DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION
UNIVERSITY OF MORATUWA

EN3160 : Image Processing and Machine Vision

Intensity Transformations and Neighborhood Filtering - assignment

200687M : Vishwamith P.G.H

$01^{st} of\, September, 2023$

Contents

1 Question one **3**

2 Question two **4**

3 Question Three **5**

4 Question four **6**

5 Question five **8**

6 Question six **9**

7 Question seven **11**

8 Question eight **13**

9 Question nine **14**

1 Question one

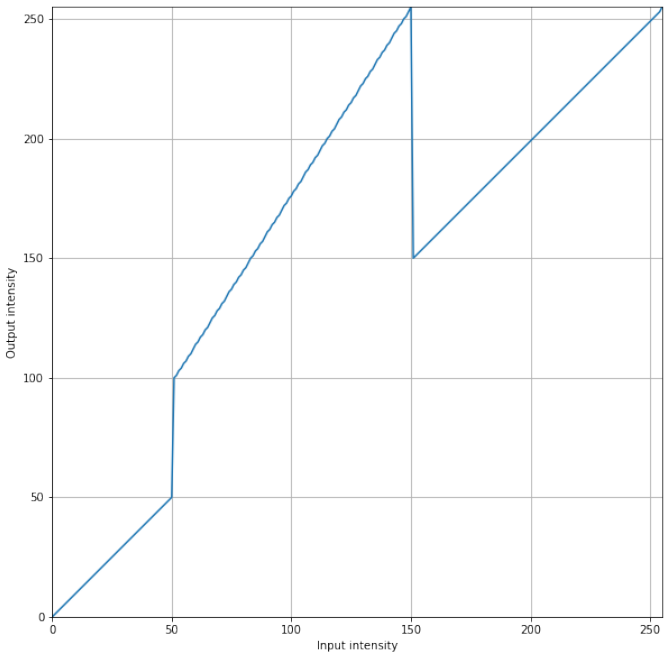
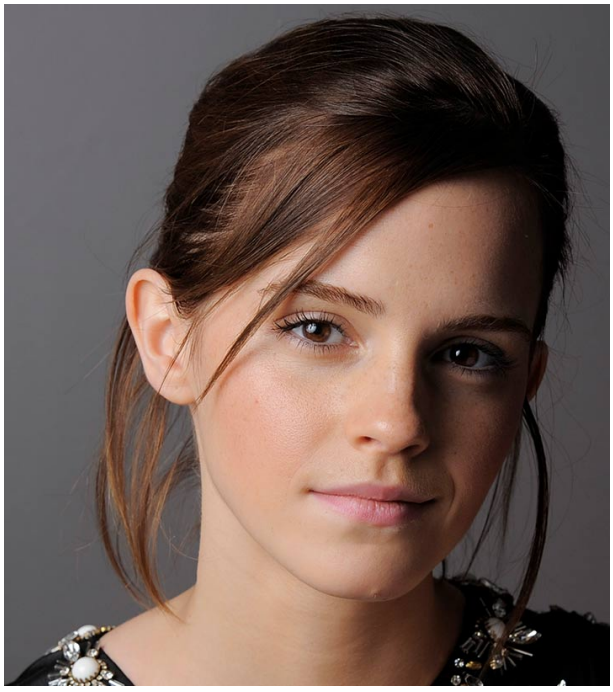


