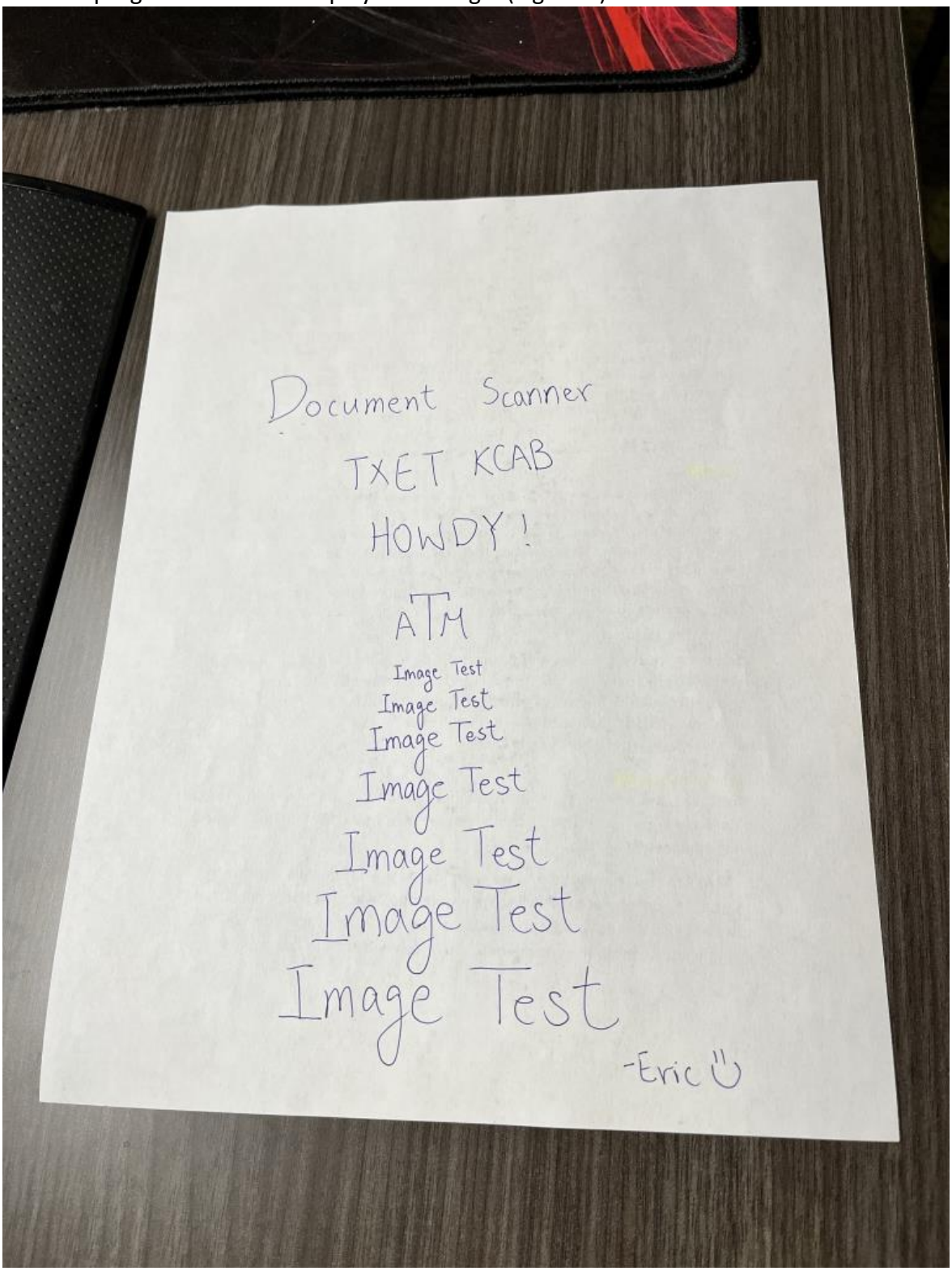


Figure 1 Original Image

Step 2:

The program converts the image to a gray scale image. (Figure 2)

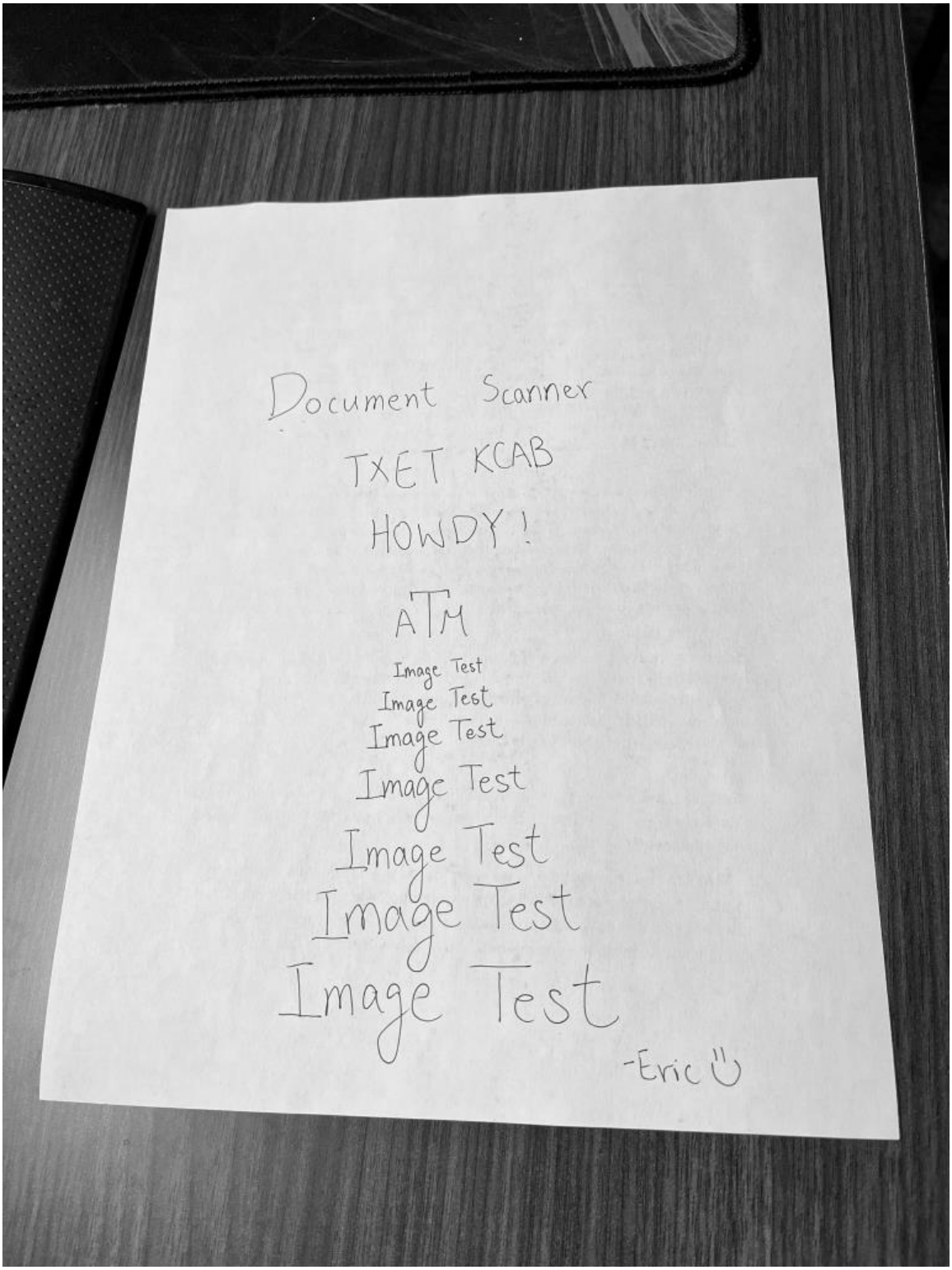


Figure 2 Gray Scale Image

Step 3:

The program applies Gaussian Blur technique to the image. (Figure 3)

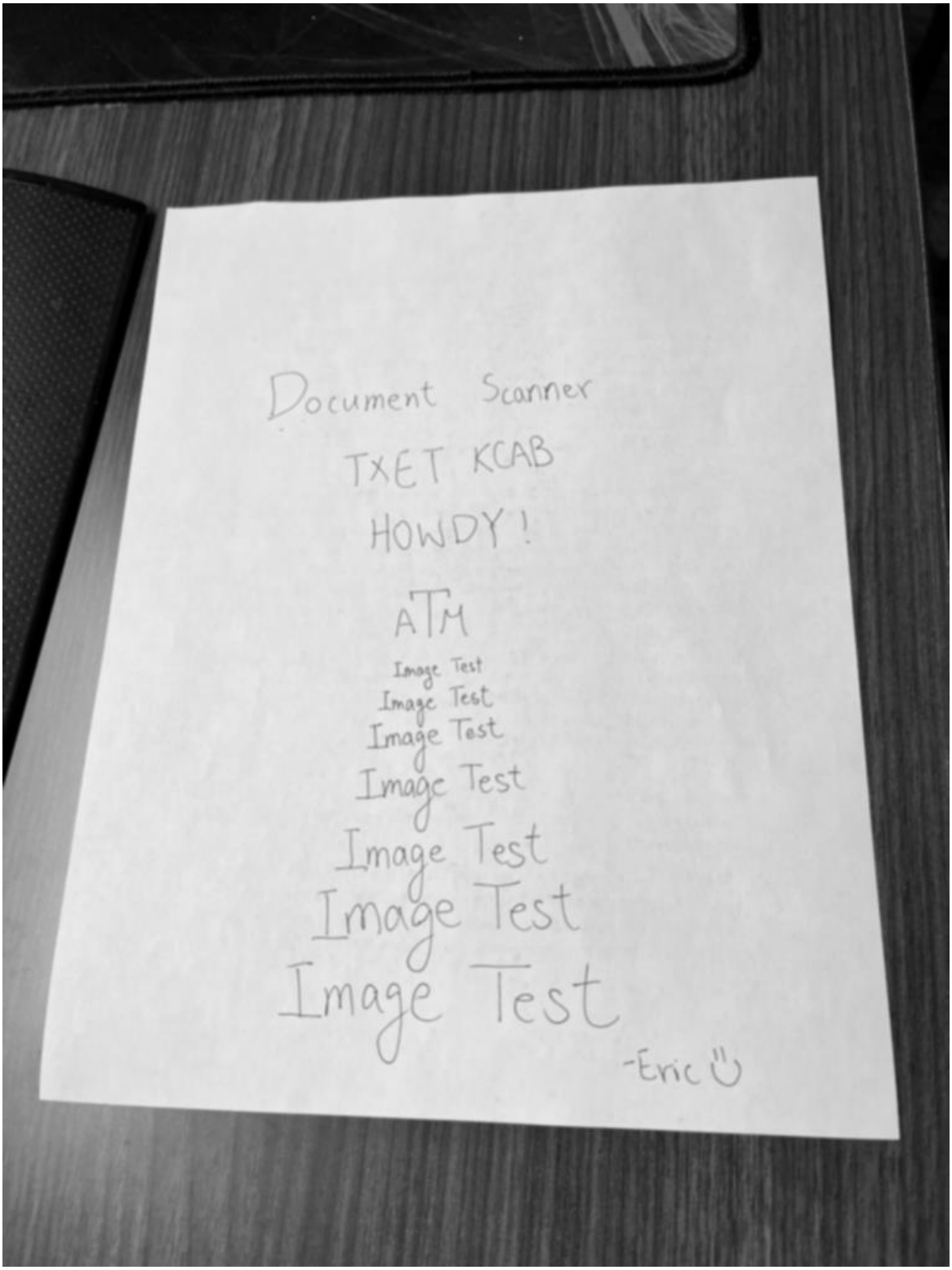


Figure 3 Gaussian Blurred Image

Step 4:

