8 Exercises

Exercise 1: Write a function to perform histogram equalization from given an image. (Do not use the built-in function in OpenCV)

Exercise 2: Write a function to convert the image to a specific histogram. The parameters of this function include two images $f(x, y)$ and $g(x, y)$.

Exercise 3: Write a function to compute:

1. The sample mean m_s and the sample variance σ_s^2

$$m_s = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)$$

$$\sigma_s^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} (f(x, y) - m_s)^2$$

where MN is the total number of pixels in the image.

2. The average value of the intensities m_h and the variance of the pixels σ_h^2 in the image using the histogram.

$$m_h = \sum_{i=0}^{L-1} r_i p(r_i)$$

$$\sigma_h^2 = \sum_{i=0}^{L-1} (r_i - m_h)^2 p(r_i)$$

Exercise 4: Write a function to compute the average value $m_{s_{xy}}$ of the intensities in sub-image S_{xy} and the variance $\sigma_{s_{xy}}^2$ of the intensities in sub-image S_{xy}. The sub-image S_{xy} has specific size, centered on (x, y).

$$m_{s_{xy}} = \sum_{i=0}^{L-1} r_i p_{s_{xy}}(r_i)$$

$$\sigma_{s_{xy}}^2 = \sum_{i=0}^{L-1} (r_i - m_{s_{xy}})^2 p_{s_{xy}}(r_i)$$

where $p_{s_{xy}}$ is the histogram of the pixels in region S_{xy}, and L denotes the possible intensity values in the input image.

Exercise 5: Write a function to perform the histogram equalization on the sub-image of the input image.

Exercise 6: Write a function to adjust contract stretching image $f(x, y)$ in range $[a, b]$ and show the histogram of image before and after performing the processing. (Do not use the built-in function in OpenCV)

Exercise 7: Write a your own function to convert to binary image from given an image with specific threshold. Show the histogram of image before and after performing the processing.(Do not use the built-in function in OpenCV)

Exercise 8: Write a function to compare two histograms. The method can be as follows:

- Compute the correlation between the two histograms

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \overline{H_1})(H_2(I) - \overline{H_2})}{\sqrt{\sum_I (H_1(I) - \overline{H_1})^2 \sum_I (H_2(I) - \overline{H_2})^2}}$$

  where

$$\overline{H_k} = \frac{1}{N} \sum_J H_k(J)$$

  and N is the total number of histogram bins.

- Compute the Chi-Squared distance between two histograms

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

- Compute the intersection between two histograms

$$d(H_1, H_2) = \sum_I min(H_1(I), H_2(I))$$

- Bhattacharyya distance used to measure the "overlap" between the two histograms

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\overline{H_1} \overline{H_2} N^2} \sum_I \sqrt{H_1(I) H_2(I)}}}$$

  where

$$\overline{H_k} = \frac{1}{N} \sum_J H_k(J)$$

Do not use the built-in function in OpenCV.

> **Note:** OpenCV library provides: *cv2.compareHist(h1, h2, method)* to compare two histograms with methods including
>
> - cv2.HISTCMP_CORREL
> - cv2.HISTCMP_CHISQR
> - cv2.HISTCMP_INTERSECT
> - cv2.HISTCMP_BHATTACHARYYA
>
> ```
> hist1 = cv2.calcHist([gray1],[0],None,[256],[0,256])
> hist2 = cv2.calcHist([gray2],[0],None,[256],[0,256])
> d = cv2.compareHist(hist1, hist2, cv2.HISTCMP_CORREL)
> ```

9 References

1. R. C. Gonzalez, R. E. Woods. Digital Image Processing. New Jersey, Prentice Hall, 2002
2. Reference 1
3. Reference 2