Figure 1: Intensity Transformation



(a) Original Image

(b) Trnasformed Image

2 Question two

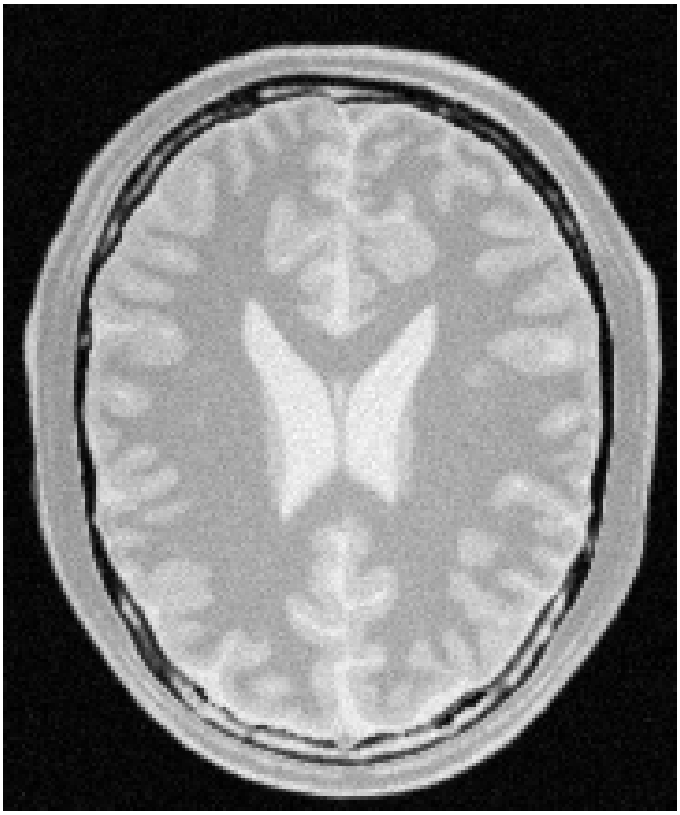
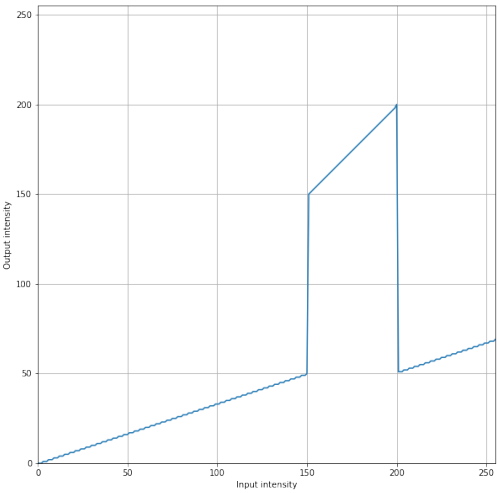
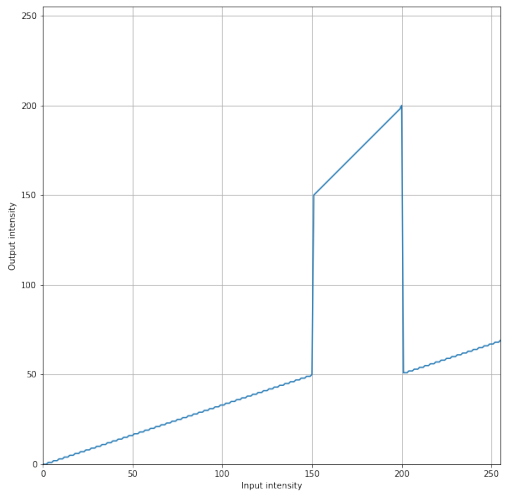


