Lab2-2 : Image Enhancement with Statistical Operation

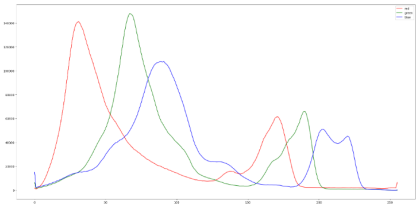
Histogram Equalization

```
[34]:
### START CODE HERE ###
def plot_histogram(image):
    colors = ['red', 'green', 'blue']
    hists = []
    for i,color in enumerate(colors):
        # print(f"i: {i}, color: {color}")
        hists.append(cv2.calcHist([image],[i],None,[256],[0,256]))
    return hists
### END CODE HERE ###
```

- Return ออกมาเป็น List ของ Histogram ที่มีขนาดเป็น 3 โดยจะเป็น Histogram ของสีแดง สีเขียว สีฟ้า (RGB) ตามลำดับ
- ใช้ฟังชั่น cv2.calcHist() โดยจะรับ parameter เป็น image และ index ของสีที่สนใจ
- นอกจากนี้ยังรับ mask ที่เราไม่ได้ต้องการใช้ และก็เป็น size และ range ของ dimension สีของรูปภาพ

Before:

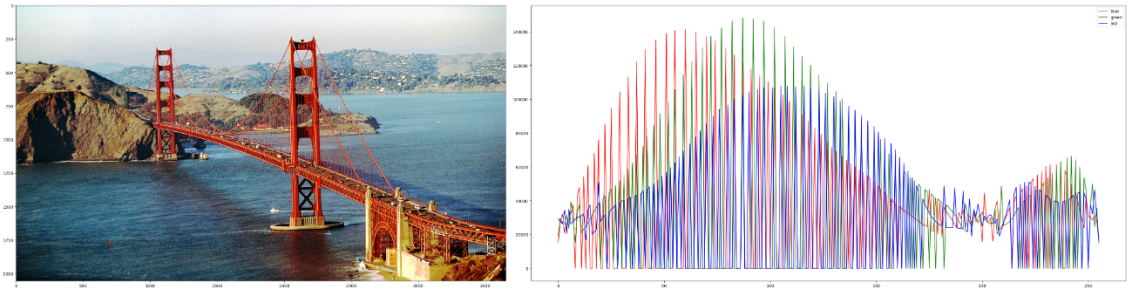
```
[36]:
### START CODE HERE ###

figg = plt.figure(figsize=(40, 10))
axs = figg.subplots(1, 2)
# fig.figsize = [20, 20]
# print(fig.available())
figg.tight_layout()
axs[0].imshow(img)
hists = plot_histogram(img)
colors = ['red', 'green', 'blue']
i = 0
for hist in hists:
    plt.plot(hist, color=colors[i])
    i += 1
plt.legend(colors)
plt.show()


### END CODE HERE ###
```



- นำ list ของ Histogram data จากฟังชั่นข้างต้นมาแสดงทีละอัน โดยใช้สีที่แตกต่างกัน โดยใช้ฟังชั่น plt.ploy(hist, color=colors[i]) โดย hist คือ iterator ของ list และ colors = [‘red’, ‘green’, ‘blue’]

After:

[38]:
```
### START CODE HERE ###

eq_img = []
eq_img.append(cv2.equalizeHist(img[:, :, 0]))
eq_img.append(cv2.equalizeHist(img[:, :, 1]))
eq_img.append(cv2.equalizeHist(img[:, :, 2]))

combile_img = np.array([eq_img[0], eq_img[1], eq_img[2]])
combile_img = np.transpose(combile_img, (1, 2, 0))

figg = plt.figure(figsize=(40, 10))


axs = figg.subplots(1, 2)
axs[0].imshow(combile_img)

hists = plot_histogram(combile_img)
i = 0
for hist in hists:
    axs[1].plot(hist, color=colors[i])
    plt.tight_layout()
    i += 1

axs[1].legend(["red", "green", "blue"])
plt.show()


### END CODE HERE ###
```