The program applies Canny Edge Detection technique to the image. (Figure 4)

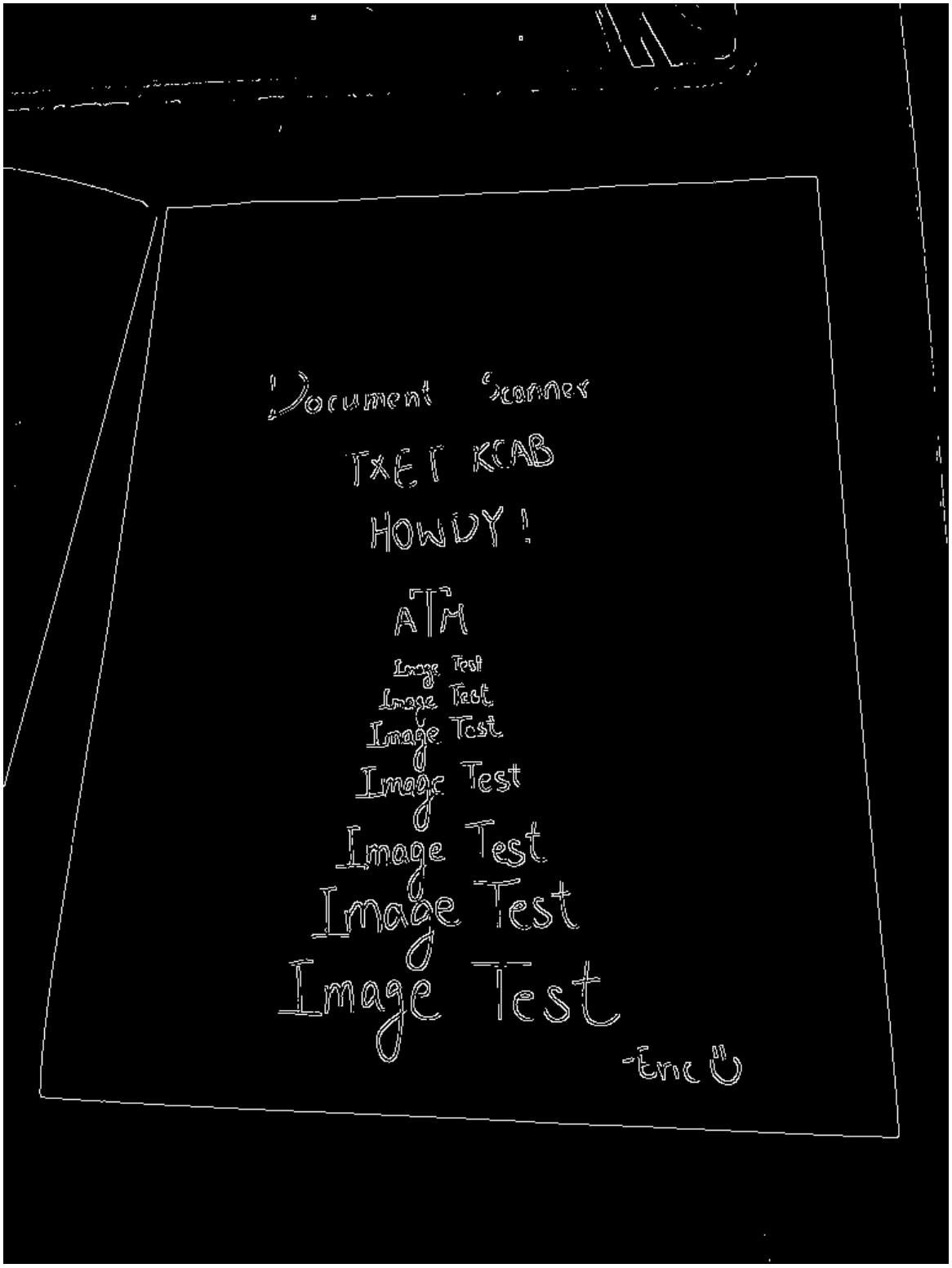


Figure 4 Canny Edge Detection Image

Step 5:

The program thickens the edges by applying dilation twice and erosion once with a 5 by 5 square structuring element. (Figure 5)

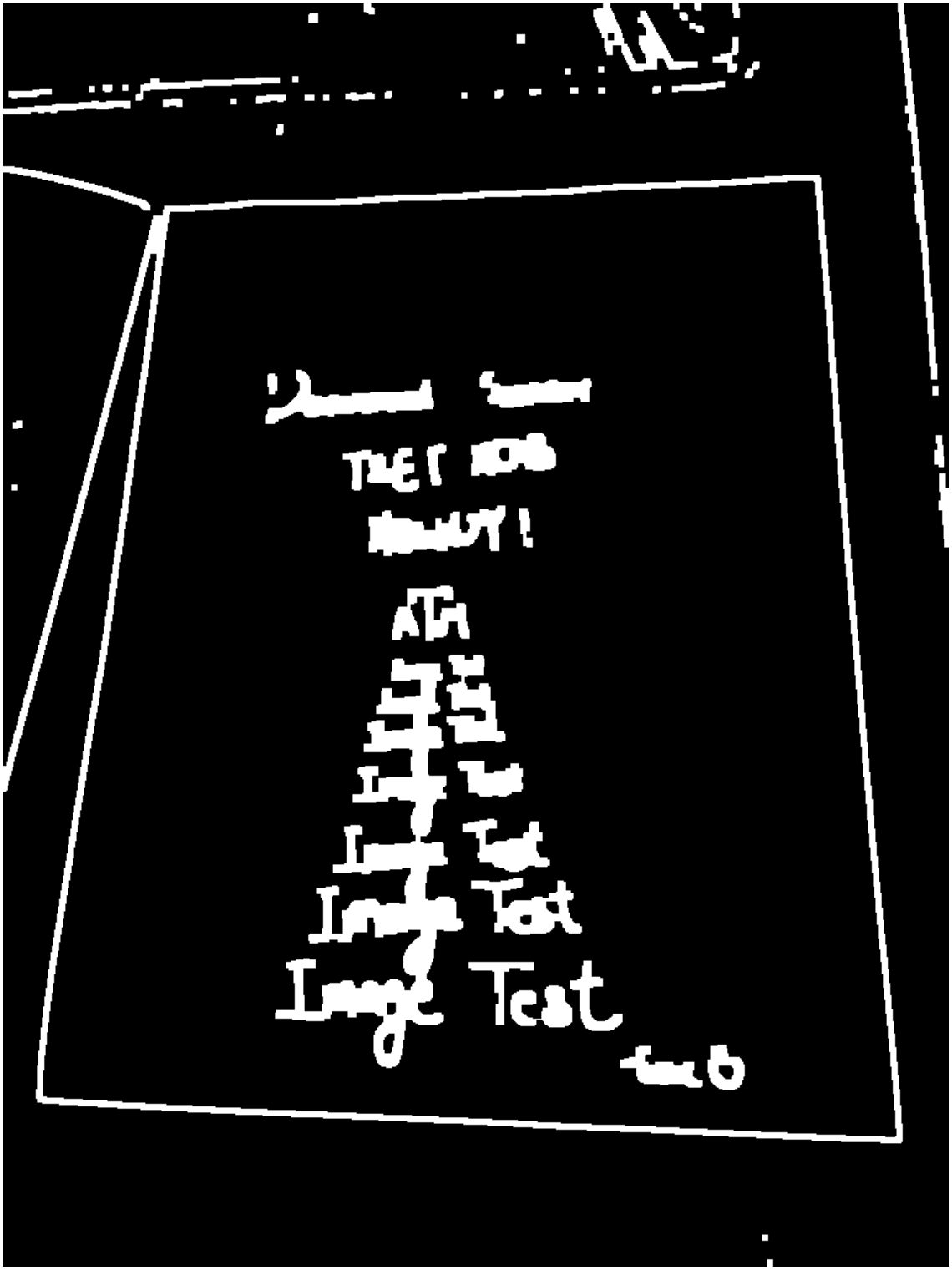


Figure 5 Thickened Edges

Step 6:

The program extracts the 4 corner points of the document and warps it so that the resulting image only contains the document itself. (Figure 6)

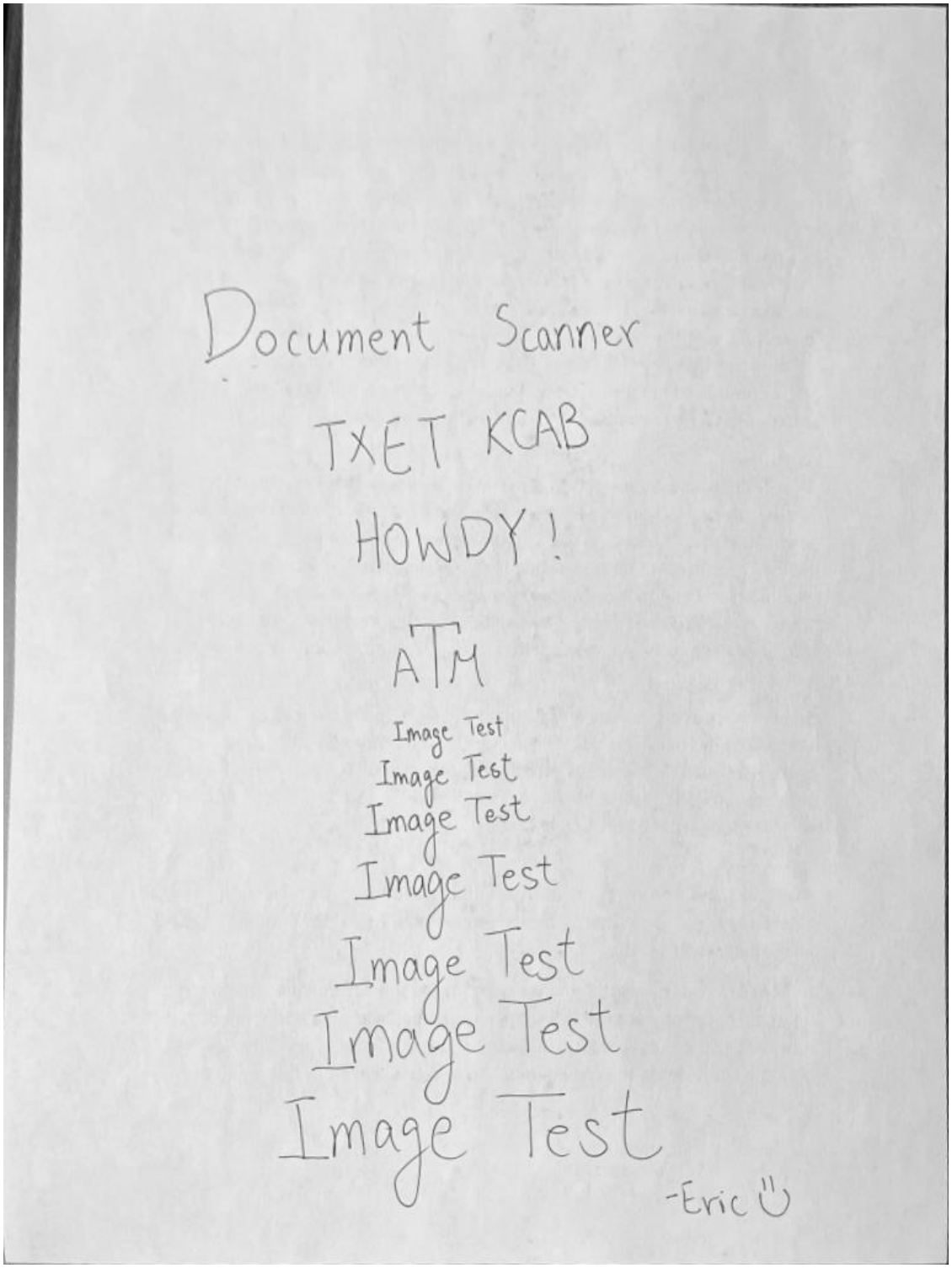


Figure 6 Warpped Image

Step 7:

