



Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information
Technology

**Department of
Artificial Intelligence and Data Science**

Name: Siddhesh Dilip Khairnar

Class: TY

Division: B

Roll No: 372028

Semester: V

Academic Year: 2023-2024

Subject Name & Code: Image Processing: ADUA31205(B)

Title of Assignment: Perform histogram equalization and matching.

Date of Performance: 22-08-2023

Date of Submission: 29-08-2023

ASSIGNMENT NO. 3

IP Assignment 3

Title : → Perform histogram equalization & matching

Learning Objective : →

- 1) Learn about histogram equalization & matching
- 2) To write code in python using opencv for the same

Software Requirement : → vs code

Theory : →

Histogram equalization is a method in image processing of contrast adjustment using histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensity can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. This method is useful in image with background & foreground that are both bright or both dark.

Open cv has a function to do this, cv2.equalizeHist(). Its input is just grayscale image and output is our histogram equalized image.

Example : →

Given below are two histogram

Modify histogram 1 as given by histogram 2.

Gray level (r_k)	0	1	2	3	4	5	6	7
no. of Pixel (n_{k1})	80	100	90	60	30	20	10	0
no. of Pixel (n_{k2})	0	0	0	60	80	100	80	90

Equalizing histogram 1

Gray level	n_{k_1}	PDF (n_{k_1}/n)	CDF	CDF $\times 7$	Histogram Eq. level	new n_{k_1}
0	80	0.20	0.20	1.40	1	80
1	100	0.25	0.45	3.15	3	100
2	90	0.25	0.68	4.76	5	90
3	60	0.15	0.83	5.81	6	90
4	30	0.07	0.90	6.30	6	90
5	20	0.05	0.95	6.65	7	30
6	10	0.02	0.97	6.79	7	30
7	0	0	0.97	6.79	7	30
	390					

Equalizing histogram 2 :-

Gray level	n_{k_2}	PDF	CDF	CDF $\times 7$	Histogram Eq. level	new n_{k_2}
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	60	0.15	0.15	1.05	1	60
4	80	0.20	0.35	2.45	2	80
5	100	0.25	0.60	4.20	4	100
6	80	0.20	0.80	5.60	6	80
7	70	0.17	0.97	6.79	7	70
	390					

Mapping :-

- ① Take first & last column of Histogram 2.
- ② Take last two column of Histogram 1.

Gray level	Histogram eq. level (2)	Histogram eq level (1)	new (1) n_{k1}
0	0	1	80
1	0	3	100
2	0	5	90
3	1	6	90
4	2	6	90
5	4	7	30
6	6	7	30
7	7	7	30

Gray level	0	1	2	3	4	5	6	7
no. of Pixel	0	0	0	80	80	100	90	30

Notes : If eq. level (1) not matching to eq. level (2) take nearest before value.

Conclusion : - Thus we successfully performed histogram equalization & matching & wrote Python code for the same.

Signature
3/10/23

Program Code:

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

path = "pexels-jonathan-borba-3076516.jpg"
img = cv.imread(path)

cv.imshow('Original Image', img)

hist, bins = np.histogram(img.flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()

plt.plot(cdf_normalized, color='b')
plt.hist(img.flatten(), 256, [0, 256], color='r')
plt.xlim([0, 256])
plt.legend(('CDF', 'Histogram'), loc='upper left')
plt.show()

graying = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
cv.imshow('Grayscale Image', graying)

equ = cv.equalizeHist(graying)
cv.imshow('Equalized Image', equ)

hist, bins = np.histogram(equ.flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()

plt.figure()
plt.plot(cdf_normalized, color='b')
plt.hist(equ.flatten(), 256, [0, 256], color='r')
plt.xlim([0, 256])
plt.legend(('CDF', 'Histogram'), loc='upper left')
plt.show()

cv.waitKey(0)
cv.destroyAllWindows()
```

Output:



