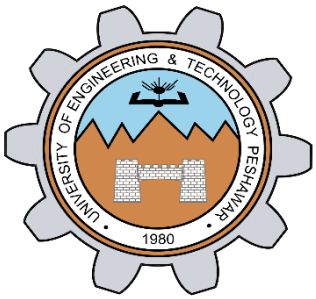
UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR

DEPARTMENT OF COMPUTER SYSTEM ENGINEERING

Digital Image Processing

Assignment no: 2



Spring 2024

Submitted by:

Maaz Habib

Registration No.:

20pwcse1952

Class Section:

C

Submitted to:

Dr. Abeer Irfan

27 MAY 2024

Implementation of histogram equalization and histogram matching (Python/MATLAB)

Histogram Equalization:

Installing library:

```
PS F:\study\Python>
PS F:\study\Python> pip install opencv-python numpy matplotlib
>>
```

Importing libraries:

Loading image:

```
# Load the image
image_path = r'F:\study\Python\image.jpg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
if image is None:
    raise ValueError(f"Error loading image from path: {image_path}")

```

Histogram Equalization, plotting original and equalized image:

```

# Apply histogram equalization
equalized_img = cv2.equalizeHist(image)

# Plotting original and equalized image
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(image, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('Equalized Image')
plt.imshow(equalized_img, cmap='gray')
plt.axis('off')

plt.show()
```

Result:

Histogram Matching:

```python
# Histogram Matching
source_path = r'F:\study\Python\image.jpg'
target_path = r'F:\study\Python\image2.jpg'

source_img = cv2.imread(source_path, cv2.IMREAD_GRAYSCALE)
target_img = cv2.imread(target_path, cv2.IMREAD_GRAYSCALE)

if source_img is None:
    raise ValueError(f"Error loading source image from path: {source_path}")
if target_img is None:
    raise ValueError(f"Error loading target image from path: {target_path}")

# Calculate the histograms
source_hist = cv2.calcHist([source_img], [0], None, [256], [0, 256])
target_hist = cv2.calcHist([target_img], [0], None, [256], [0, 256])

# Compute the cumulative distribution functions (CDFs)
source_cdf = np.cumsum(source_hist)
source_cdf = source_cdf / source_cdf[-1]  # Normalize

target_cdf = np.cumsum(target_hist)
target_cdf = target_cdf / target_cdf[-1]  # Normalize

# Perform histogram matching
inverse_cdf_mapping = np.interp(source_cdf, target_cdf, np.arange(256))

matched_img = np.interp(source_img.flatten(), np.arange(256), inverse_cdf_mapping)
matched_img = matched_img.reshape(source_img.shape).astype(np.uint8)

```

```
57  # Plot the original, target, and matched images
58  plt.figure(figsize=(15, 5))
59
60  plt.subplot(1, 3, 1)
61  plt.title('Source Image')
62  plt.imshow(source_img, cmap='gray')
63  plt.axis('off')
64
65  plt.subplot(1, 3, 2)
66  plt.title('Target Image')
67  plt.imshow(target_img, cmap='gray')
68  plt.axis('off')
69
70  plt.subplot(1, 3, 3)
71  plt.title('Matched Image')
72  plt.imshow(matched_img, cmap='gray')
73  plt.axis('off')
74
75  plt.show()
76
```

Result: