**EAST WEST UNIVERSITY**

**CSE438**

**Section: 01**

**Lab: 03 Report**

**Topic: Match the reference image using histogram matching**

**Submitted By:**

**Name: Md Sifat Ullah Sheikh**

**ID: 2022-1-60-029**

**Submitted To:**

**K.M. Safin Kamal**

**Lecturer**

**Department of Computer Science & Engineering**

**Date: 14 July 2025**

**Q1.** Adjust the histogram of the following image to match the reference image using histogram matching. Show the histogram of original, reference, and output images.

**CODE:**

import cv2

import matplotlib.pyplot as plt

from skimage.exposure import match\_histograms

src\_path = '/kaggle/input/cse-438-lab-03/CSE\_438\_LAB\_03/Picture3.png'

ref\_path = '/kaggle/input/cse-438-lab-03/CSE\_438\_LAB\_03/Picture4.png'

src\_rgb = cv2.cvtColor(cv2.imread(src\_path), cv2.COLOR\_BGR2RGB)

ref\_rgb = cv2.cvtColor(cv2.imread(ref\_path), cv2.COLOR\_BGR2RGB)

matched\_rgb = match\_histograms(src\_rgb, ref\_rgb, channel\_axis=-1)

def plot\_rgb\_histogram(image, ax, title):

color = ('r', 'g', 'b')

for i, col in enumerate(color):

hist = cv2.calcHist([image], [i], None, [256], [0,256])

ax.plot(hist, color=col)

ax.set\_title(title)

ax.set\_xlim([0, 256])

ax.set\_xlabel('Pixel Value')

ax.set\_ylabel('Frequency')

plt.figure(figsize=(18, 5))

titles = ['Source Image', 'Reference Image', 'Matched Image']

images = [src\_rgb, ref\_rgb, matched\_rgb]

for i in range(3):

plt.subplot(1, 3, i+1)

plt.imshow(images[i])

plt.title(titles[i])

plt.axis('off')

plt.tight\_layout()

plt.show()

fig, axes = plt.subplots(1, 3, figsize=(18, 4))

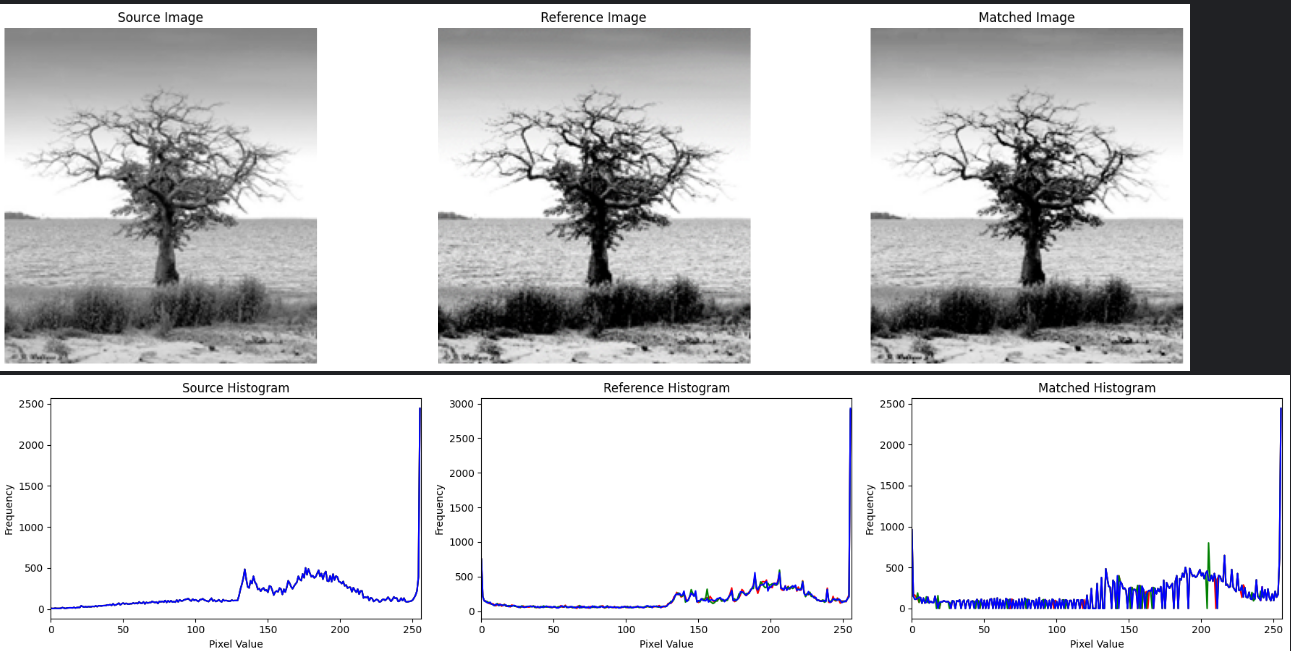
hist\_titles = ['Source Histogram', 'Reference Histogram', 'Matched Histogram']