Figure 3: Original Image

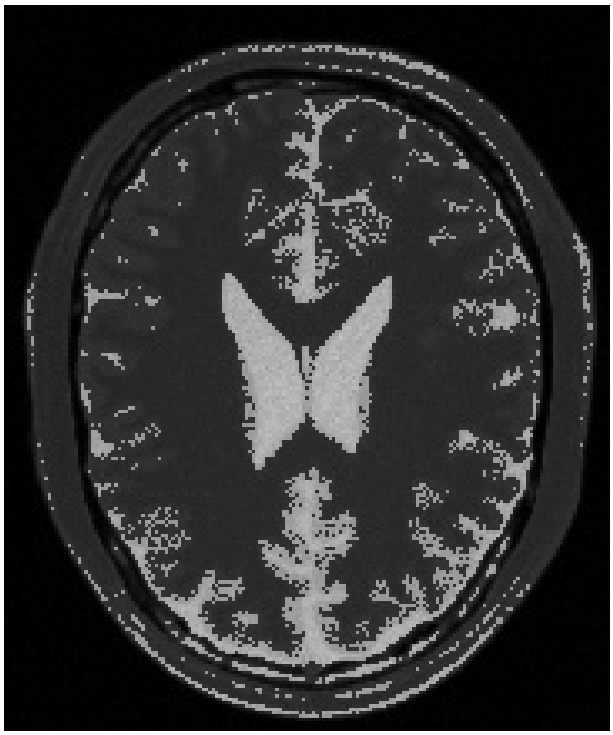


(a) Intensity Transform to Enhance Grayish Matter

(b) Grayish Matter Enhanced Image



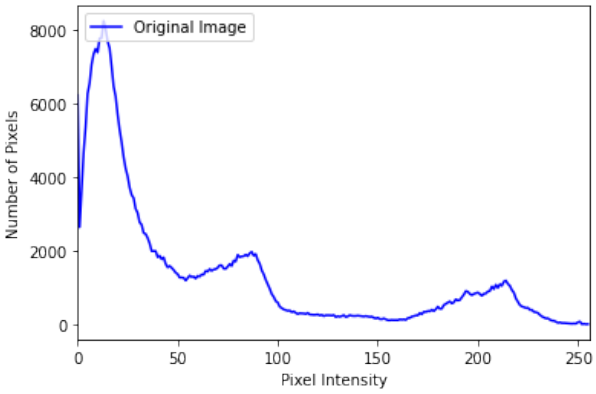
(a) Intensity Transform to Enhance Whitish Matter