- นำรูปภาพเดิม(ซึ่งเป็น array 3 มิติ) มาแยกเป็น array 2 มิติจำนวน 3 อัน เป็นของ channel R, G และ B ตามลำดับ จากนั้นนำแต่ละ Channel มาเข้าเป็นพารามิเตอร์ของฟังก์ชัน cv2.equalizeHist() เพื่อกระจายเฉดสีแต่ละสี

Image Histogram Matching

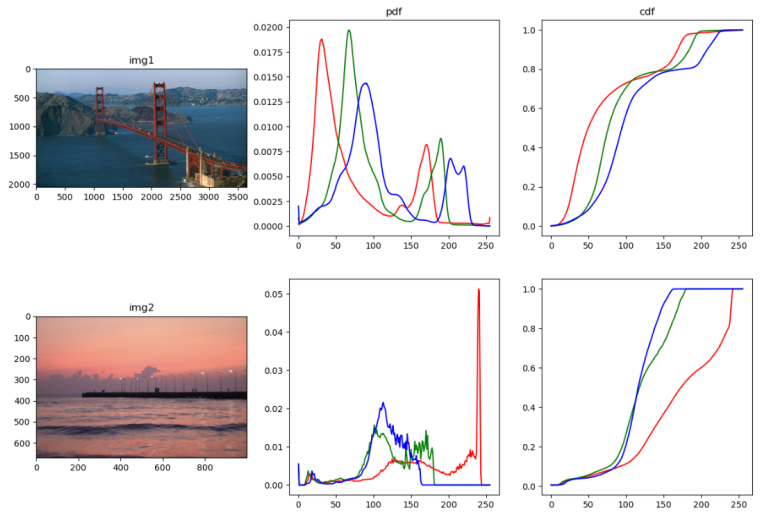
Load image:

```
[48]: ### START CODE HERE ###


img1 = cv2.imread("goldengate.jpg")
img2 = cv2.imread("image.png")


img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

### END CODE HERE ###
```

- โหลดรูปภาพโดยใช้ฟังชั่น cv2.imread() แล้วนำมา convert จากที่ได้เป็น BGR ให้เป็น RGB ด้วยฟังก์ชัน cv2.cvtColor()

Display Histogram matching:

```
[69]: ### START CODE HERE ###

fcol = "white"
bcol = "black"
img_l = [img1, img2]
colors = ["red", "green", "blue"]

figg = plt.figure(figsize=(15, 10))

axs = figg.subplots(2, 3)

i = 0
axs[i][1].set_title("pdf")
axs[i][2].set_title("cdf")
for img in img_l:
    # axs[i][1].spines.color=fcol
    axs[i][0].set_title(f"img{i+1}")
    axs[i][0].imshow(img,)
    hists = plot_histogram(img)
    j = 0
    for hist in hists:
        hist = hist/hist.sum()
        axs[i][1].plot(hist, color=colors[j])
        cum_hist = hist.cumsum()
        # cum_hist = cum_hist/cum_hist.sum()
        axs[i][2].plot(cum_hist, color=colors[j])
        j += 1
    i += 1

### END CODE HERE ###
```