for i, ax in enumerate(axes):

plot\_rgb\_histogram(images[i], ax, hist\_titles[i])

plt.tight\_layout()

plt.show()



import cv2

import matplotlib.pyplot as plt

from skimage.exposure import match\_histograms

src\_path = '/kaggle/input/cse-438-lab-03/CSE\_438\_LAB\_03/Picture3.png'

ref\_path = '/kaggle/input/cse-438-lab-03/CSE\_438\_LAB\_03/Picture4.png'

src\_rgb = cv2.cvtColor(cv2.imread(src\_path), cv2.COLOR\_BGR2RGB)

ref\_rgb = cv2.cvtColor(cv2.imread(ref\_path), cv2.COLOR\_BGR2RGB)

matched\_rgb = match\_histograms(src\_rgb, ref\_rgb, channel\_axis=-1)

def plot\_rgb\_histogram(image, ax, title):

color = ('r', 'g', 'b')

for i, col in enumerate(color):

hist = cv2.calcHist([image], [i], None, [256], [0,256])

ax.plot(hist, color=col)

ax.set\_title(title)

ax.set\_xlim([0, 256])

ax.set\_xlabel('Pixel Value')

ax.set\_ylabel('Frequency')

plt.figure(figsize=(18, 5))

titles = ['Source Image', 'Reference Image', 'Matched Image']

images = [src\_rgb, ref\_rgb, matched\_rgb]

for i in range(3):

plt.subplot(1, 3, i+1)

plt.imshow(images[i])

plt.title(titles[i])

plt.axis('off')

plt.tight\_layout()

plt.show()

fig, axes = plt.subplots(1, 3, figsize=(18, 4))

hist\_titles = ['Source Histogram', 'Reference Histogram', 'Matched Histogram']

for i, ax in enumerate(axes):

plot\_rgb\_histogram(images[i], ax, hist\_titles[i])

plt.tight\_layout()

plt.show()

A collage of images of snow and mountains

AI-generated content may be incorrect.

**Q2.** Change the contrast of the image using histogram equalization. Show the histogram of both input and output images.

**CODE**

import cv2

import matplotlib.pyplot as plt

img\_path = '/kaggle/input/cse-438-lab-03/CSE\_438\_LAB\_03/Picture5.png'

img\_bgr = cv2.imread(img\_path)

img\_ycrcb = cv2.cvtColor(img\_bgr, cv2.COLOR\_BGR2YCrCb)

y, cr, cb = cv2.split(img\_ycrcb)

y\_eq = cv2.equalizeHist(y)

img\_ycrcb\_eq = cv2.merge([y\_eq, cr, cb])

img\_bgr\_eq = cv2.cvtColor(img\_ycrcb\_eq, cv2.COLOR\_YCrCb2BGR)

img\_rgb = cv2.cvtColor(img\_bgr, cv2.COLOR\_BGR2RGB)

img\_rgb\_eq = cv2.cvtColor(img\_bgr\_eq, cv2.COLOR\_BGR2RGB)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.imshow(img\_rgb)

plt.title('Original Color Image')

plt.axis('off')

plt.subplot(1, 2, 2)

plt.imshow(img\_rgb\_eq)

plt.title('Equalized Color Image')

plt.axis('off')

plt.tight\_layout()

plt.show()

# Save the equalized color image

cv2.imwrite('/kaggle/working/color\_equalized\_image.png', img\_bgr\_eq)

print("✅ Color equalized image saved at: /kaggle/working/color\_equalized\_image.png")

A close-up of a ct scan

AI-generated content may be incorrect.

def plot\_rgb\_hist(image, ax, title):

colors = ('r', 'g', 'b')

for i, col in enumerate(colors):

hist = cv2.calcHist([image], [i], None, [256], [0, 256])

ax.plot(hist, color=col)

ax.set\_title(title)

ax.set\_xlim([0, 256])

ax.set\_xlabel("Pixel Value")

ax.set\_ylabel("Frequency")

fig, axs = plt.subplots(1, 2, figsize=(14, 4))

plot\_rgb\_hist(img\_rgb, axs[0], "Original RGB Histogram")

plot\_rgb\_hist(img\_rgb\_eq, axs[1], "Equalized RGB Histogram")

plt.tight\_layout()

plt.show()

cv2.imwrite('/kaggle/working/color\_equalized\_image.png', img\_bgr\_eq)

print("✅ Color equalized image saved at: /kaggle/working/color\_equalized\_image.png")

A close-up of a graph

AI-generated content may be incorrect.