(b) Whitish Matter Enhanced Image

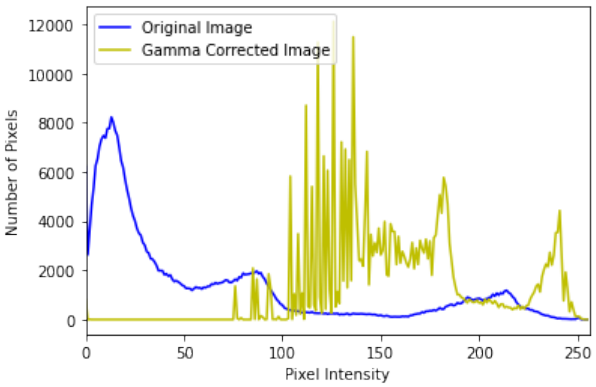
3 Question Three

(a) Image Before Gamma Correction



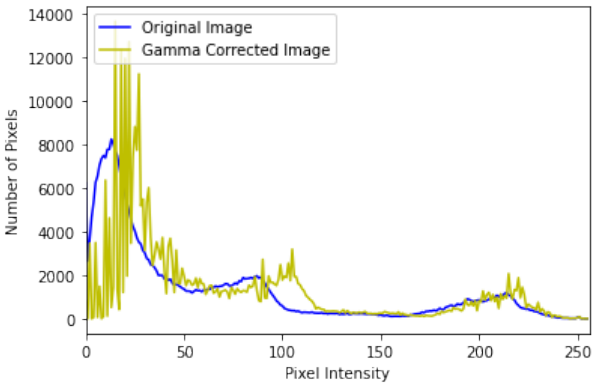
(b) Color Histogram

(a) Gamma = 0.2



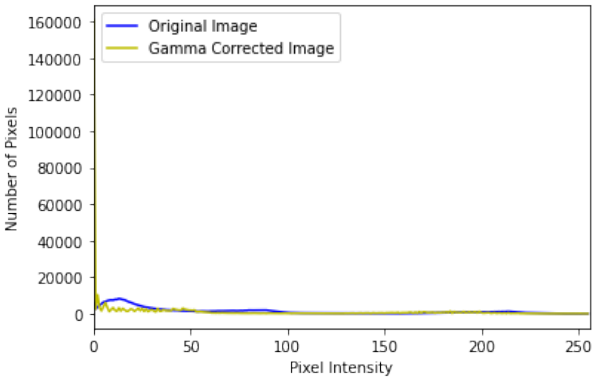
(b) Color Histogram

(a) Gamma = 0.8



(b) Color Histogram

(a) Gamma = 2



(b) Color Histogram

The 0.8 is the best choice for gamma value

4 Question four

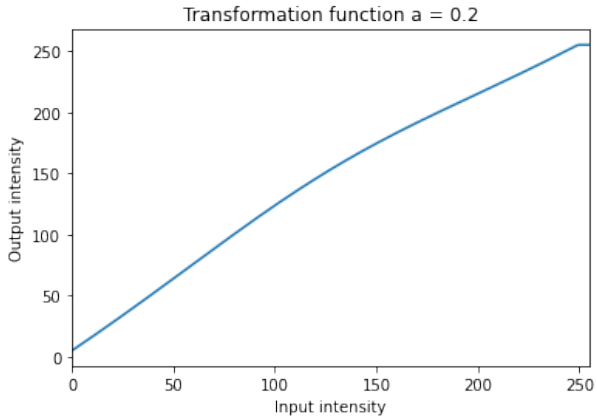
(a) Original Image

(b) Hue Image

(a) Saturation

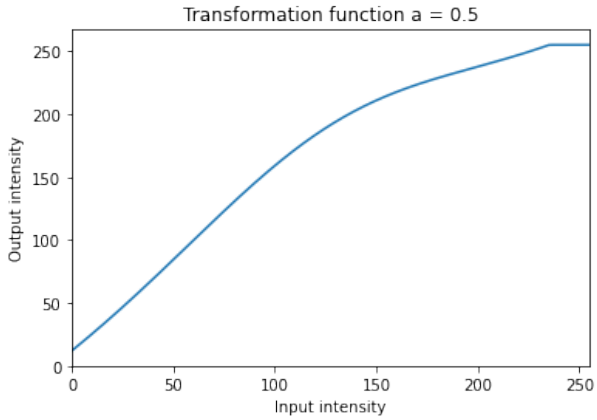
(b) Value Image

(a) Enhanced Image a = 0.2



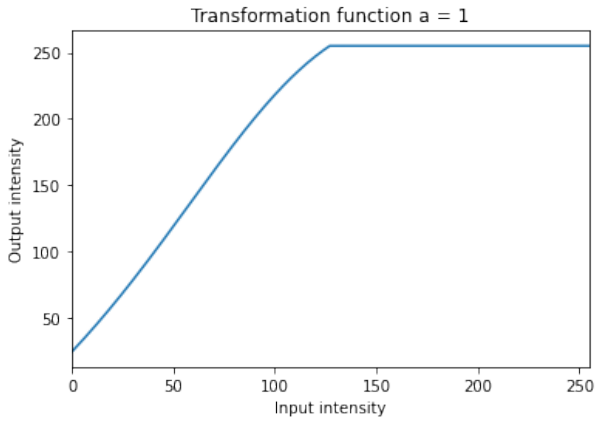
(b) Intensity Transform a = 0.2

(a) Enhanced Image a = 0.5



(b) Intensity Transform a = 0.5

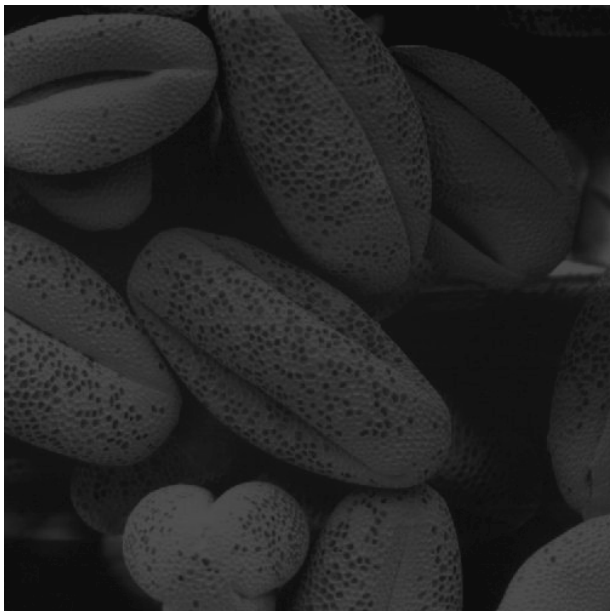
(a) Enhanced Image a = 1



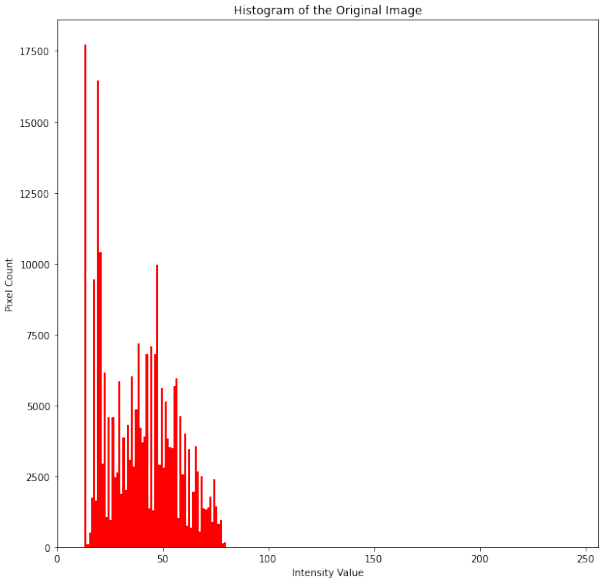
(b) Intensity Transform a = 1

The suitable alpha value is 0.5

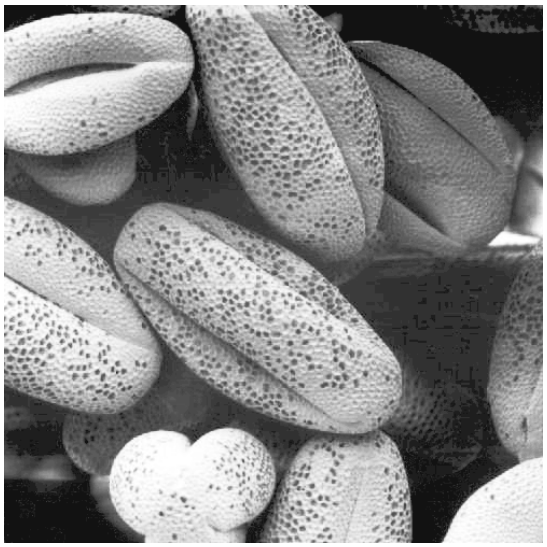
5 Question five



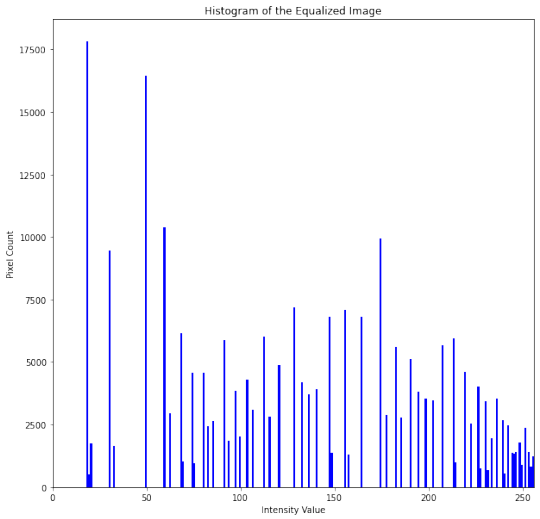
(a) Original Image



(b) Color Histogram

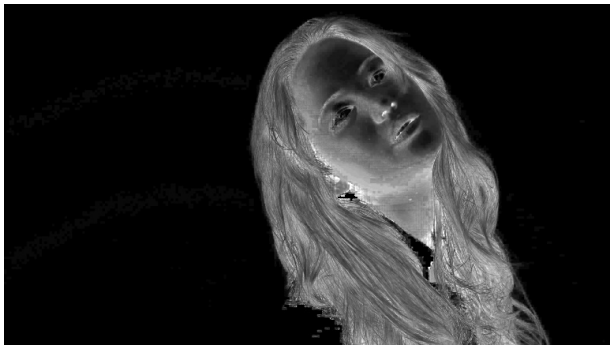


(a) Histogram Equalized Image



(b) Color Equalized Histogram

6 Question six



(a) Saturation Image

(b) Hue Image

(a) Value Image

(b) Foreground Only Image

The plane to extract the foreground is the saturation plane.

```