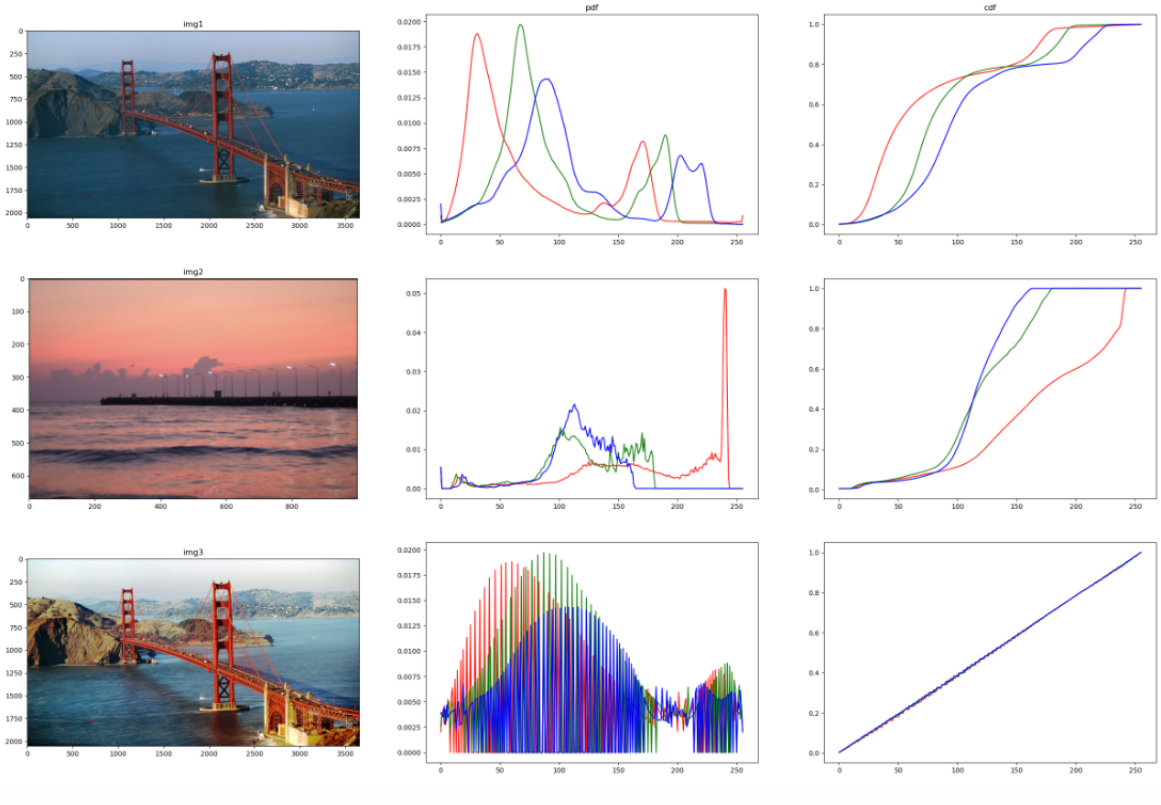


- ในการ display PDF เราสามารถ plot ได้โดยตรงจาก histrogram data ข้อก่อนหน้าได้เลย แต่ว่าในแนวแกน y เราต้องเป็นความน่าจะเป็น เราจึงต้องหารผลรวมของ histogram นั้นด้วย

- ส่วนในการ display CDF เราจำเป็นต้องนำมาเข้าฟังชั่น cumsum() เพื่อซึ่งฟังก์ชันนี้จะ return ออกมาเป็น list ใหม่ซึ่งคำนวณออกมาเป็น cumulative แล้ว ยกตัวอย่างเช่นหาก input เป็น [1,2,3,4] จะได้ output เป็น [1, 3, 6, 10]

Compare with Equalized Image

```
[80]: ### START CODE HERE ###

img_l = [img1, img2, combile_img]

figg = plt.figure(figsize=(30, 20))

axs = figg.subplots(3, 3)
i = 0
axs[i][1].set_title("pdf")
axs[i][2].set_title("cdf")
for img in img_l:
    axs[i][0].set_title(f"img{i+1}")
    axs[i][0].imshow(img)
    hists = plot_histogram(img)
    j = 0
    for hist in hists:
        hist_norm = hist/hist.sum()
        # hist_norm = hist/hist.sum()
        axs[i][1].plot(hist_norm, color=colors[j])
        cum_hist = hist_norm.cumsum()
        axs[i][2].plot(cum_hist, color=colors[j])
        j += 1
    i += 1


### END CODE HERE ###
```



- จากข้อนี้ก็ใช้หลังการเดียวกันกับข้อก่อนหน้า แต่เราต้องแก้การพล็อตของเราในฟังก์ชัน subplot ให้เป็น 3 แถวด้วย แล้วก็ทำแบบเดียวกันกับข้อก่อนหน้าได้เลย