***Programming Environment***

*I used jupyterNotebook version 3.12.*

*And*

*I used opencv ,matplotlib and numpy libraries in python.*

***Results***

*Give your code outputs here. Your code should produce at least the following two figures for each image.*

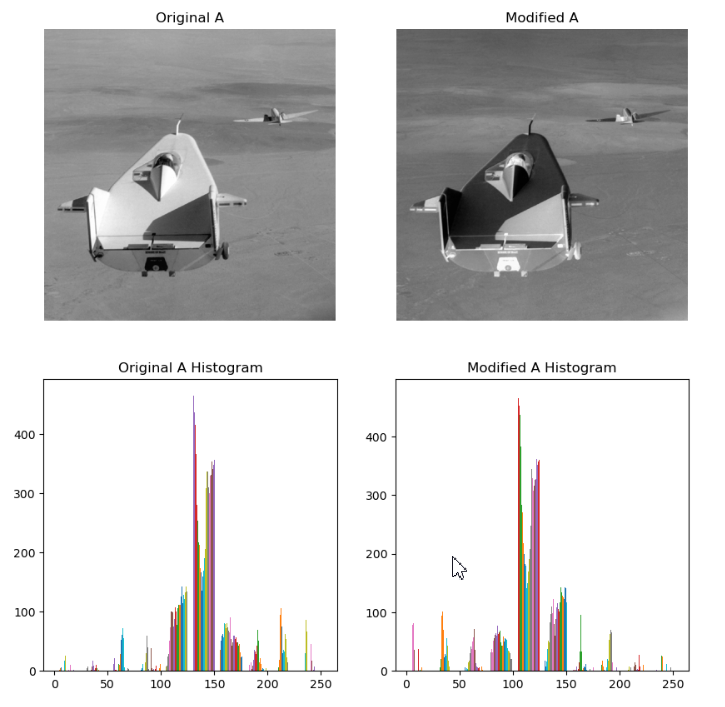


Figure 1: The original A and modified A images histograms.

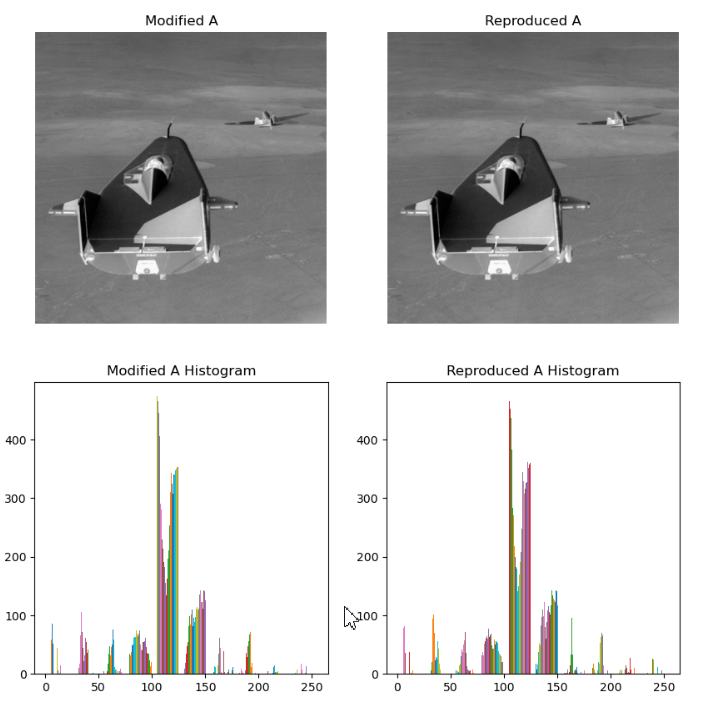


Figure 2: The modified A and reproduced image A histograms.