The program applies the bottom-hat technique to get rid of the noise for better thresholding result. (Figure 7)

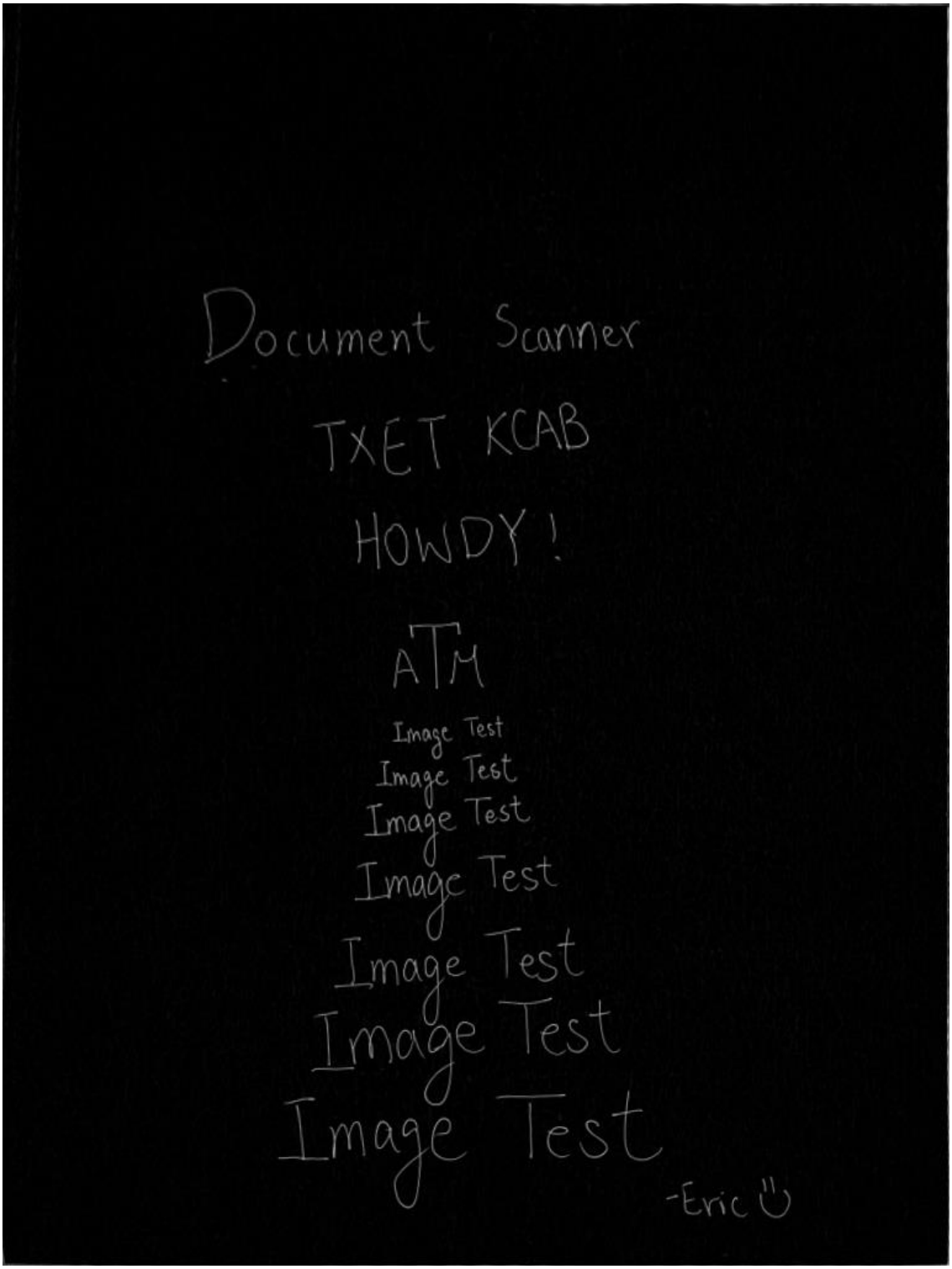


Figure 7 Bottom-hat Result

Step 8:

The program applies the inverted thresholding technique to get the final result. (Figure 8)

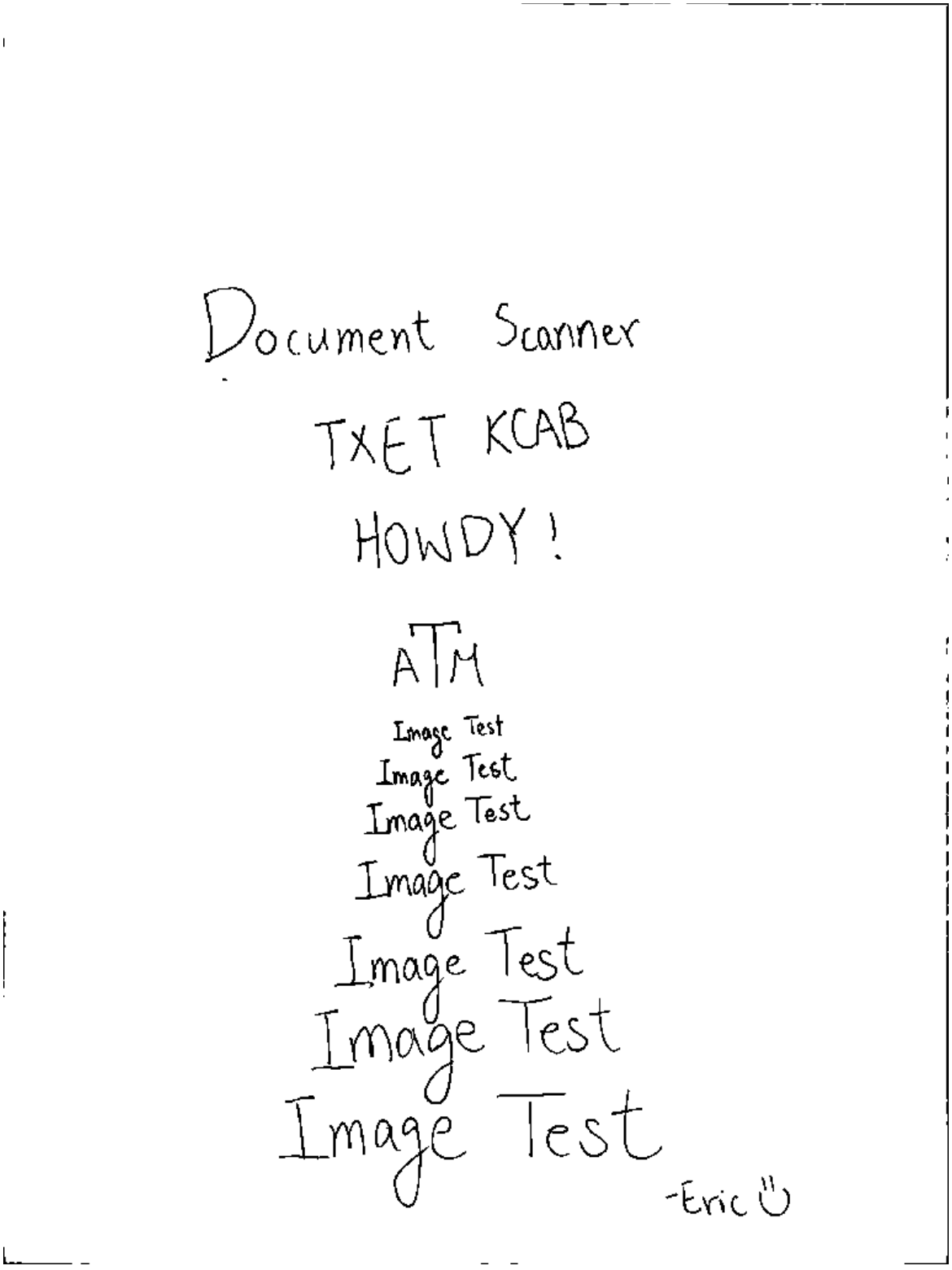
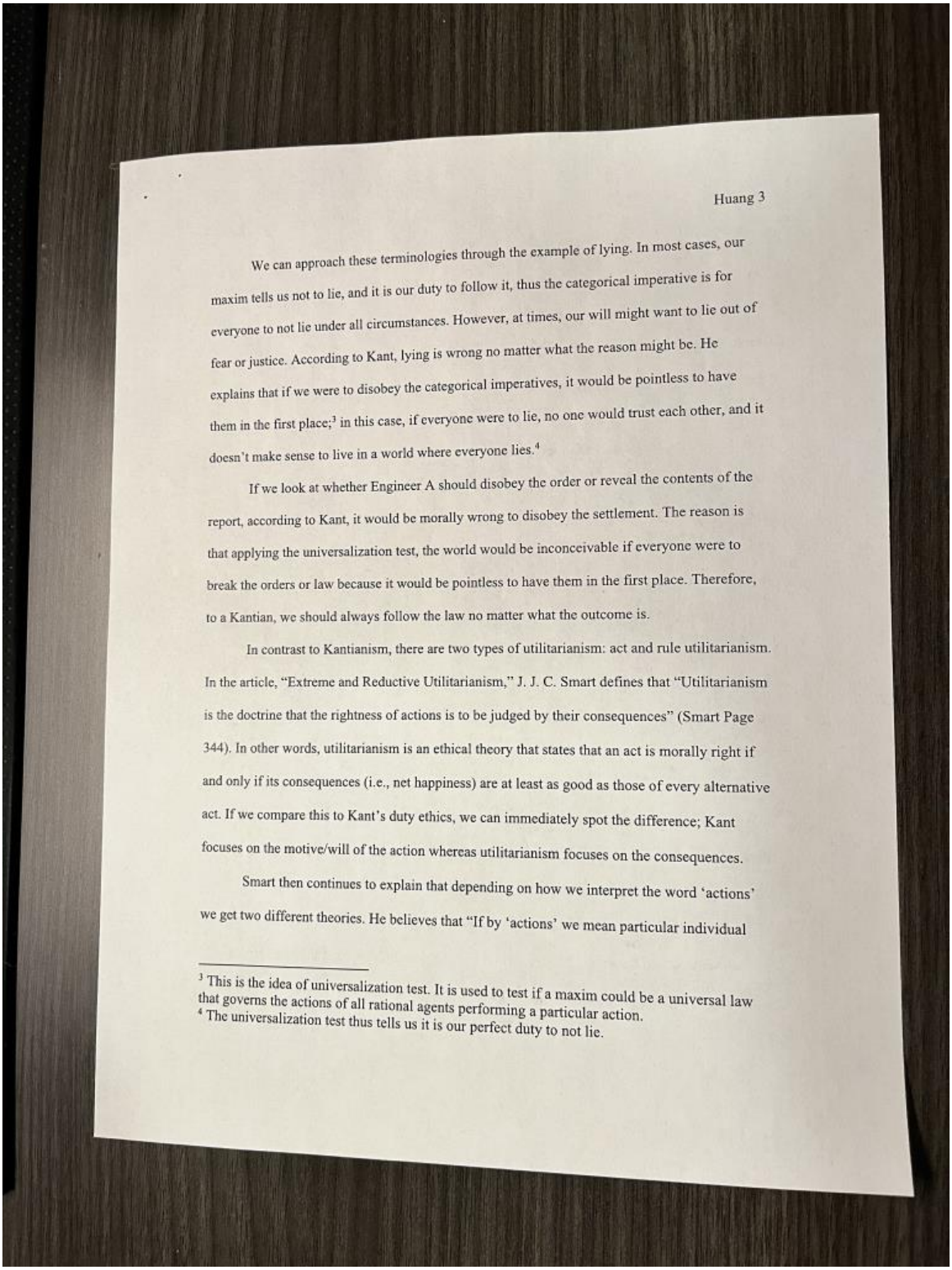