array([222621, 222630, 222641, 222659, 222684, 222772, 222806, 222842,
       222899, 222946, 222995, 223063, 223157, 223241, 223332, 223448,
       223603, 223786, 223952, 224173, 224384, 224592, 224844, 225070,
       225337, 225582, 225814, 226064, 226311, 226591, 226850, 227168,
       227457, 227764, 228106, 228483, 228912, 229252, 229684, 230083,
       230548, 230943, 231435, 231933, 232401, 233413, 233954, 234580,
       235596, 236146, 236893, 237428, 238049, 238567, 239100, 239656,
       240247, 240776, 241327, 241862, 242475, 243021, 243687, 244298,
       244936, 245496, 246059, 246706, 247231, 247757, 248327, 248836,
       249334, 249796, 250313, 250814, 251298, 251767, 252251, 252753,
       253166, 253600, 254050, 254549, 254999, 255465, 255906, 256392,
       256820, 257289, 257743, 258182, 258630, 259107, 259583, 260026,
       260507, 260946, 261417, 261880, 262341, 262831, 263335, 263831,
       264319, 264851, 265396, 265923, 266468, 267051, 267644, 268183,
       268716, 269255, 269777, 270366, 270906, 271471, 272024, 272550,
       273115, 273668, 274200, 274750, 275288, 275892, 276417, 276912,
       277422, 277944, 278487, 278991, 279478, 279993, 280485, 281036,
       281526, 281992, 282515, 283048, 283563, 284086, 284646, 285160,
       285702, 286188, 286707, 287248, 287787, 288299, 288803, 289321,
       289865, 290428, 290950, 291463, 292003, 292542, 293096, 293651,
       294165, 294729, 295313, 295897, 296454, 296989, 297526, 298087,
       298618, 299149, 299718, 300260, 300796, 301336, 301907, 302467,
       303022, 303571, 304096, 304675, 305195, 305763, 306314, 306841,
       307402, 307954, 308481, 309055, 309586, 310122, 310666, 311196,
       311680, 312185, 312657, 313171, 313666, 314152, 314601, 314992,
...
       322793, 322983, 323202, 323396, 323590, 323805, 323994, 324205,
       324405, 324610, 324849, 325116, 325372, 325592, 325906, 326171,
       326452, 326776, 327120, 327356, 327613, 327928, 328196, 328500,
       328759, 329019, 329451, 329918, 330507, 330881, 331638, 331776],
      dtype=int64)
```

Figure 19: Cumulative Sum Of Histogram

(a) Equalized Foreground Image

(b) Final Image

7 Question seven

```python
# Sobel Vertical Kernel
sobel_vertical_kernel = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]], dtype=np.float32)
vertical_gradient = cv.filter2D(input_image, -1, sobel_vertical_kernel)

# Sobel Horizontal Kernel
sobel_horizontal_kernel = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]], dtype=np.float32)
horizontal_gradient = cv.filter2D(input_image, -1, sobel_horizontal_kernel)

# Gradient Magnitude Kernel
gradient_magnitude = np.sqrt(vertical_gradient**2 + horizontal_gradient**2)
```