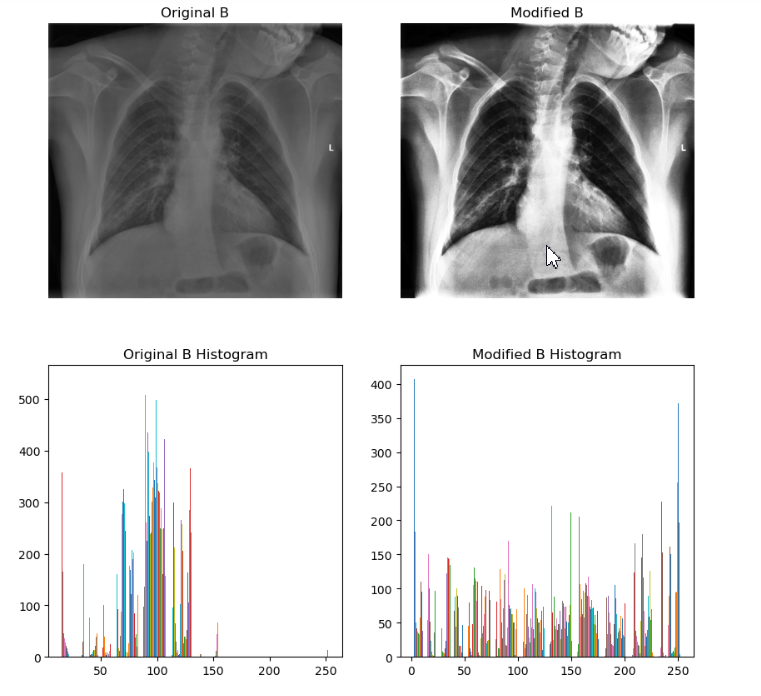


Figure 3: The original B and modified B images histograms.

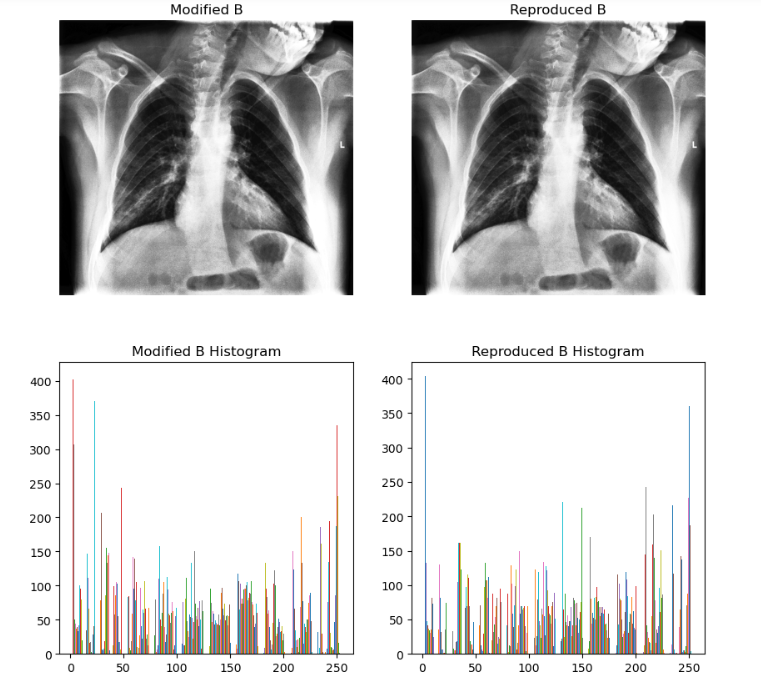


Figure 4: The modified B and reproduced B image histograms.

ekran görüntüsü, mum, araba, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 5: The original C and modified C images histograms.