Figure 8 Final Result

Results of the other images:

Case 2:



Huang 3

We can approach these terminologies through the example of lying. In most cases, our maxim tells us not to lie, and it is our duty to follow it, thus the categorical imperative is for everyone to not lie under all circumstances. However, at times, our will might want to lie out of fear or justice. According to Kant, lying is wrong no matter what the reason might be. He explains that if we were to disobey the categorical imperatives, it would be pointless to have them in the first place;[3] in this case, if everyone were to lie, no one would trust each other, and it doesn't make sense to live in a world where everyone lies.[4]

If we look at whether Engineer A should disobey the order or reveal the contents of the report, according to Kant, it would be morally wrong to disobey the settlement. The reason is that applying the universalization test, the world would be inconceivable if everyone were to break the orders or law because it would be pointless to have them in the first place. Therefore, to a Kantian, we should always follow the law no matter what the outcome is.

In contrast to Kantianism, there are two types of utilitarianism: act and rule utilitarianism. In the article, "Extreme and Reductive Utilitarianism," J. J. C. Smart defines that "Utilitarianism is the doctrine that the rightness of actions is to be judged by their consequences" (Smart Page 344). In other words, utilitarianism is an ethical theory that states that an act is morally right if and only if its consequences (i.e., net happiness) are at least as good as those of every alternative act. If we compare this to Kant's duty ethics, we can immediately spot the difference; Kant focuses on the motive/will of the action whereas utilitarianism focuses on the consequences.

Smart then continues to explain that depending on how we interpret the word 'actions' we get two different theories. He believes that "If by 'actions' we mean particular individual

[3] This is the idea of universalization test. It is used to test if a maxim could be a universal law that governs the actions of all rational agents performing a particular action.
[4] The universalization test thus tells us it is our perfect duty to not lie.

Figure 9 Original Image

We can approach these terminologies through the example of lying. In most cases, our maxim tells us not to lie, and it is our duty to follow it, thus the categorical imperative is for everyone to not lie under all circumstances. However, at times, our will might want to lie out of fear or justice. According to Kant, lying is wrong no matter what the reason might be. He explains that if we were to disobey the categorical imperatives, it would be pointless to have them in the first place;[3] in this case, if everyone were to lie, no one would trust each other, and it doesn't make sense to live in a world where everyone lies.[4]

If we look at whether Engineer A should disobey the order or reveal the contents of the report, according to Kant, it would be morally wrong to disobey the settlement. The reason is that applying the universalization test, the world would be inconceivable if everyone were to break the orders or law because it would be pointless to have them in the first place. Therefore, to a Kantian, we should always follow the law no matter what the outcome is.

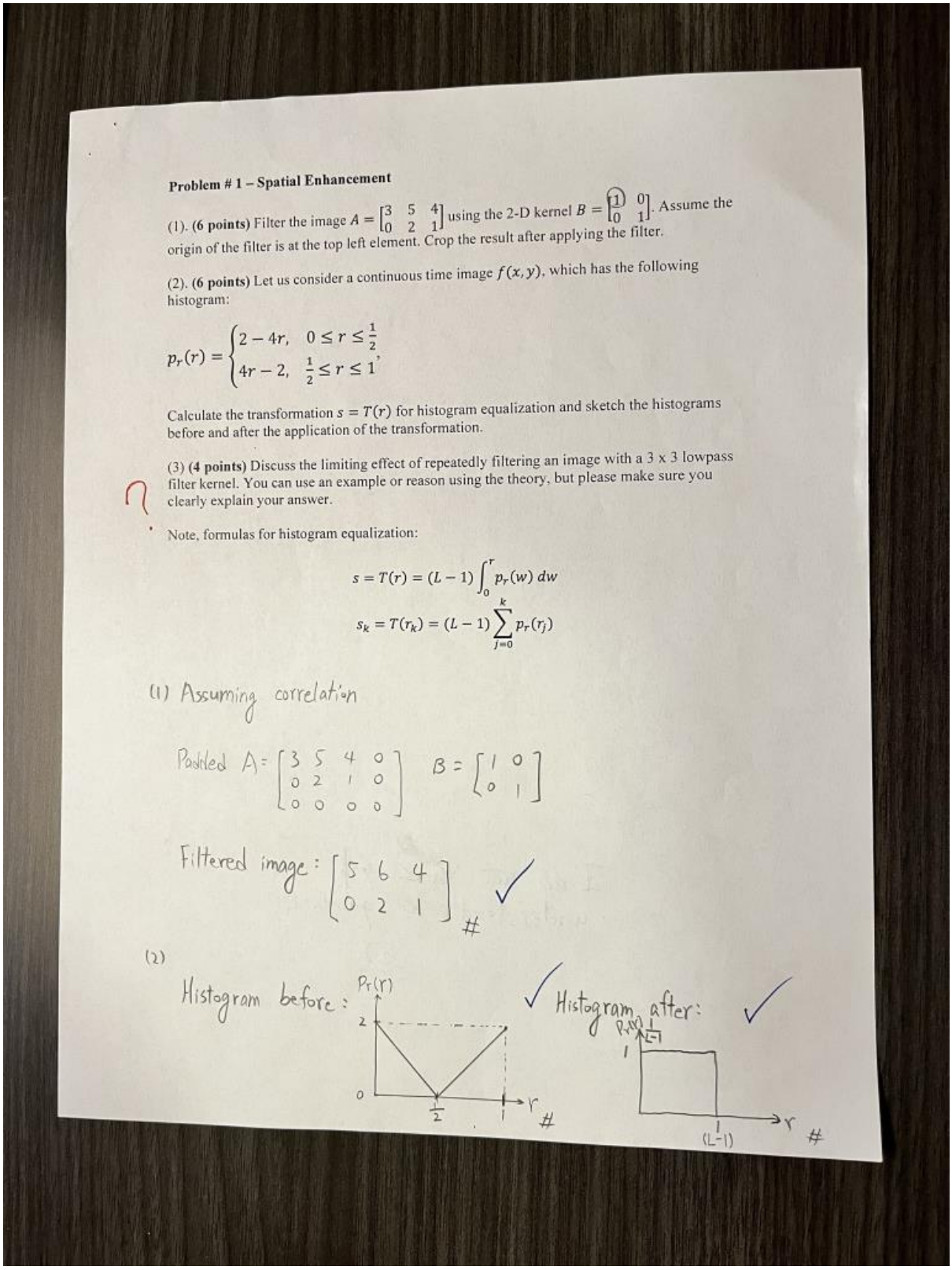
In contrast to Kantianism, there are two types of utilitarianism: act and rule utilitarianism. In the article, "Extreme and Reductive Utilitarianism," J. J. C. Smart defines that "Utilitarianism is the doctrine that the rightness of actions is to be judged by their consequences" (Smart Page 344). In other words, utilitarianism is an ethical theory that states that an act is morally right if and only if its consequences (i.e., net happiness) are at least as good as those of every alternative act. If we compare this to Kant's duty ethics, we can immediately spot the difference; Kant focuses on the motive/will of the action whereas utilitarianism focuses on the consequences.

Smart then continues to explain that depending on how we interpret the word 'actions' we get two different theories. He believes that "If by 'actions' we mean particular individual

[3] This is the idea of universalization test. It is used to test if a maxim could be a universal law that governs the actions of all rational agents performing a particular action.
[4] The universalization test thus tells us it is our perfect duty to not lie.

Figure 10 Case 2 Scanned

Case 3:



Problem # 1 – Spatial Enhancement

(1). **(6 points)** Filter the image $A = \begin{bmatrix} 3 & 5 & 4 \\ 0 & 2 & 1 \end{bmatrix}$ using the 2-D kernel $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Assume the origin of the filter is at the top left element. Crop the result after applying the filter.

(2). **(6 points)** Let us consider a continuous time image $f(x, y)$, which has the following histogram:

$$p_r(r) = \begin{cases} 2 - 4r, & 0 \le r \le \frac{1}{2} \\ 4r - 2, & \frac{1}{2} \le r \le 1 \end{cases},$$

Calculate the transformation $s = T(r)$ for histogram equalization and sketch the histograms before and after the application of the transformation.

(3) **(4 points)** Discuss the limiting effect of repeatedly filtering an image with a 3 x 3 lowpass filter kernel. You can use an example or reason using the theory, but please make sure you clearly explain your answer.

Note, formulas for histogram equalization:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

(1) Assuming correlation

Padded $A = \begin{bmatrix} 3 & 5 & 4 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Filtered image: $\begin{bmatrix} 5 & 6 & 4 \\ 0 & 2 & 1 \end{bmatrix}$ #

(2)

Histogram before: $P_r(r)$, 2, 0, $\frac{1}{2}$, r #

Histogram after: $P_s(s)$, $\frac{1}{L-1}$, 1, $\frac{1}{(L-1)}$, r #

Figure 11 Original Image

Problem # 1 – Spatial Enhancement

(1). **(6 points)** Filter the image $A = \begin{bmatrix} 3 & 5 & 4 \\ 0 & 2 & 1 \end{bmatrix}$ using the 2-D kernel $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Assume the origin of the filter is at the top left element. Crop the result after applying the filter.

(2). **(6 points)** Let us consider a continuous time image $f(x, y)$, which has the following histogram:

$$p_r(r) = \begin{cases} 2 - 4r, & 0 \le r \le \frac{1}{2} \\ 4r - 2, & \frac{1}{2} \le r \le 1 \end{cases}$$

Calculate the transformation $s = T(r)$ for histogram equalization and sketch the histograms before and after the application of the transformation.

(3) **(4 points)** Discuss the limiting effect of repeatedly filtering an image with a 3 x 3 lowpass filter kernel. You can use an example or reason using the theory, but please make sure you clearly explain your answer.

