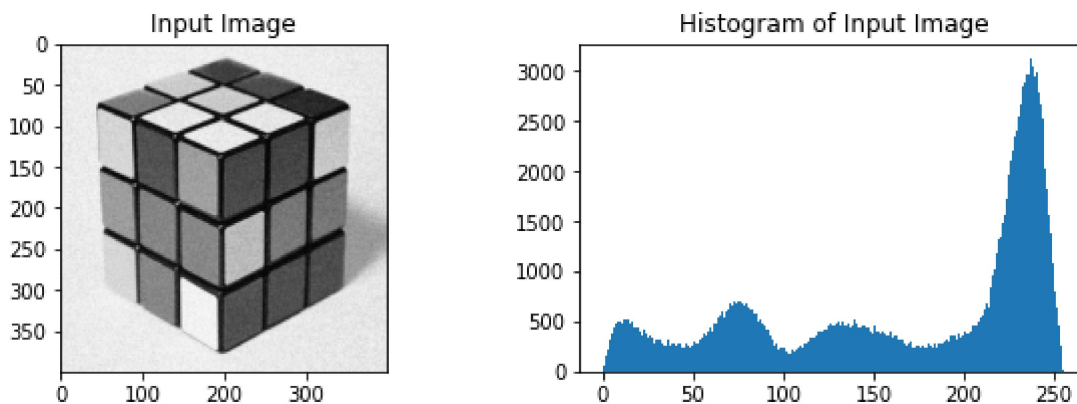


importing required libraries

```
In [ ]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Input Image

```
In [ ]: path = "../img/rubiks_cube.png"
inp_img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
inp_img = cv2.resize(inp_img, (400,400))
figure, axis = plt.subplots(1,2, figsize=(10, 3))
axis[0].set_title("Input Image")
axis[0].imshow(inp_img, "gray")
axis[1].set_title("Histogram of Input Image")
axis[1].hist(inp_img.ravel(), 256,[0,256])
plt.show()
```



Creating Required Helper Function

probability distribution function of the intensity

```
In [ ]: def _get_pdf(img):
    pdf = [0.0 for _ in range(256)]
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            x = img.item(i,j)
            pdf[x]+=1
    pdf=np.array(pdf)/(img.shape[0]*img.shape[1])
    return pdf
```

cumulative distribution function of the intensity

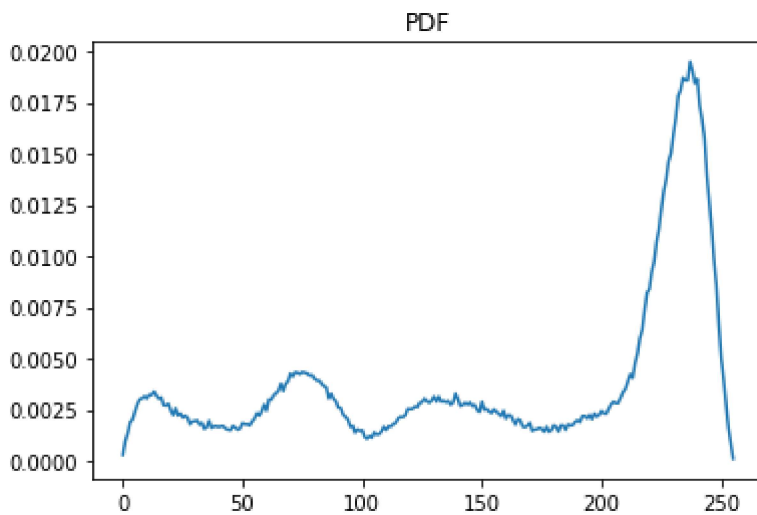
```
In [ ]: def _get_cdf_s(pdf):
    cdf = [0.0 for _ in range(256)]
```