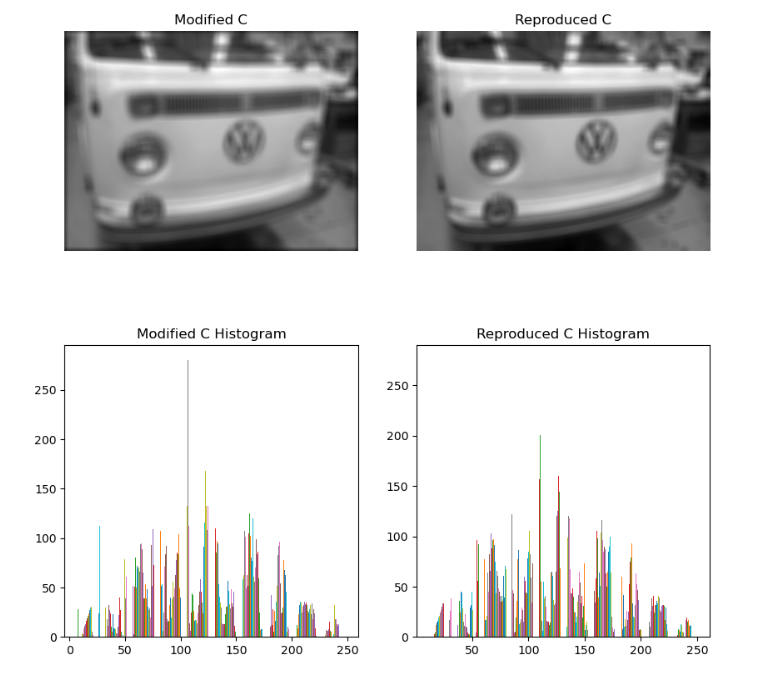


Figure 6: The modified C and reproduced C image histograms.

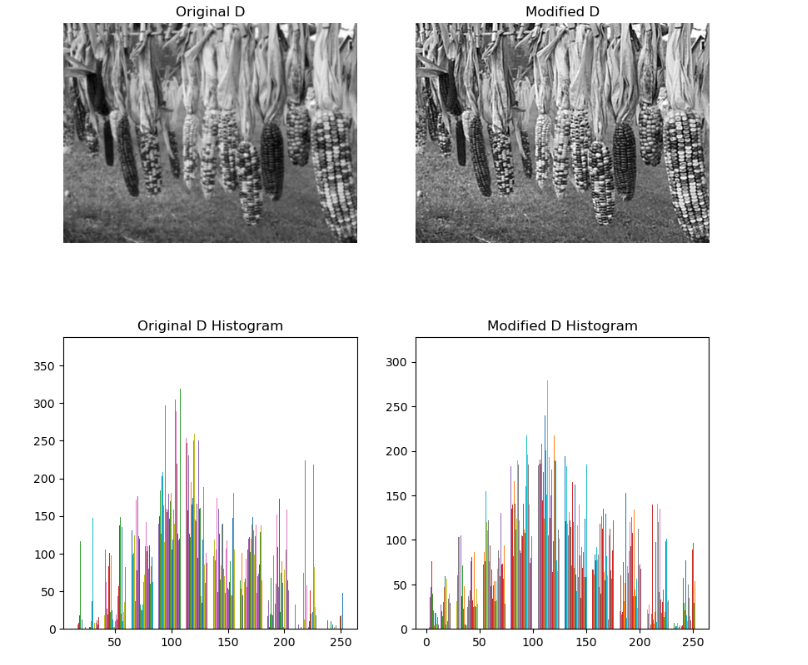


Figure 7: The original D and modified D images histograms.

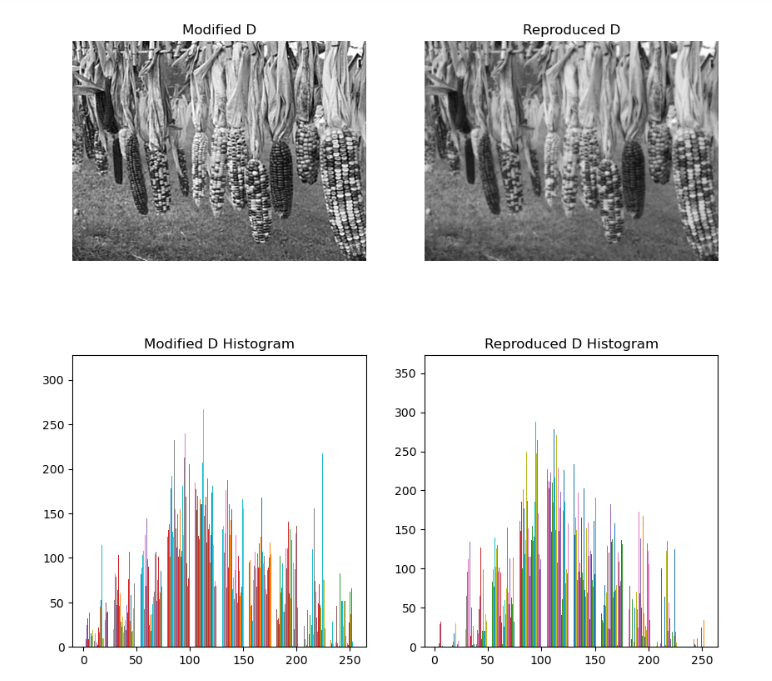


Figure 8: The modified D and reproduced D image histograms.

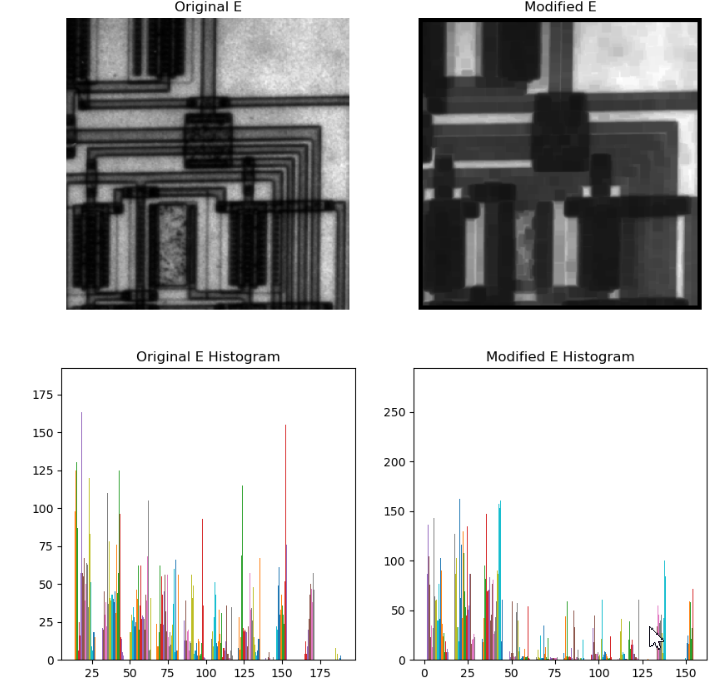


Figure 9: The original Eand modified E images histograms.

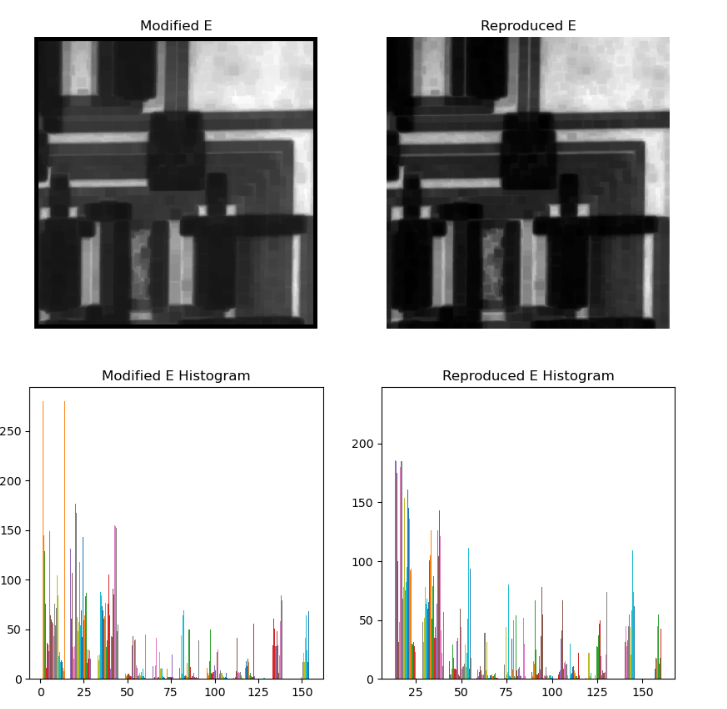


Figure 10: The modified E and reproduced E image histograms.

