

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

# Upload the images
uploaded = files.upload()

# Use the correct image paths from the uploaded files
source_image_path = list(uploaded.keys())[0] # First uploaded file as source
reference_image_path = list(uploaded.keys())[1] # Second uploaded file as reference
# Load the source and reference images
source_image = cv2.imread(source_image_path, cv2.IMREAD_GRAYSCALE)
reference_image = cv2.imread(reference_image_path, cv2.IMREAD_GRAYSCALE)

if source_image is None or reference_image is None:
    raise FileNotFoundError("One or both images could not be loaded.")

# Function definitions remain unchanged
def calculate_histogram(image):
    """Calculate the histogram of an image."""
    hist, _ = np.histogram(image.flatten(), bins=256, range=[0, 256])
    return hist

def calculate_cdf(histogram):
    """Calculate the cumulative distribution function (CDF) for a given histogram."""
    cdf = histogram.cumsum()
    cdf_normalized = cdf / cdf.max() # Normalize CDF to range from 0 to 1
    return cdf_normalized

def create_mapping_function(cdf_source, cdf_reference):
    """Create a mapping function to match the CDF of the source image to the CDF of the reference image."""
    mapping = np.zeros(256)
    for src_pixel_value in range(256):
        ref_pixel_value = np.argmin(np.abs(cdf_source[src_pixel_value] - cdf_reference))
        mapping[src_pixel_value] = ref_pixel_value
    return mapping

def apply_mapping_function(image, mapping_function):
    """Apply the mapping function to the source image to get the matched image."""
    matched_image = cv2.LUT(image, mapping_function.astype(np.uint8))
    return matched_image
```

```
# Perform the histogram matching process
hist_source = calculate_histogram(source_image)
hist_reference = calculate_histogram(reference_image)
cdf_source = calculate_cdf(hist_source)
cdf_reference = calculate_cdf(hist_reference)

mapping_function = create_mapping_function(cdf_source, cdf_reference)
matched_image = apply_mapping_function(source_image, mapping_function)
# Save and display the resulting image
cv2.imwrite('matched_image.jpg', matched_image)

plt.figure(figsize=(12, 8))
plt.subplot(2, 2, 1)
plt.title('Source Image')
plt.imshow(source_image, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.title('Reference Image')
plt.imshow(reference_image, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.title('Matched Image')
plt.imshow(matched_image, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.title('Mapping Function')
plt.plot(mapping_function)
plt.xlim([0, 255])
plt.ylim([0, 255])
plt.xlabel('Source Pixel Value')
plt.ylabel('Reference Pixel Value')

plt.tight_layout()
plt.show()

print("Histogram-matched image has been saved as 'matched_image.jpg'.")
```