Note, formulas for histogram equalization:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

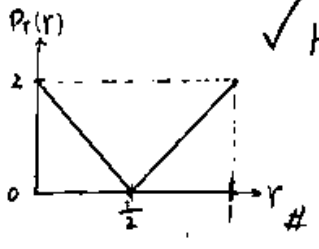
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

(1) Assuming correlation

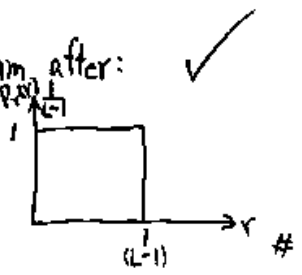
Padded $A = \begin{bmatrix} 3 & 5 & 4 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Filtered image: $\begin{bmatrix} 5 & 6 & 4 \\ 0 & 2 & 1 \end{bmatrix}$ # ✓

(2)

Histogram before: 

✓

Histogram after: 

✓

Figure 12 Case 3 Scanned

Case 4:

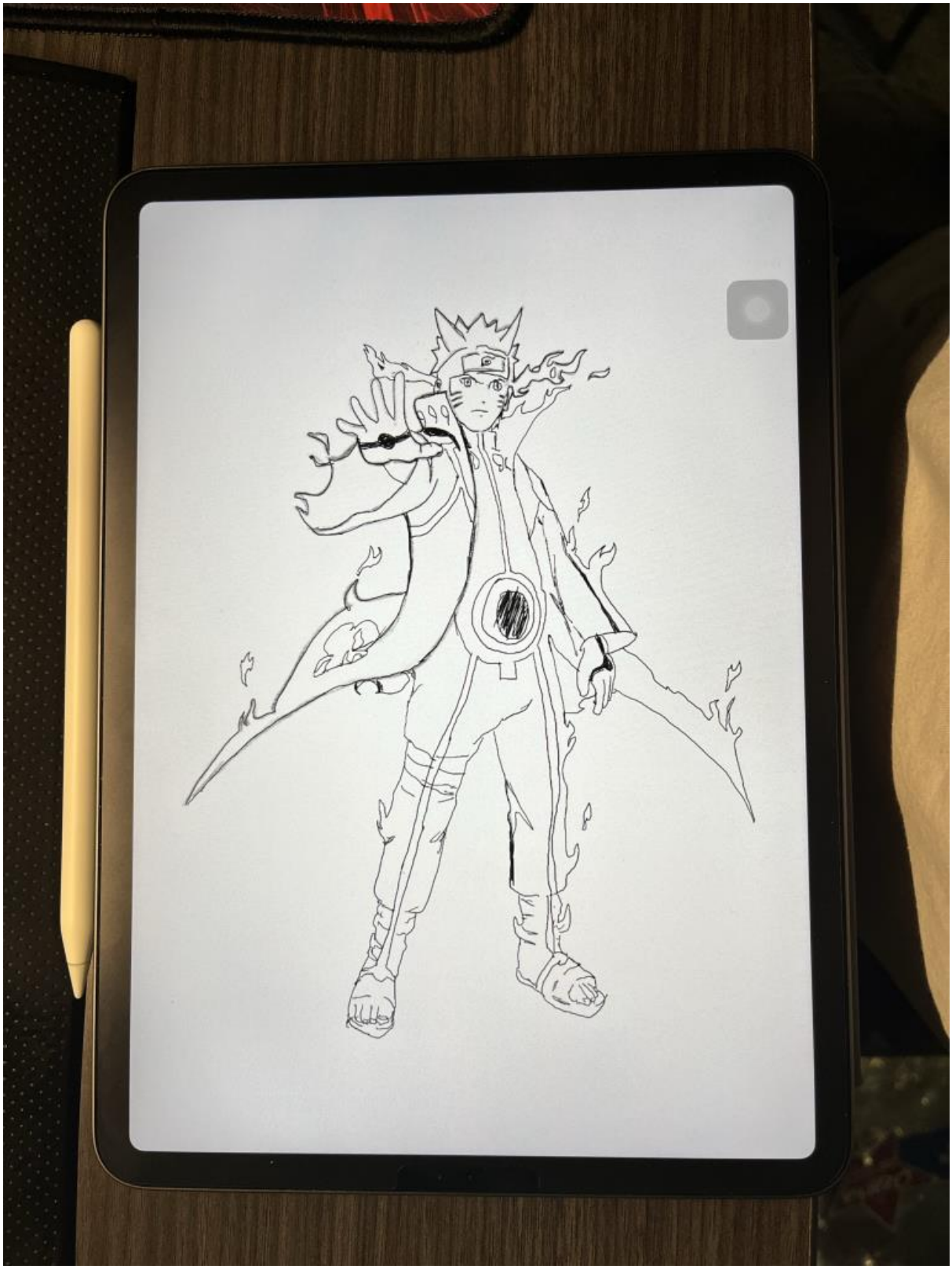


Figure 13 Original Image

Figure 14 Case 4 Scanned

Case 5:

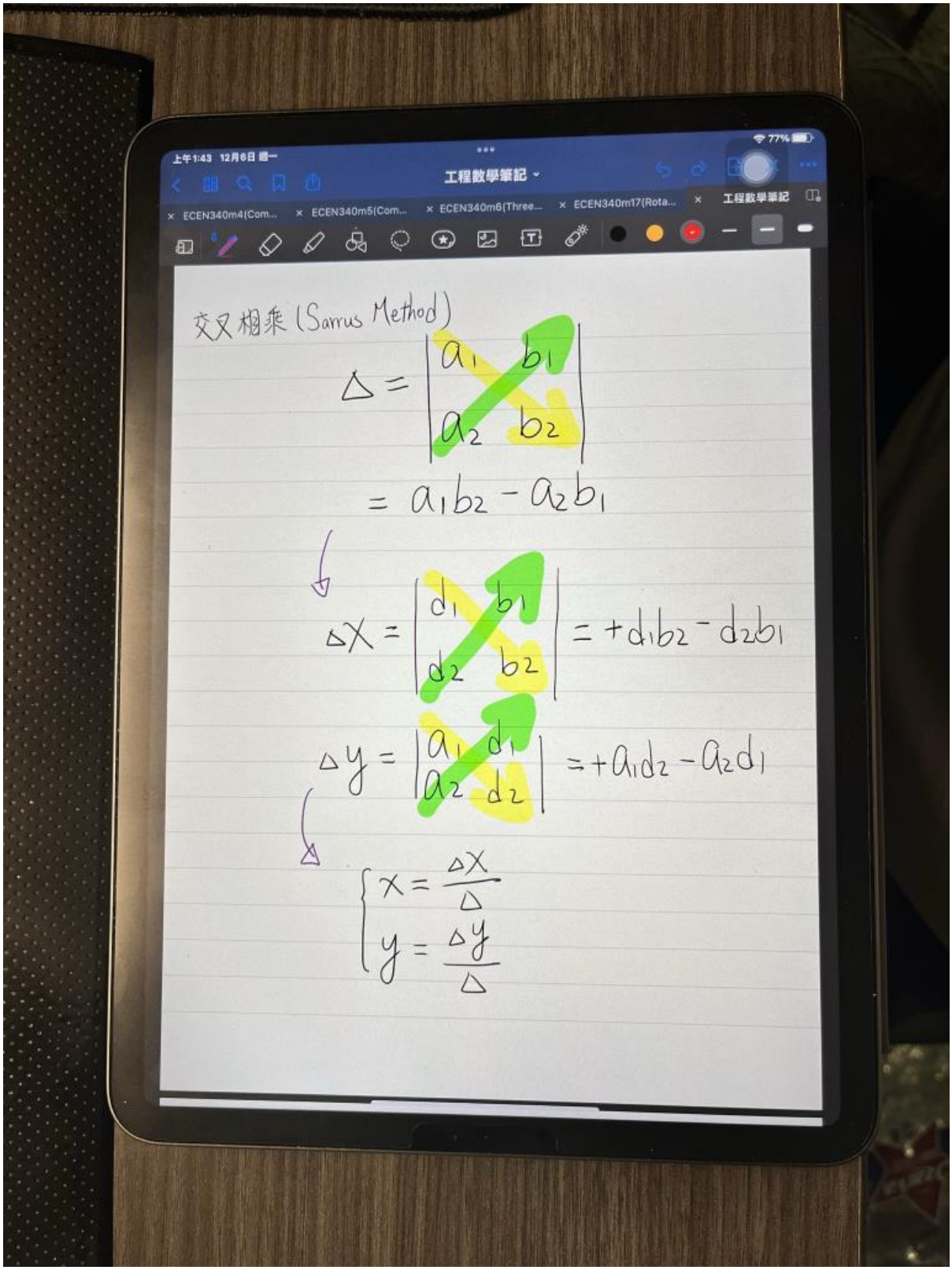


Figure 15 Original Image

交叉相乘 (Sarrus Method)

$$\Delta = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1$$

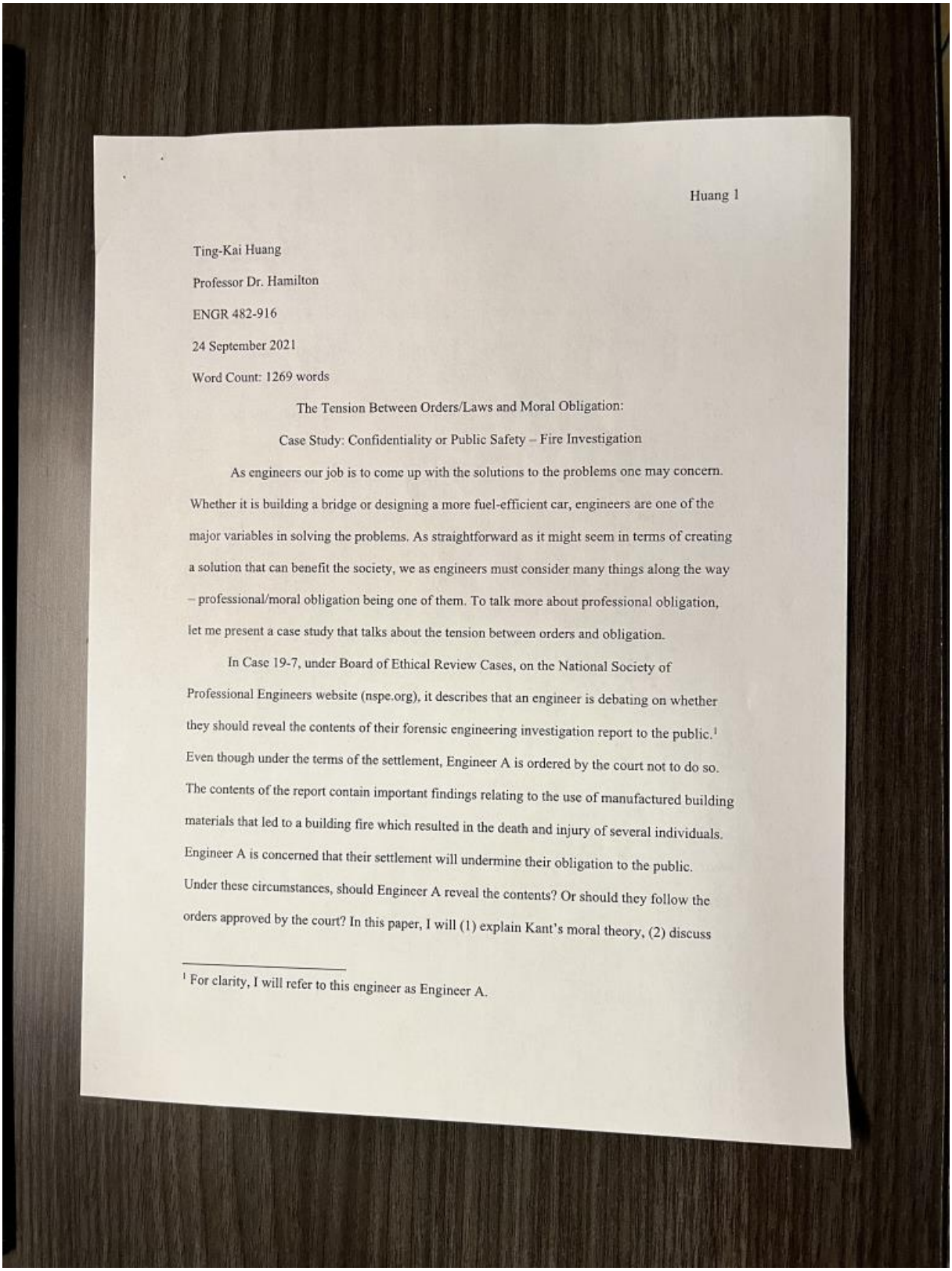
$$\Delta x = \begin{vmatrix} d_1 & b_1 \\ d_2 & b_2 \end{vmatrix} = +d_1 b_2 - d_2 b_1$$

$$\Delta y = \begin{vmatrix} a_1 & d_1 \\ a_2 & d_2 \end{vmatrix} = +a_1 d_2 - a_2 d_1$$

$$\begin{cases} x = \dfrac{\Delta x}{\Delta} \\ y = \dfrac{\Delta y}{\Delta} \end{cases}$$

Figure 16 Case 5 Scanned

Case 6:



Ting-Kai Huang

Professor Dr. Hamilton

ENGR 482-916

24 September 2021

Word Count: 1269 words

The Tension Between Orders/Laws and Moral Obligation:

Case Study: Confidentiality or Public Safety – Fire Investigation

As engineers our job is to come up with the solutions to the problems one may concern. Whether it is building a bridge or designing a more fuel-efficient car, engineers are one of the major variables in solving the problems. As straightforward as it might seem in terms of creating a solution that can benefit the society, we as engineers must consider many things along the way – professional/moral obligation being one of them. To talk more about professional obligation, let me present a case study that talks about the tension between orders and obligation.

