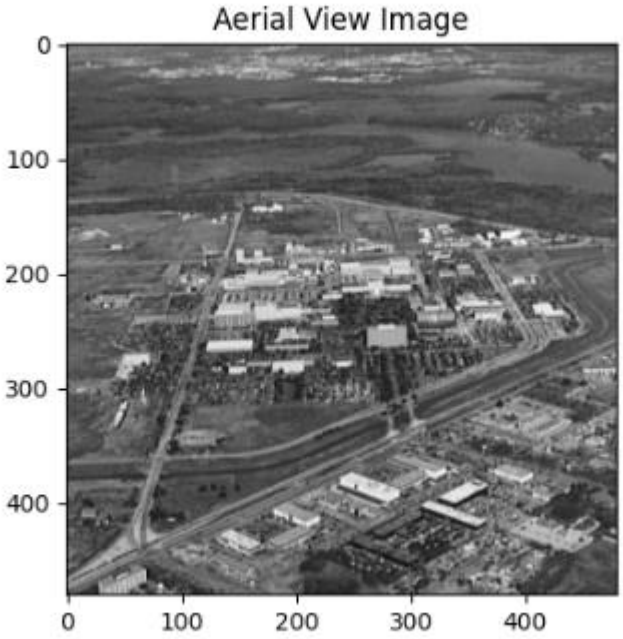
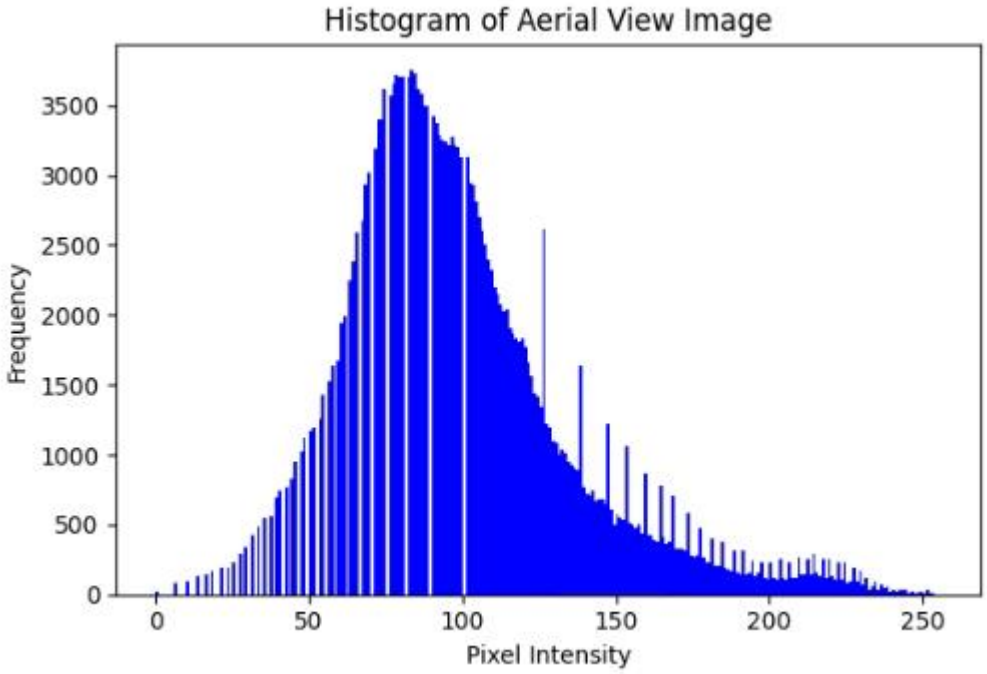
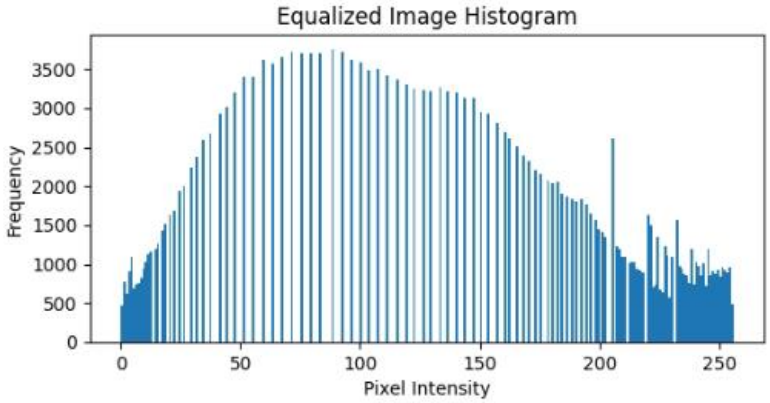
1. Please depict the histogram and graph of the assigned image "aerial_view.tif", and print out the source code?