metin, ekran görüntüsü, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 11: The original F and modified F images histograms.

metin, ekran görüntüsü, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 12: The modified F and reproduced F image histograms.

***Discussion***

*This is the main part of your assignment. Justify your predictions and selected parameters by answering the following questions for each image (A – F).*

* *What spatial domain technique was used to modify the source image and why?*
* *What are the parameters (if any) of the applied method?*
* *Have you obtained exactly the same output or an approximation? (You can compare the images by calculating the sum of the pointwise differences.)*

*Image A*

*- I used the negative images technique because the colors have changed with their opposites between the original image and the modified image.*

*- I got exactly the same output image and histogram.*

*Image B*

*- I used the histogram equalization technique because the histogram graph was much more evenly distributed between the original image histogram and the modified image histogram.*

*- I got exactly the same output image and histogram.*

*Image C*

*- I used the spatial filtering technique because the modified image was a blurred version of the original image.*

*-I used a ones matrix of 13 x 13 units as a parameter.*

*- I got exactly the same output image and histogram.*

*Image D*

*- I used the laplacian filtering technique because the modified image was a sharpened version of the original image.*

*- I got the very similar output image and histogram.*

*Image E*

*- Since the changed image is a dark version of the original image, I used the min filtering technique.*

*- I used the 7x7 matrix as a parameter.*

*- I got very similar output image and histogram.*

*Image F*

*- Since the changed image is a lighter version of the light pixels of the original image, I used the gamma filtering technique.*

*- I used the value 0.6 as a parameter.*

*- I got exactly the same output image and histogram.*

***Reflections***

*I learned how to apply spatial domain and histogram techniques in Python. I had difficulty approaching the correct values of the parameters. I tried more than one filter on some images, I think I could finally find the right filters.*

***Source Code***

* ***Image A Source Codes***

imageA = cv2.imread('images/A\_original.png',cv2.IMREAD\_GRAYSCALE)

modifiedA =cv2.imread('images/A\_modified.png',cv2.IMREAD\_GRAYSCALE)

reproducedA = 255 – imageA

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageA, cmap='gray')

plt.title('Original A')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageA)

plt.title('Original A Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedA, cmap='gray')

plt.title('Modified A')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedA)

plt.title('Modified A Histogram')

plt.show()

#------------------------------------------------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedA, cmap='gray')

plt.title('Modified A')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedA)

plt.title('Modified A Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedA, cmap='gray')

plt.title('Reproduced A')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedA)

plt.title('Reproduced A Histogram')

plt.show()

* ***Image B Source Codes***

imageB = cv2.imread('images/B\_original.png', cv2.IMREAD\_GRAYSCALE)

modifiedB = cv2.imread('images/B\_modified.png', cv2.IMREAD\_GRAYSCALE)

# Histogram Equalization

reproducedB = cv2.equalizeHist(imageB)

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageB, cmap='gray')

plt.title('Original B')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageB)

plt.title('Original B Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedB, cmap='gray')

plt.title('Modified B')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedB)

plt.title('Modified B Histogram')

plt.show()

#------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedB, cmap='gray')

plt.title('Modified B')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedB)

plt.title('Modified B Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedB, cmap='gray')

plt.title('Reproduced B')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedB)

plt.title('Reproduced B Histogram')

plt.show()