In Case 19-7, under Board of Ethical Review Cases, on the National Society of Professional Engineers website (nspe.org), it describes that an engineer is debating on whether they should reveal the contents of their forensic engineering investigation report to the public.[1] Even though under the terms of the settlement, Engineer A is ordered by the court not to do so. The contents of the report contain important findings relating to the use of manufactured building materials that led to a building fire which resulted in the death and injury of several individuals. Engineer A is concerned that their settlement will undermine their obligation to the public. Under these circumstances, should Engineer A reveal the contents? Or should they follow the orders approved by the court? In this paper, I will (1) explain Kant's moral theory, (2) discuss

[1] For clarity, I will refer to this engineer as Engineer A.

Figure 17 Original Image

Ting-Kai Huang

Professor Dr. Hamilton

ENGR 482-916

24 September 2021

Word Count: 1269 words

The Tension Between Orders/Laws and Moral Obligation:

Case Study: Confidentiality or Public Safety – Fire Investigation

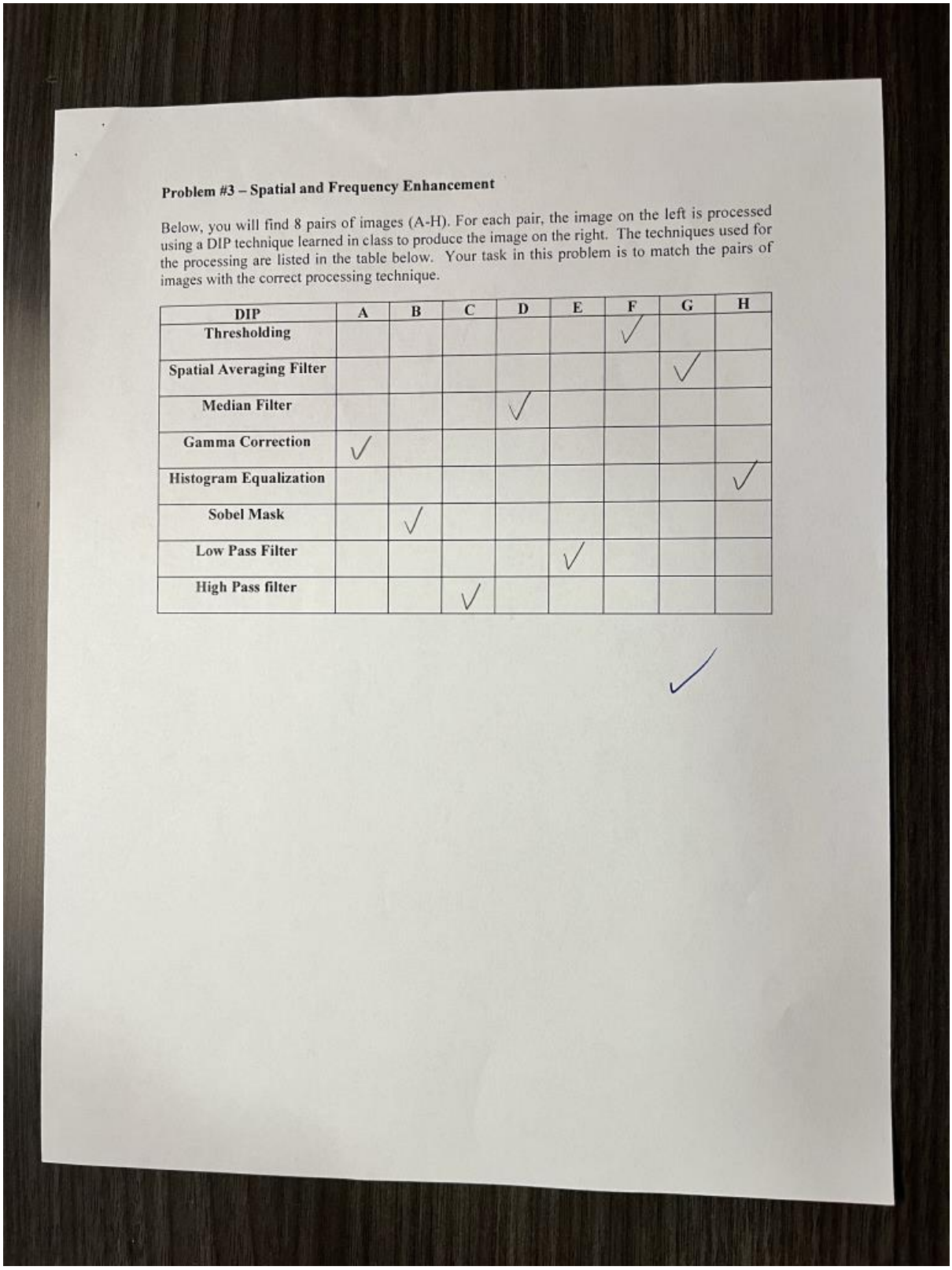
As engineers our job is to come up with the solutions to the problems one may concern. Whether it is building a bridge or designing a more fuel-efficient car, engineers are one of the major variables in solving the problems. As straightforward as it might seem in terms of creating a solution that can benefit the society, we as engineers must consider many things along the way – professional/moral obligation being one of them. To talk more about professional obligation, let me present a case study that talks about the tension between orders and obligation.

In Case 19-7, under Board of Ethical Review Cases, on the National Society of Professional Engineers website (nspe.org), it describes that an engineer is debating on whether they should reveal the contents of their forensic engineering investigation report to the public.[1] Even though under the terms of the settlement, Engineer A is ordered by the court not to do so. The contents of the report contain important findings relating to the use of manufactured building materials that led to a building fire which resulted in the death and injury of several individuals. Engineer A is concerned that their settlement will undermine their obligation to the public. Under these circumstances, should Engineer A reveal the contents? Or should they follow the orders approved by the court? In this paper, I will (1) explain Kant's moral theory, (2) discuss

[1] For clarity, I will refer to this engineer as Engineer A.

Figure 18 Case 6 Scanned

Case 7:



Problem #3 – Spatial and Frequency Enhancement

Below, you will find 8 pairs of images (A-H). For each pair, the image on the left is processed using a DIP technique learned in class to produce the image on the right. The techniques used for the processing are listed in the table below. Your task in this problem is to match the pairs of images with the correct processing technique.

DIP	A	B	C	D	E	F	G	H
Thresholding						✓		
Spatial Averaging Filter							✓	
Median Filter				✓				
Gamma Correction	✓							
Histogram Equalization								✓
Sobel Mask		✓						
Low Pass Filter					✓			
High Pass filter			✓					

✓

Figure 19 Original Image

Problem #3 – Spatial and Frequency Enhancement

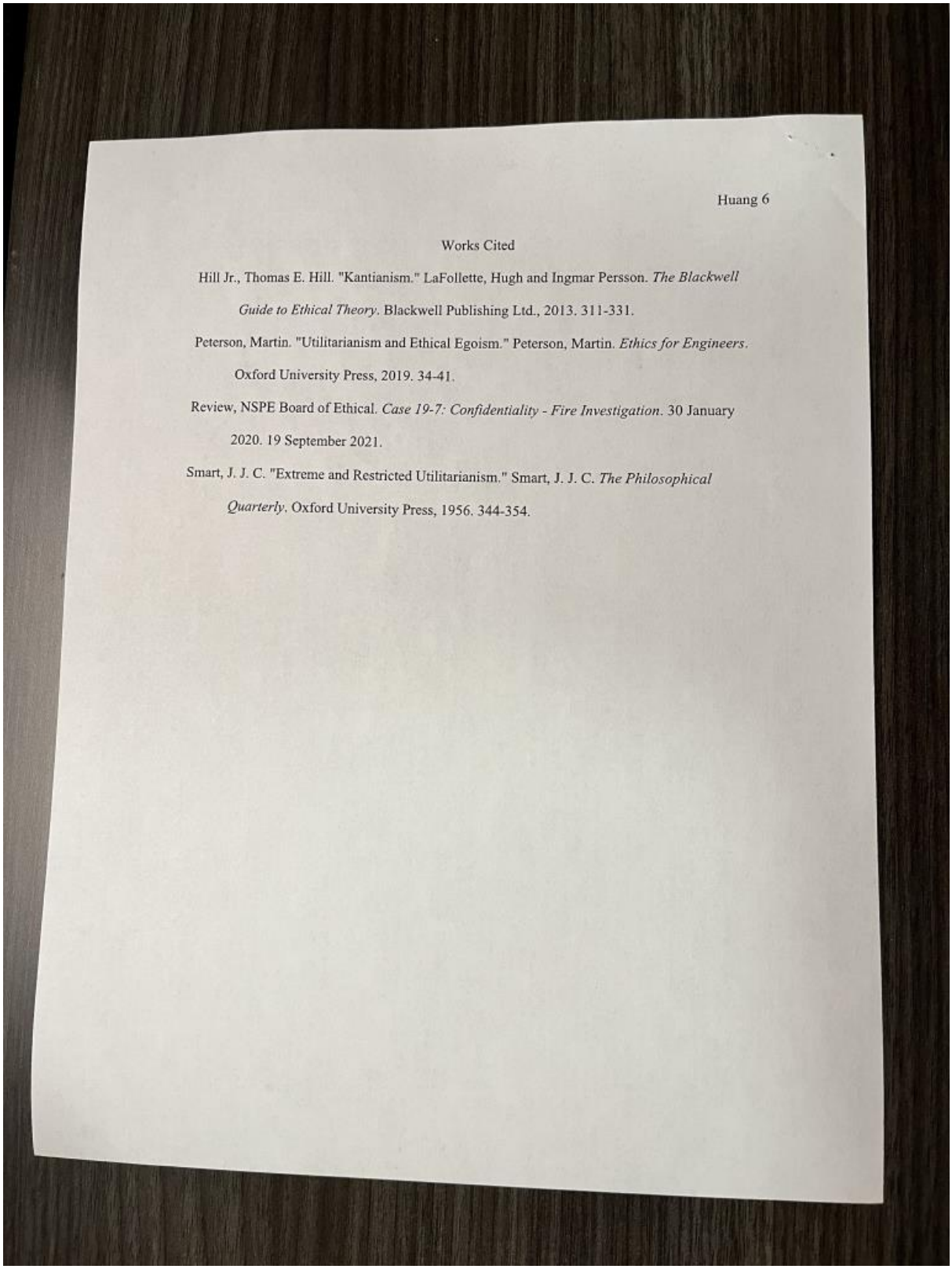
Below, you will find 8 pairs of images (A-H). For each pair, the image on the left is processed using a DIP technique learned in class to produce the image on the right. The techniques used for the processing are listed in the table below. Your task in this problem is to match the pairs of images with the correct processing technique.