```
s = [0.0 for _ in range(256)]
cumulative_sum = 0
for i in range(256):
    cumulative_sum+=pdf[i]
    cdf[i]=cumulative_sum
    s[i]=round(cumulative_sum*255)
return cdf,s
```

Histogram Equalization

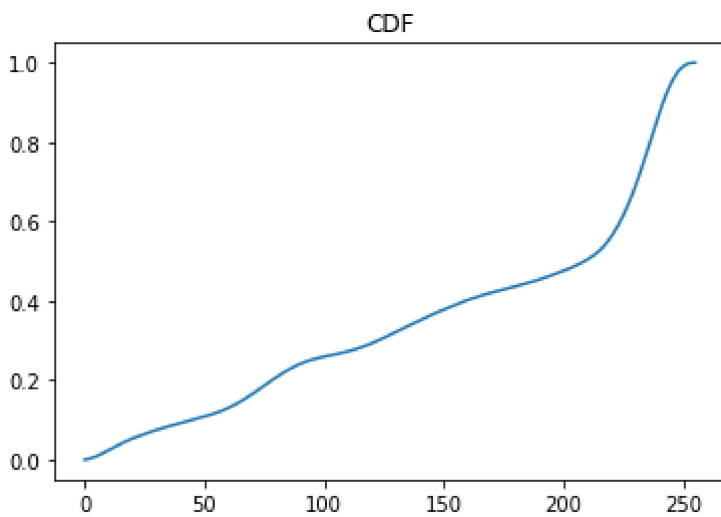
PDF of the input Image

```
In [ ]: pdf_input = _get_pdf(inp_img)
plt.plot(pdf_input,label="input")
plt.title("PDF")
plt.show()
```



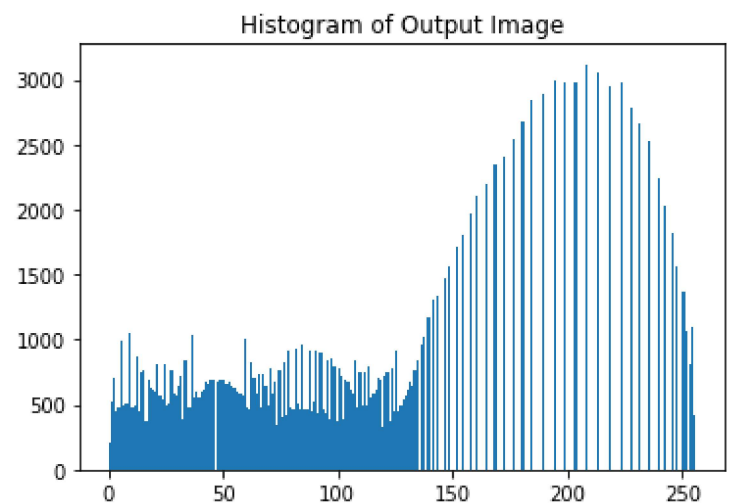
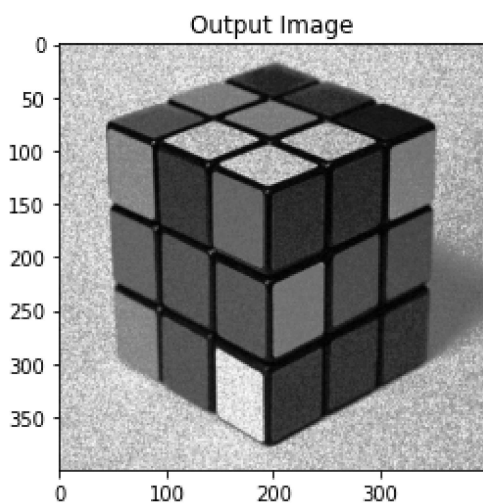
CDF of the input image

```
In [ ]: cdf_input,s = _get_cdf_s(pdf_input)
plt.plot(cdf_input, label="input")
plt.title("CDF")
plt.show()
```



Generating Output