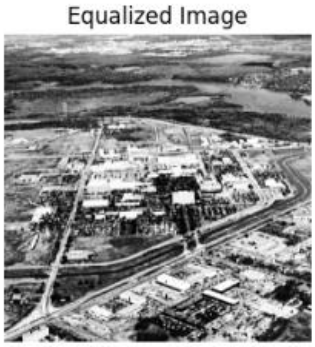




```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Load the input image (grayscale)
image_path = 'aerial_view.tif'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Step 2: Compute the histogram
hist, bins = np.histogram(image.flatten(), 256, [0, 256])

# Step 3: Plot the image and its histogram
plt.figure(figsize=(6, 8))

# Plot the image
plt.subplot(2, 1, 1)
plt.imshow(image, cmap='gray')
plt.title('Aerial View Image')

# Plot the histogram
plt.subplot(2, 1, 2)
plt.hist(image.flatten(), 256, [0, 256], color='b')
plt.title('Histogram of Aerial View Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Display both
plt.tight_layout()
plt.show()
```

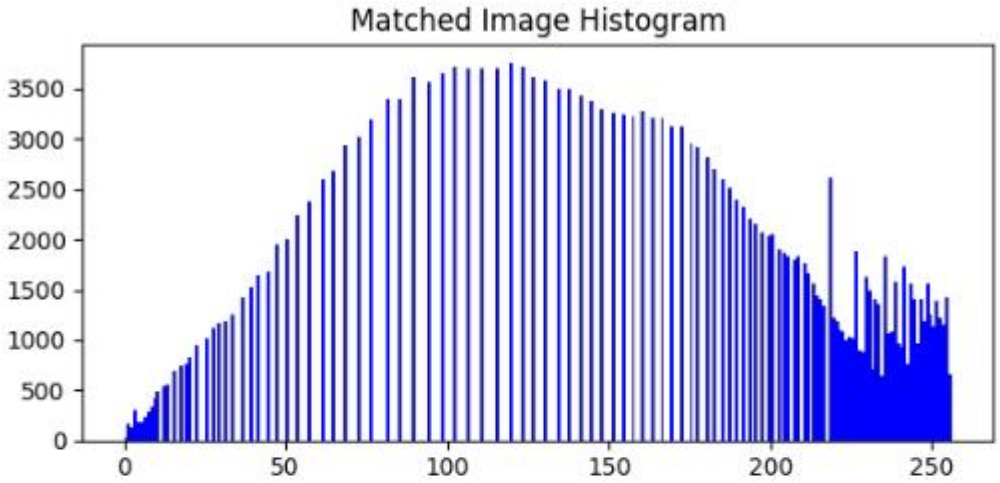
2. Please plot the histogram and graph of the image after Histogram Equalization, and print out the source code?

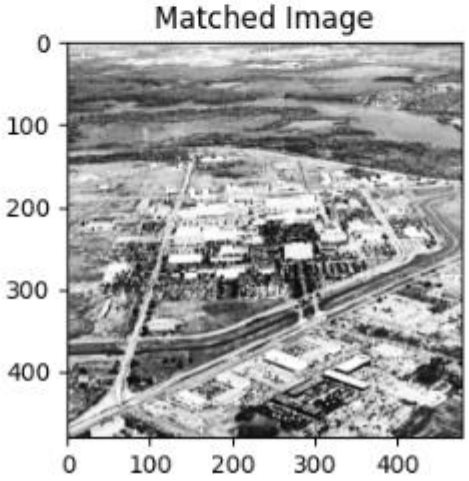





```python
import cv2
import matplotlib.pyplot as plt

# Load the image in grayscale
image = cv2.imread('aerial_view.tif', cv2.IMREAD_GRAYSCALE)

# Apply Histogram Equalization
equalized_image = cv2.equalizeHist(image)

# Plotting the original and equalized histograms
plt.figure(figsize=(12, 6))

# Original Image Histogram
plt.subplot(2, 2, 1)
plt.hist(image.ravel(), bins=256, range=(0, 256))
plt.title('Original Image Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Equalized Image Histogram
plt.subplot(2, 2, 2)
plt.hist(equalized_image.ravel(), bins=256, range=(0, 256))
plt.title('Equalized Image Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Displaying Original Image
plt.subplot(2, 2, 3)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.axis('off')

# Displaying Equalized Image
plt.subplot(2, 2, 4)
plt.imshow(equalized_image, cmap='gray')
plt.title('Equalized Image')
plt.axis('off')

# Show the plots
plt.tight_layout()
plt.show()
```