* ***Image C Source Codes***

imageC = cv2.imread("images/C\_original.png", cv2.IMREAD\_GRAYSCALE)

modifiedC = cv2.imread("images/C\_modified.png",cv2.IMREAD\_GRAYSCALE)

filter\_matrix = np.ones((13, 13), np.float32) / 169

# Spatial filtering işlemi

reproducedC = cv2.filter2D(imageC, -1, filter\_matrix)

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageC, cmap='gray')

plt.title('Original C')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageC)

plt.title('Original C Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedC, cmap='gray')

plt.title('Modified C')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedC)

plt.title('Modified C Histogram')

plt.show()

#-----------------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedC, cmap='gray')

plt.title('Modified C')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedC)

plt.title('Modified C Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedC, cmap='gray')

plt.title('Reproduced C')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedC)

plt.title('Reproduced C Histogram')

plt.show()

* ***Image D Source Codes***

imageD = cv2.imread("images/D\_original.png", cv2.IMREAD\_GRAYSCALE)

modifiedD = cv2.imread("images/D\_modified.png", cv2.IMREAD\_GRAYSCALE)

#laplacian filter

laplacian\_image = cv2.Laplacian(imageD, cv2.CV\_64F)

laplacian\_image = cv2.convertScaleAbs(laplacian\_image)

reproducedD = imageD - laplacian\_image

# Show the shapedned image

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageD, cmap='gray')

plt.title('Original D')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageD)

plt.title('Original D Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedD, cmap='gray')

plt.title('Modified D')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedD)

plt.title('Modified D Histogram')

plt.show()

#-----------------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedD, cmap='gray')

plt.title('Modified D')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedD)

plt.title('Modified D Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedD, cmap='gray')

plt.title('Reproduced D')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedD)

plt.title('Reproduced D Histogram')

plt.show()

* ***Image E Source Codes***

imageE = cv2.imread("images/E\_original.png",cv2.IMREAD\_GRAYSCALE)

modifiedE = cv2.imread("images/E\_modified.png",cv2.IMREAD\_GRAYSCALE)

# filtre boyutu 7x7

kernel\_size = 7

# Minimum filtreyi uygulama

reproducedE = cv2.erode(imageE, np.ones((kernel\_size, kernel\_size), np.uint8))

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageE, cmap='gray')

plt.title('Original E')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageE)

plt.title('Original E Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedE, cmap='gray')

plt.title('Modified E')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedE)

plt.title('Modified E Histogram')

plt.show()

#-----------------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedE, cmap='gray')

plt.title('Modified E')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedE)

plt.title('Modified E Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedE, cmap='gray')

plt.title('Reproduced E')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedE)

plt.title('Reproduced E Histogram')

plt.show()

* ***Image F Source Codes***

imageF = cv2.imread('images/F\_original.png', cv2.IMREAD\_GRAYSCALE)

modifiedF = cv2.imread('images/F\_modified.png', cv2.IMREAD\_GRAYSCALE)

# Gamma değeri

gamma1 = 0.6

# Görüntüyü float32 türüne dönüştür

image\_float = imageF.astype(np.float32)

# Gamma dönüşümü uygula

reproducedF = np.power(image\_float, gamma1)

# 0 ile 255 arasına ölçekle

reproducedF = (reproducedF / np.max(reproducedF)) \* 255

# Ölçeklenmiş görüntüyü uint8 türüne dönüştür

reproducedF = reproducedF.astype(np.uint8)

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(imageF, cmap='gray')

plt.title('Original F')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(imageF)

plt.title('Original F Histogram')

plt.subplot(2, 2, 2)

plt.imshow(modifiedF, cmap='gray')

plt.title('Modified F')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(modifiedF)

plt.title('Modified F Histogram')

plt.show()

#-----------------------------------

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(modifiedF, cmap='gray')

plt.title('Modified F')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.hist(modifiedF)

plt.title('Modified F Histogram')

plt.subplot(2, 2, 2)

plt.imshow(reproducedF, cmap='gray')

plt.title('Reproduced F')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.hist(reproducedF)

plt.title('Reproduced F Histogram')

plt.show()