```
In [ ]: output = np.zeros((inp_img.shape), dtype="uint8")
        for i in range(output.shape[0]):
            for j in range(output.shape[1]):
                x = inp_img.item(i,j)
                output[i][j]=s[x] #equalization
        pdf_output = _get_pdf(output)
        cdf_output, s1 = _get_cdf_s(pdf_output)
        figure, axis = plt.subplots(1,2, figsize=(13,4))
        axis[0].imshow(output, "gray")
        axis[0].set_title("Output Image")
        axis[1].hist(output.ravel(),256,[0,256])
        axis[1].set_title("Histogram of Output Image")
        plt.show()
```



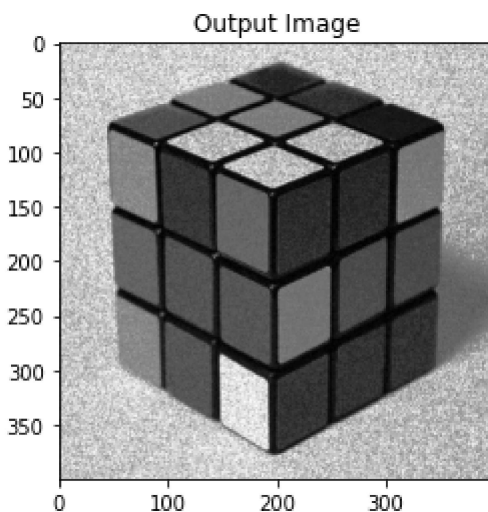
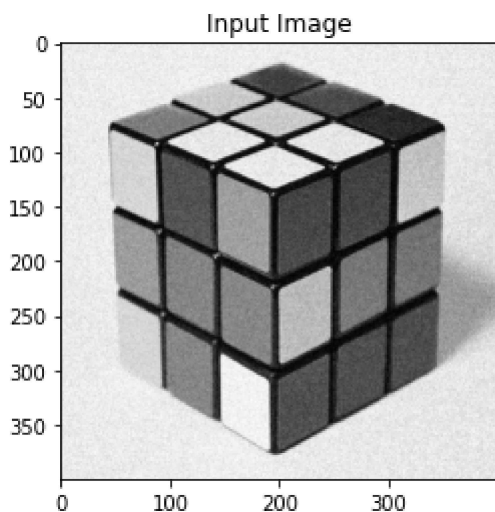
Comparision between Input and Output

Images

```
In [ ]: figure , axis = plt.subplots(1,2, figsize=(10,4))
```

```
axis[0].set_title("Input Image")
axis[0].imshow(inp_img, "gray")

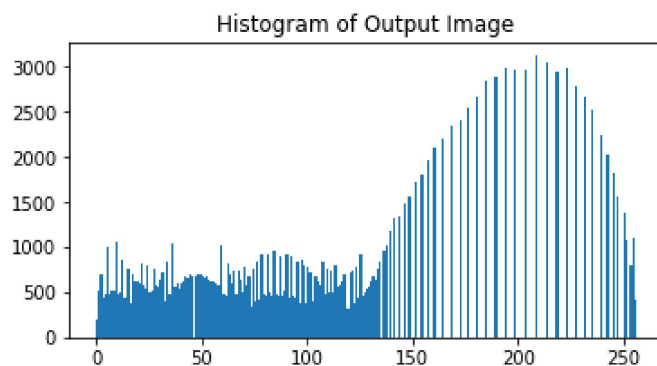
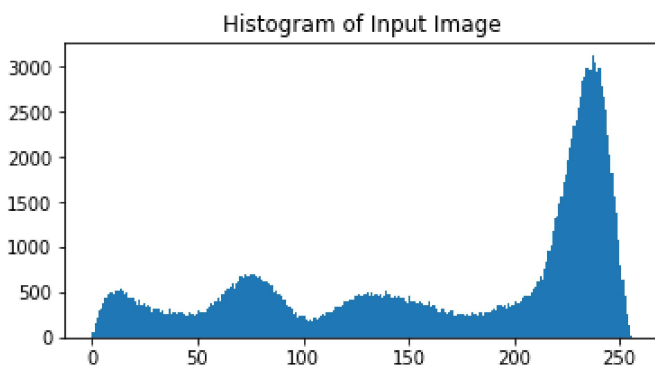
axis[1].set_title("Output Image")
axis[1].imshow(output, "gray")
plt.show()
```



Histogram

```
In [ ]: figure , axis = plt.subplots(1,2, figsize=(13,3))
axis[0].set_title("Histogram of Input Image")
axis[0].hist(inp_img.ravel(),256,[0,256])

axis[1].set_title("Histogram of Output Image")
axis[1].hist(output.ravel(),256,[0,256])
plt.show()
```



CDF

```
In [ ]: figure , axis = plt.subplots(1,2, figsize=(13,3))
axis[0].set_title("CDF of Input Image")
axis[0].plot(cdf_input)
axis[1].set_title("CDF of Output Image")
axis[1].plot(cdf_output)
plt.show()

plt.plot(cdf_input, label="input")
plt.plot(cdf_output, label="output")
```

```
plt.title("CDF")  
plt.legend(loc="upper left")  
plt.show()
```