DIP	A	B	C	D	E	F	G	H
Thresholding						✓		
Spatial Averaging Filter							✓	
Median Filter				✓				
Gamma Correction	✓							
Histogram Equalization								✓
Sobel Mask		✓						
Low Pass Filter					✓			
High Pass filter			✓					

✓

Figure 20 Case 7 Scanned

Case 8:



Works Cited

Hill Jr., Thomas E. Hill. "Kantianism." LaFollette, Hugh and Ingmar Persson. *The Blackwell Guide to Ethical Theory*. Blackwell Publishing Ltd., 2013. 311-331.

Peterson, Martin. "Utilitarianism and Ethical Egoism." Peterson, Martin. *Ethics for Engineers*. Oxford University Press, 2019. 34-41.

Review, NSPE Board of Ethical. *Case 19-7: Confidentiality - Fire Investigation*. 30 January 2020. 19 September 2021.

Smart, J. J. C. "Extreme and Restricted Utilitarianism." Smart, J. J. C. *The Philosophical Quarterly*. Oxford University Press, 1956. 344-354.

Figure 21 Original Image

Works Cited

Hill Jr., Thomas E. Hill. "Kantianism." LaFollette, Hugh and Ingmar Persson. *The Blackwell Guide to Ethical Theory*. Blackwell Publishing Ltd., 2013. 311-331.

Peterson, Martin. "Utilitarianism and Ethical Egoism." Peterson, Martin. *Ethics for Engineers*. Oxford University Press, 2019. 34-41.

Review, NSPE Board of Ethical. *Case 19-7: Confidentiality - Fire Investigation*. 30 January 2020. 19 September 2021.

Smart, J. J. C. "Extreme and Restricted Utilitarianism." Smart, J. J. C. *The Philosophical Quarterly*. Oxford University Press, 1956. 344-354.

Figure 22 Case 8 Scanned

Case 9: (Different from Case 4)

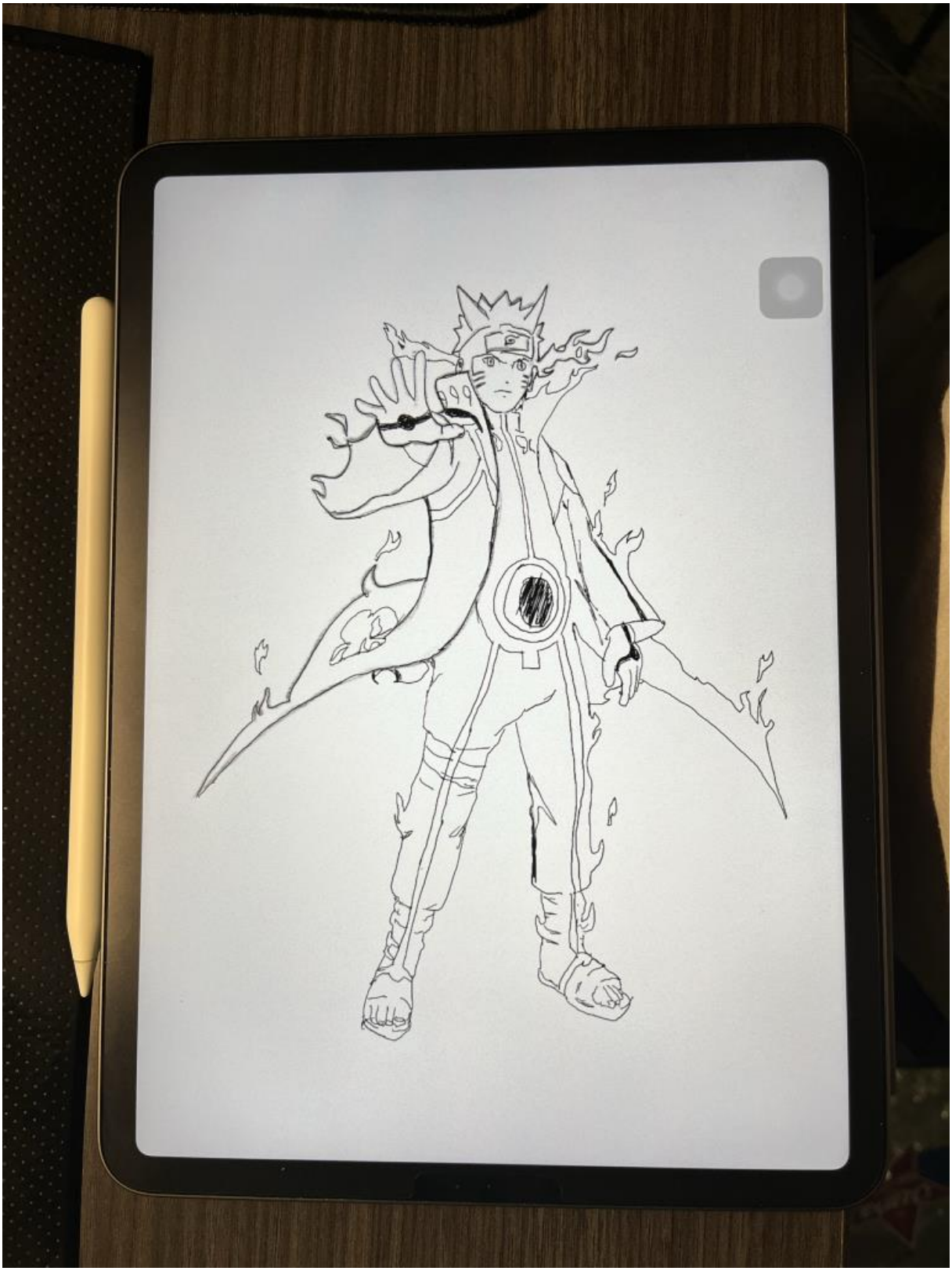


Figure 23 Original Image

Figure 24 Case 9 Scanned

Case 10: (Different from Case 4 and Case 9)