3.Please plot the histogram and graph of the image after Histogram Matching (specificiation) by $pz(zq) = c \cdot zq$ ^0.4 , and print out the source code? (NOTE: the parameter, c, needs to calculate in advance)





```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Load the input image and convert to grayscale
image_path = 'aerial_view.tif'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Step 2: Calculate the constant c
# We assume the intensity values are in the range [0, 255]
z = np.arange(256)
target_pdf = z ** 0.4
c = 1 / np.sum(target_pdf)
target_pdf_normalized = c * target_pdf

# Step 3: Calculate the cumulative distribution function (CDF) for the target distribution
target_cdf = np.cumsum(target_pdf_normalized)
target_cdf_normalized = target_cdf / target_cdf[-1]  # Normalize to range [0, 1]

# Step 4: Compute the histogram and CDF of the input image
input_hist, bins = np.histogram(image.flatten(), 256, [0, 256])
input_cdf = input_hist.cumsum()
input_cdf_normalized = input_cdf / input_cdf[-1]

# Step 5: Perform histogram matching by finding the best mapping
mapping = np.zeros(256, dtype=np.uint8)
for i in range(256):
    diff = np.abs(target_cdf_normalized - input_cdf_normalized[i])
    mapping[i] = np.argmin(diff)

# Step 6: Apply the mapping to the original image to get the matched image
matched_image = mapping[image]

# Step 7: Plot the original and matched histograms
plt.figure(figsize=(12, 6))

# Original histogram
plt.subplot(2, 2, 1)
plt.hist(image.flatten(), 256, [0, 256], color='r')
plt.title('Original Image Histogram')

# Matched histogram (now blue instead of green)
plt.subplot(2, 2, 2)
plt.hist(matched_image.flatten(), 256, [0, 256], color='b')  # Changed to blue
plt.title('Matched Image Histogram')
```

```python
# Step 8: Display the original and matched images
plt.subplot(2, 2, 3)
plt.imshow(image, cmap='gray')
plt.title('Original Image')

plt.subplot(2, 2, 4)
plt.imshow(matched_image, cmap='gray')
plt.title('Matched Image')

plt.tight_layout()
plt.show()
```

4. Please comment the original, the histogram-equalized and the histogram-matching images ?

1. Original Image

- Description:
    - The original image is aerial_view.tif, an aerial photograph.
    - Brightness and contrast may be insufficient, resulting in unclear details.
    - Pixel distribution may be concentrated in certain brightness areas, affecting visibility.
- Analysis:
    - The pixel value distribution may be uneven, with some areas too bright or too dark.
    - A skewed histogram indicates low contrast.
- Suggestions:
    - Consider using histogram equalization or matching techniques to enhance details and contrast.

2. Histogram Equalized Image

- Description:
    - Enhances the contrast of the image, making the brightness distribution more uniform and increasing details.
- Analysis:
    - Brightness and contrast significantly improve, especially in dark and bright areas.
    - The histogram shows a uniform distribution of pixel values, effectively utilizing the brightness range.
- Suggestions:
    - Histogram equalization effectively enhances contrast but may lead

to increased noise.

3. Histogram Matched Image

- Description:
    - Adjusts the histogram of the image to match that of another image, enhancing color and brightness consistency.
    - Matching is done using the specified parameter pz(zq)=c · zq0.4p_z(z_q) = c \cdot z_q^{0.4}pz(zq)=c · zq0.4.
- Analysis:
    - Brightness and contrast achieve the desired effect, with some areas appearing more prominent or softer.
    - The histogram after matching shows the pixel value distribution tending toward the target histogram.
- Suggestions:
    - Histogram matching is particularly effective in processing multiple images, but excessive matching may lead to loss of detail or unnatural colors.

Summary

- Original Image: Lacks contrast and detail, requires processing.
- Histogram Equalized Image: Contrast and detail significantly improved, with a uniform pixel value distribution.
- Histogram Matched Image: Achieves the target histogram, enhancing the consistency and visibility of the image.