

## 1. Header file

"hist\_func.h"

-purpose of the code

In this code, the essential function for calculating PDF and CDF are defined. Also, constant variables are defined for convenient in developing those codes. When you build the codes below, you should include this header file.

## 2. PDF/CDF Generation

- purpose of the code

this code is for calculating pdf and cdf value from grayscale input image. the function that calculates pdf and cdf is defined in "hist\_func.h" header file. When there is a histogram of an image, if we normalize that histogram by dividing the number of pixels, it is a PDF value. And CDF is the value cumulatively adding PDF value.

-environment

visual studio 2019 with opencv 2.4.13.6

- how to run this code

```
cl PDF_CDF.cpp
```

-how to adjust parameters

to change input image: edit "input.jpg" at line number 5

```
Mat input = imread("input.jpg", CV_LOAD_IMAGE_COLOR);
```

## 3. Histogram Stretching

- purpose of this code

This is a code that implements histogram stretching for grayscale input. For enhancing the contrast, histogram stretching is used. When intensity between x1 and x2 should be stretched, we can control the range of intensity by adjusting histogram stretching function's parameters.

-environment

visual studio 2019 with opencv 2.4.13.6

- how to run this code

```
cl hist_stretching.cpp
```

-how to adjust parameters

to change input image: edit "input.jpg" at line number 7

```
Mat input = imread("input.jpg", CV_LOAD_IMAGE_COLOR);
```

to change a stretching range: edit the vlaue of x1, x2, y1, y2 at line number 27

```
linear_stretching(input_gray, stretched, trans_func_stretch, x1, x2, y1, y2);
```

#### **4. Histogram Equalization(gray, RGB, YUV)**

- purpose of this code

This is a code that implements histogram equalization for grayscale/color\_RGB/color\_YUV input. this is a procedure that enhances the contrast. When histogram equalization is performed on color image, RGB channels should be converted to YUV and then among YUV channels, only Y channel should be equalized to prevent a color distortion. To get to know why we should do histogram equalization on Y channel, here is the example of histogram equalization on RGB channels.

-environment

visual studio 2019 with opencv 2.4.13.6

- how to run this code

```
cl hist_eq.cpp
```

```
cl hist_eq_RGB.cpp
```

```
cl hist_eq_YUV.cpp
```

-how to adjust parameters

to change input image: edit "input.jpg" at line number 7

```
Mat input = imread("input.jpg", CV_LOAD_IMAGE_COLOR);
```

#### **5. Histogram Matching(gray, color)**

- purpose of this code

Histogram matching generates a processed image that has a specified histogram. There are three steps to implement a histogram matching. First, compute the transfer function for histogram equalization of an input image. Second, compute the transfer function for histogram equalization of a reference image. Third, apply the intensity mapping using transfer function for input image and

transfer function inversed for reference image. The histogram results of implementing these codes are similar to the histogram of reference image. When we develop a histogram matching on color image, we should follow the steps as hist\_eq\_YUV is implemented.

-environment

visual studio 2019 with opencv 2.4.13.6

- how to run this code

```
cl hist_matching.cpp
```

```
cl hist_matching_Color.cpp
```

-how to adjust parameters

to change input image: edit "input.jpg" at line number 8

```
Mat input = imread("input.jpg", CV_LOAD_IMAGE_COLOR);
```