Figure 21: Cumulative Sum Of Histogram

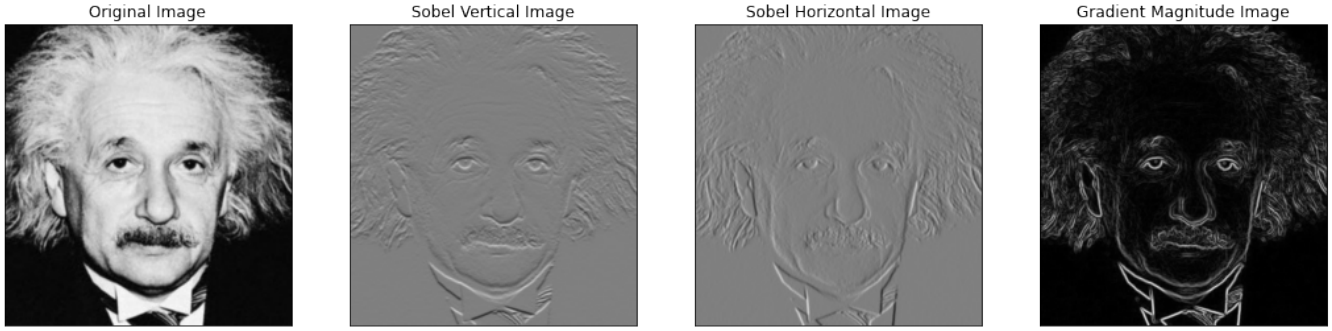


Figure 22: Cumulative Sum Of Histogram

```python
# Sobel Vertical Kernel
sobel_vertical_kernel = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]], dtype=np.float32)
sobel_x = cv.filter2D(input_image, -1, sobel_vertical_kernel)

# Sobel Horizontal Kernel
sobel_horizontal_kernel = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]], dtype=np.float32)
sobel_y = cv.filter2D(input_image, -1, sobel_horizontal_kernel)

img_gradient_x = np.zeros(input_image.shape)
img_gradient_y = np.zeros(input_image.shape)
rows, columns = input_image.shape

# Carry out padding
padding_value = 0
padded_image = np.full((rows + 2, columns + 2), padding_value, dtype=np.uint8)

# Copy input_image into the center of the padded image
padded_image[1:rows + 1, 1:columns + 1] = input_image

for i in range(rows):
    for j in range(columns):
        img_gradient_x[i, j] = np.sum(np.multiply(sobel_horizontal_kernel, padded_image[i:i + 3, j:j + 3]))

for i in range(rows):
    for j in range(columns):
        img_gradient_y[i, j] = np.sum(np.multiply(sobel_vertical_kernel, padded_image[i:i + 3, j:j + 3]))

# Calculate the Gradient Magnitude
img_gradient_magnitude = np.sqrt(img_gradient_x**2 + img_gradient_y**2)
```

Figure 23: Cumulative Sum Of Histogram

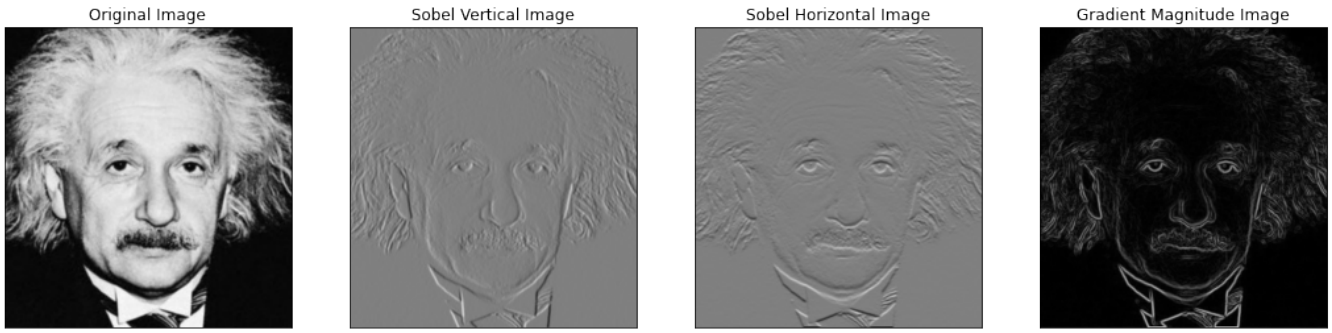


Figure 24: Cumulative Sum Of Histogram

```python
# Define the Sobel Vertical Kernel
sobel_ver_kernel1 = np.array([[1], [2], [1]])
sobel_ver_kernel2 = np.array([[1, 0, -1]])

# Define the Sobel Horizontal Kernel
sobel_hor_kernel1 = np.array([[1], [0], [-1]])
sobel_hor_kernel2 = np.array([[1, 2, 1]])

# Initialize intermediate images
img_xp_1 = np.zeros(input_image.shape)
img_xp = np.zeros(input_image.shape)
img_yp_1 = np.zeros(input_image.shape)
img_yp = np.zeros(input_image.shape)

# Perform convolution using scipy.signal
img_xp_1 = sig.convolve2d(input_image, sobel_ver_kernel1, mode="same")
img_xp = sig.convolve2d(img_xp_1, sobel_ver_kernel2, mode="same")
img_yp_1 = sig.convolve2d(input_image, sobel_hor_kernel1, mode="same")
img_yp = sig.convolve2d(img_yp_1, sobel_hor_kernel2, mode="same")

# Calculate the Gradient Magnitude
img_grad_p = np.sqrt(img_xp**2 + img_yp**2)
```

