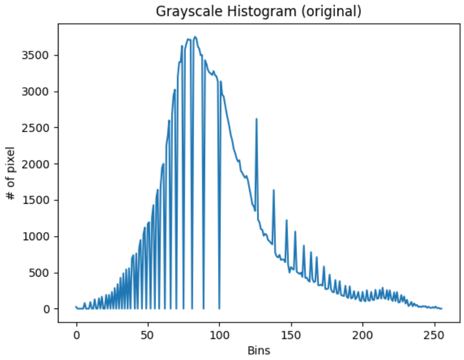
DIP HW 3-1

人工智慧碩二 311505011 黃丰楷

1. Please depict the histogram and graph of the assigned image “aerial_view.tif”, and print out the source code? (10)

- Original image

- Original histogram



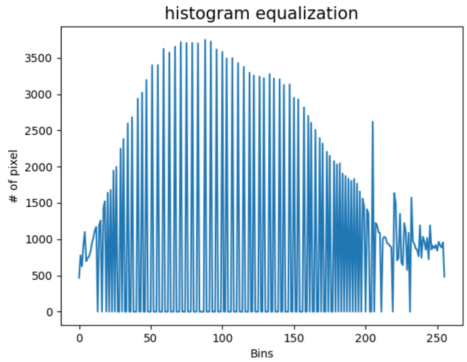
- Source code

```
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("./aerial_view.tif",cv2.IMREAD_GRAYSCALE)
print(img.shape)

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


fig = plt.figure()
plt.plot(hist)
plt.ylabel('# of pixel')
plt.xlabel('Bins')
plt.title("Grayscale Histogram (original)")
plt.savefig("3_1_original_hist.png")

cv2.imwrite("3_1_original.png",img)
```

2. Please plot the histogram and graph of the image after Histogram Equalization, and print out the source code? (30)

- Histogram Equalization image

- Histogram Equalization histogram



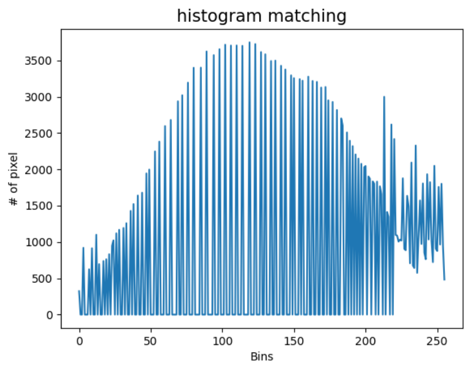
- Source code

```
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("aerial_view.tif", cv2.IMREAD_GRAYSCALE)


eq_img = cv2.equalizeHist(img)

hist = cv2.calcHist([eq_img], [0], None, [256], [0, 256])
equalized_hist = cv2.calcHist([eq_img], [0], None, [256], [0, 256])


cv2.imwrite("3_1_hist_eq.png", eq_img)

fig = plt.figure()
plt.plot(hist)
plt.ylabel('# of pixel')
plt.xlabel('Bins')
plt.title("histogram equalization", fontsize=15)

plt.savefig("3_1_hist_eq_hist.png")
```

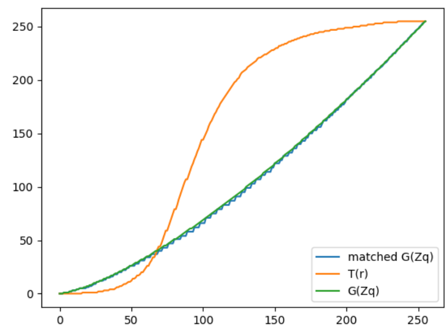
3. Please plot the histogram and graph of the image after Histogram Matching (specificiation) by $p_z(z_q) = c \cdot z_q^{0.4}$, and print out the source code? (NOTE: the parameter, *c*, needs to calculate in advance) (40)

- Histogram Matching image

- Histogram Matching histogram



- CDF



- Source code

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

```
eq_img = cv2.imread("aerial_view.tif", cv2.IMREAD_GRAYSCALE)


hist_eq = cv2.calcHist([eq_img], [0], None, [256], [0, 256])

pdf = np.squeeze(hist_eq/eq_img.size)

c = 1/np.cumsum(np.arange(256)**0.4)[-1]
print("c:", c)  # c =  0.0005968222208221872
print("sum of pdf: ", np.sum(c*np.arange(256)**0.4))

G_source = np.round(np.cumsum(pdf)*255)

G_reference = np.round(np.cumsum(c*np.arange(256)**0.4)*255) # c =  0.0005968222208221872



match_img = np.empty_like(eq_img)
eq_pixels = list(set(eq_img.flatten()))

for i, sk in enumerate(G_source):
    for j, Gzq in enumerate(G_reference):
        if Gzq == sk:
            match_img[np.where(eq_img == i)] = j
            break
        elif (Gzq<sk) and (G_reference[j+1]>sk):
            value = np.array([j,j+1])
            dis = np.abs(sk - np.array([Gzq, G_reference[j+1]]))
            match_img[np.where(eq_img == i)] = value[np.argmin(dis)]
            break

hist_match = cv2.calcHist([match_img], [0], None, [256], [0, 256])
pdf_match = hist_match/match_img.size
cdf_match = np.uint8(np.cumsum(pdf_match)*255)

fig =  plt.figure()

plt.plot(hist_match)
plt.ylabel('# of pixel')
plt.xlabel('Bins')
plt.title("histogram matching", fontsize=15)
cv2.imwrite("3_1_hist_match.png",match_img)


plt.savefig("3_1_hist_match_hist.png")
cv2.imwrite("3_1_hist_match.png",match_img)


fig =  plt.figure()

line1, = plt.plot(cdf_match, label = "matched G(Zq)" )
line2, =plt.plot(G_source, label = "T(r)")
line3, =plt.plot(G_reference, label = "G(Zq)")

plt.legend(handles = [line1, line2, line3], loc='lower right')
plt.savefig("3_1_hist_match_cdf.png")
```

4. Please comment the original, the histogram-equalized and the
histogram-matching images ? (20)

1. **Original Image**:

   The original image represents the unaltered, raw data as captured by a camera. It retains the characteristics of the scene or object at the time of capture, including its inherent contrast, brightness, and color information. Depending on factors such as lighting conditions, camera settings, and the nature of the subject, the original image may exhibit a wide range of visual qualities.

2. **Histogram-Equalized Image**:

   Histogram equalization is a technique used to enhance the contrast of an image by redistributing pixel intensity values. After equalization, the image's histogram becomes more uniformly spread across the entire range of possible intensities. This results in a visually more appealing image with improved detail and clarity. Areas that were initially underrepresented in terms of intensity levels are now better represented, which leads to a more balanced and vivid representation of the underlying scene.

3. **Histogram-Matched Image**:

   Histogram matching, also known as histogram specification, is a method where the histogram of an image is adjusted to match a specified target histogram. This is particularly useful when you want an image to have a specific distribution of pixel values. The result is an image that not only has enhanced contrast but also closely follows the specified intensity distribution. This can be especially helpful in scenarios where you want to standardize the appearance of images or make them consistent with a particular reference.

In summary, the original image serves as the starting point, capturing the raw data. Histogram equalization enhances contrast by redistributing pixel intensity values. Histogram matching goes a step further by not only enhancing contrast but also aligning the image's histogram with a predefined target distribution. These techniques collectively serve to improve the visual quality and interpretability of digital images.