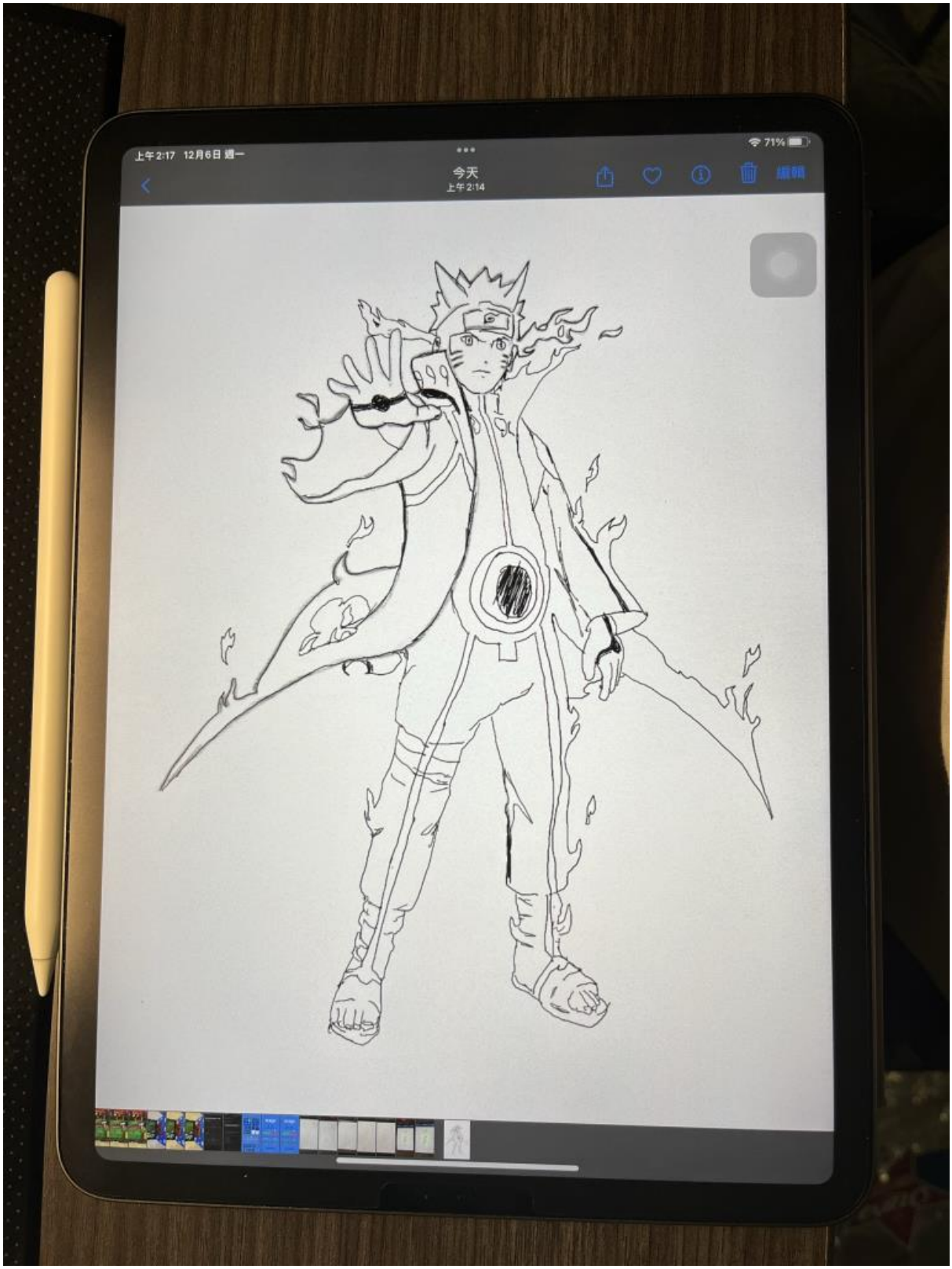


Figure 25 Original Image

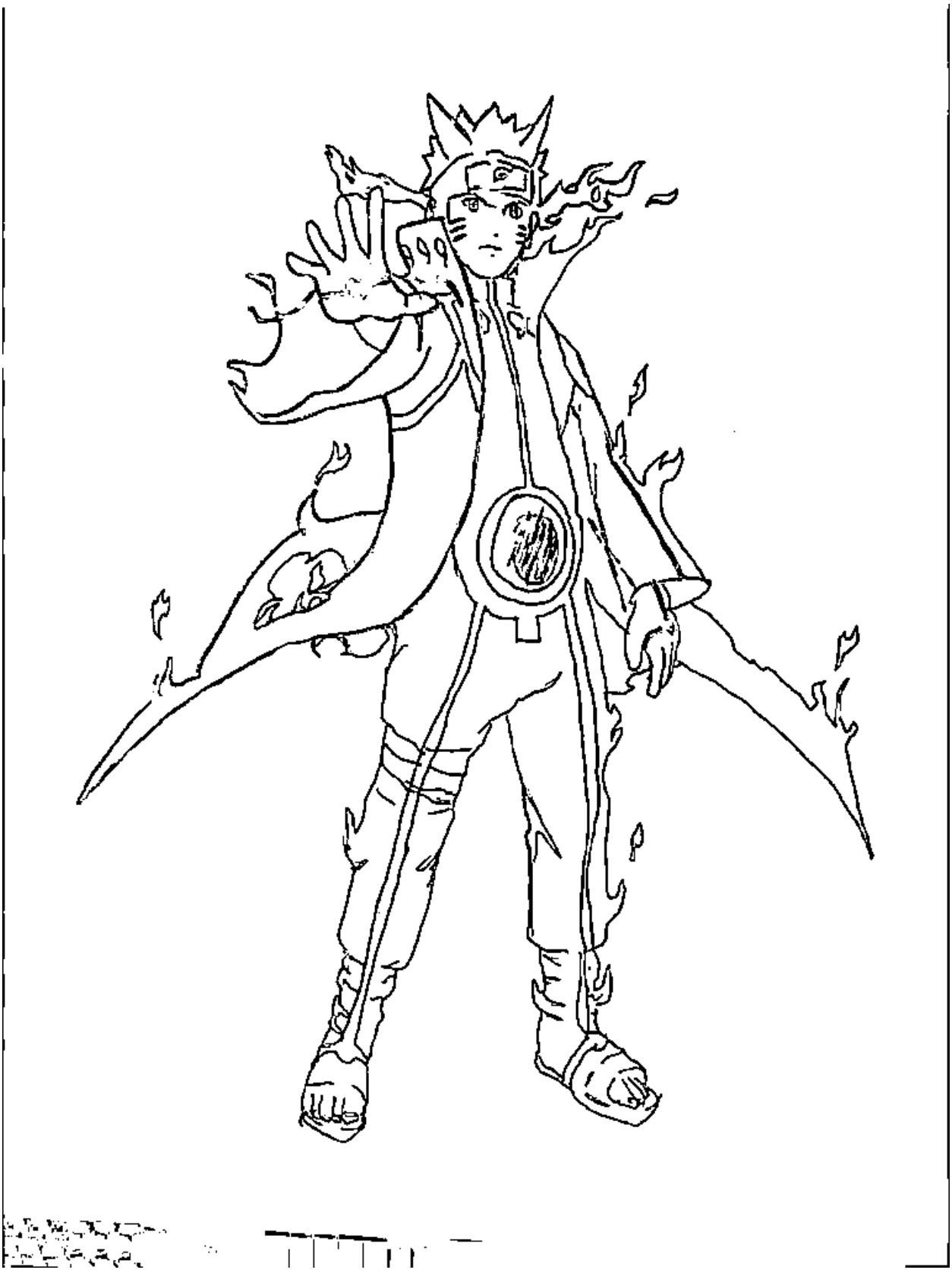


Figure 26 Case 10 Scanned

Case 11:

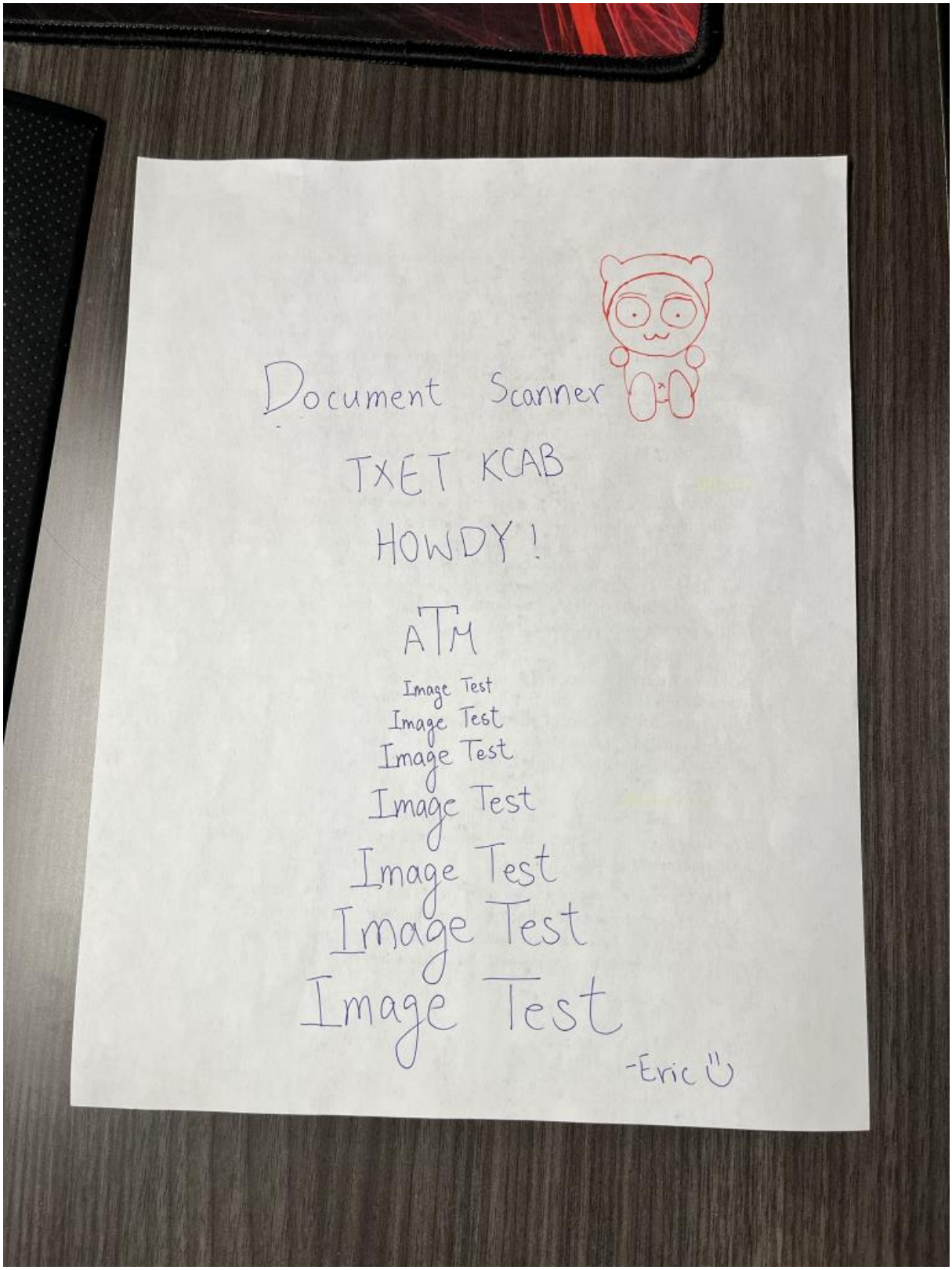


Figure 27 Original Image

Document Scanner

TXET KCAB

HOWDY!

ATM

Image Test
Image Test
Image Test
Image Test
Image Test
Image Test
Image Test

-Eric

Figure 28 Case 11 Scanned

Appendix

Appendix A: Program Codes (Python)

```
import cv2
import numpy as np

# Get image contours
def getContours(img):
    contours, hierarchy =
cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    max_contour = []
    max_area = 0
    for cnt in contours:
        #get contour area
        area = cv2.contourArea(cnt)
        if area > max_area:
            max_area = area
            max_contour = cnt
    perimeter = cv2.arcLength(max_contour,True)
    # print(perimeter)
    # print(max_area)
    return max_contour, perimeter

# Get contour corner points
def getCornerPoints(contour, peri):
    approx = cv2.approxPolyDP(contour,0.02 * peri,True)
    return approx

img_name_list = [f'IMG_{x}.JPEG' for x in range(10,12)]
path_list = [f"C:\\Users\\Eric\\iCloudDrive\\Desktop\\Python3\\Digital Image
Processing\\iCloud Photos\\{x}" for x in img_name_list]
path_list2 = [f"C:\\Users\\Eric\\iCloudDrive\\Desktop\\Python3\\Digital Image
Processing\\iCloud Photos\\{x}" for x in img_name_list]
thres1 = 200
thres2 = 200

while True:
    for path in path_list:
        try:
            img = cv2.imread(path)
            img_Resized = cv2.resize(img,(768,1024))
            imgContour = img_Resized.copy()

            # Pre-processing
            # Convert the image to a gray scale image
```

```
            imgGray = cv2.cvtColor(img_Resized,cv2.COLOR_BGR2GRAY)

            # Apply the Gaussian Blur
            imgBlur = cv2.GaussianBlur(imgGray,(5,5),1)

            # Apply Canny filter to get the edges
            imgCanny = cv2.Canny(imgBlur,thres1,thres2)

            # make the edges thicker to better detect the document
            kernel_ones = np.ones((5,5))
            edge_dilate = cv2.dilate(imgCanny, kernel_ones, iterations=2)
            edge_erode = cv2.erode(edge_dilate,kernel_ones, iterations=1)

            contour, perimeter = getContours(edge_erode)
            corner_points = getCornerPoints(contour, perimeter)
            # print(corner_points[1][0])

            # new image size
            width,height = img_Resized.shape[1], img_Resized.shape[0]

            # Make sure the corner points are in the right order for warping the
image.
            min = 1280 * 768
            max = 0
            max_list = []
            min_list = []
            width_list = []
            height_list = []
            corner_points_list = []

            for i in range(4):
                corner_points_list.append(corner_points[i][0])

            for i in corner_points_list:
                if i[0] * i[-1] > max:
                    max = i[0] * i[-1]
                    max_list = list(i)
                if i[0] * i[-1] < min:
                    min = i[0] * i[-1]
                    min_list = list(i)

            for i in corner_points_list:
                if list(i) != max_list and list(i) != min_list:
                    if i[0] > i[-1]:
                        width_list = list(i)
```

```
                    else:
                        height_list = list(i)

            new_points = [min_list,width_list,height_list,max_list]
            # print(new_points)

            # Create an image with only the paper (stretched out image of the
paper)
            # It is crucial to have the points in the right order, that's why we
reordered the points earlier
            original_points =
np.float32([new_points[0],new_points[1],new_points[2],new_points[3]])
            mapped_new_corners =
np.float32([[0,0],[width,0],[0,height],[width,height]]) # min, width, height, max
(has to be in this order)
            matrix =
cv2.getPerspectiveTransform(original_points,mapped_new_corners) # warp the image
so that the resulting image only has the document itself
            stretched_paper = cv2.warpPerspective(imgGray,matrix,(width,height))

            # Try bottom-hat
            dilated = cv2.dilate(stretched_paper, kernel_ones, iterations=1)
            closed = cv2.erode(dilated, kernel_ones, iterations=1)
            bot_hat = closed - stretched_paper

            # Image Thresholding
            _, result = cv2.threshold(bot_hat,35,255,cv2.THRESH_BINARY_INV)

            cv2.imshow(f"Original Image {path_list2.index(path)+1}",img_Resized)
            cv2.imshow("Gray Scale Image",imgGray)
            cv2.imshow("Gaussian Blur",imgBlur)
            cv2.imshow("Edge Dectection",imgCanny)
            cv2.imshow("Eroded Edge", edge_erode)
            cv2.imshow("Warped Image", stretched_paper)
            cv2.imshow("Bottom Hat", bot_hat)
            cv2.imshow(f"Result Image {path_list2.index(path)+1}",result)
            cv2.waitKey(0)
            # if the image passed, aka didn't throw the ValueError, then remove
the image from the list
            path_list.remove(path)
        except ValueError:
            pass

    # if every image in the list passed, aka no ValueErrors, then break out the
while loop
```

```
    if len(path_list) == 0:
        break
    # else, use different threshold values for the canny edge detection until
successfully detect the document
    else:
        thres1 -= 25
        thres2 -= 25
```