Figure 25: Cumulative Sum Of Histogram

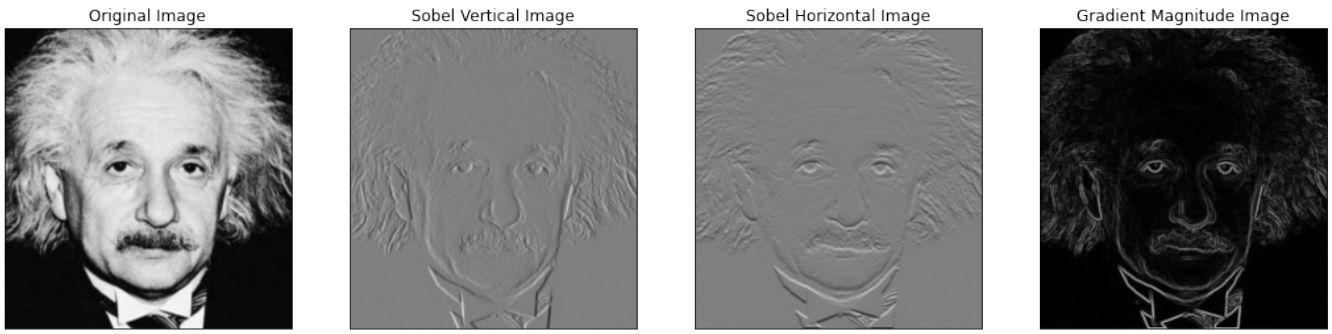


Figure 26: Cumulative Sum Of Histogram

8 Question eight

(a) Image 1: Nearest Neighbor

(b) Image 1: Bilinear Interpolation

(a) Image 2: Nearest Neighbor

(b) Image 2: Bilinear Interpolation

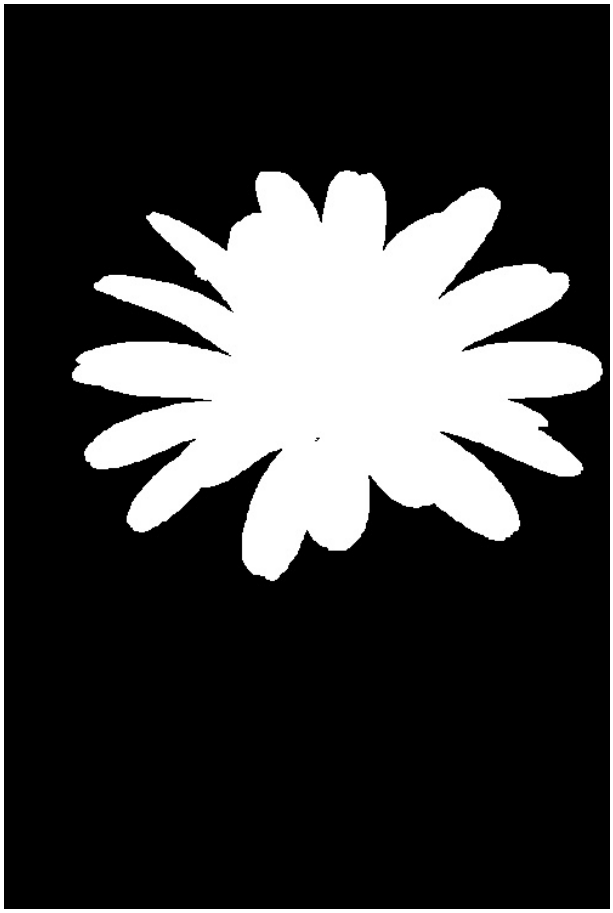
(a) Image 6: Nearest Neighbor

(b) Image 6: Bilinear Interpolation

Table 1: SSD values

Image	SSD in Nearest Neighbor	SSD in Bilinear Interpolation
1	0.0004811121335890121	0.0004775562417232688
2	0.0001830374980420635	0.0001642905306185653 8
6	0.0004698807527731496	0.0004888167863974442

9 Question nine



(a) Segmentation mask

(b) Foreground Image

Figure 31: Background Image

(a) Original Image

(b) Enhanced Image

When using grabCut, there can be pixels near the image's edges that potentially belong to both the foreground and background. As a result, when the two images are combined, these edge pixels are added together giving a value higher than 255. These values are automatically mapped back to zero by the OpenCV library, causing them to appear darker.

github link