HW4 Report

Computer Vision
Section 02
21800181 Kim Jisu

- **HW 4**

- Purpose

This is a program that read image "moon.png" and do histogram equalization. Original image and equalized image are input, draw histogram(bin = 16) and put the value , number of each component of a normalized histogram(bin = 8) of the input image, on the image.

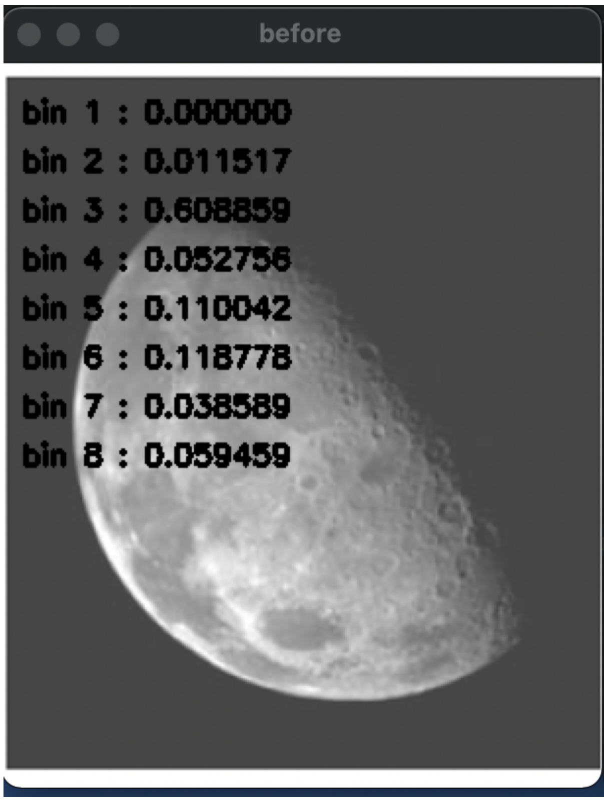
- Principle

int cvRound(double value)
Rounds floating-point number to the nearest integer.
void cv::calcHist(const Mat * images, int nimages, const int * channels, InputArray mask, OutputArray hist, int dims, const int * histSize, const float ** ranges, bool uniform = true, bool accumulate = false)
Calculates a histogram of a set of arrays.
void cv::normalize (InputArray src, InputOutputArray dst, double alpha = 1, double beta = 0, int norm_type = NORM_L2, int dtype = -1, InputArray mask = noArray())
Normalizes the norm or value range of an array.
void cv::equalizeHist(InputArray src, OutputArray dst)
Equalizes the histogram of a grayscale image.

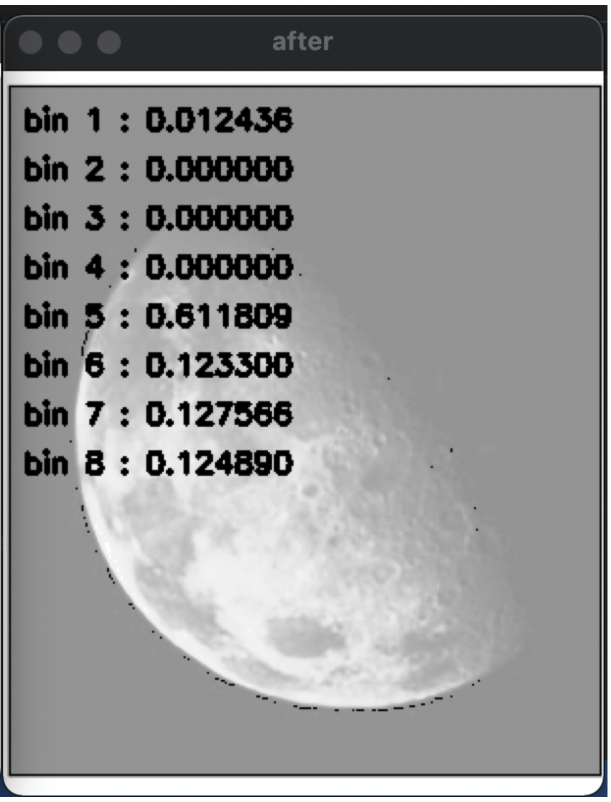
- Process

Read "moon.png" image and perform histogram equalization.
Draw histogram of each image.(bin = 16)
Put the value , number of each component of a normalized histogram(bin = 8) of the input image, on the image. (bin = 8)
Display output image.

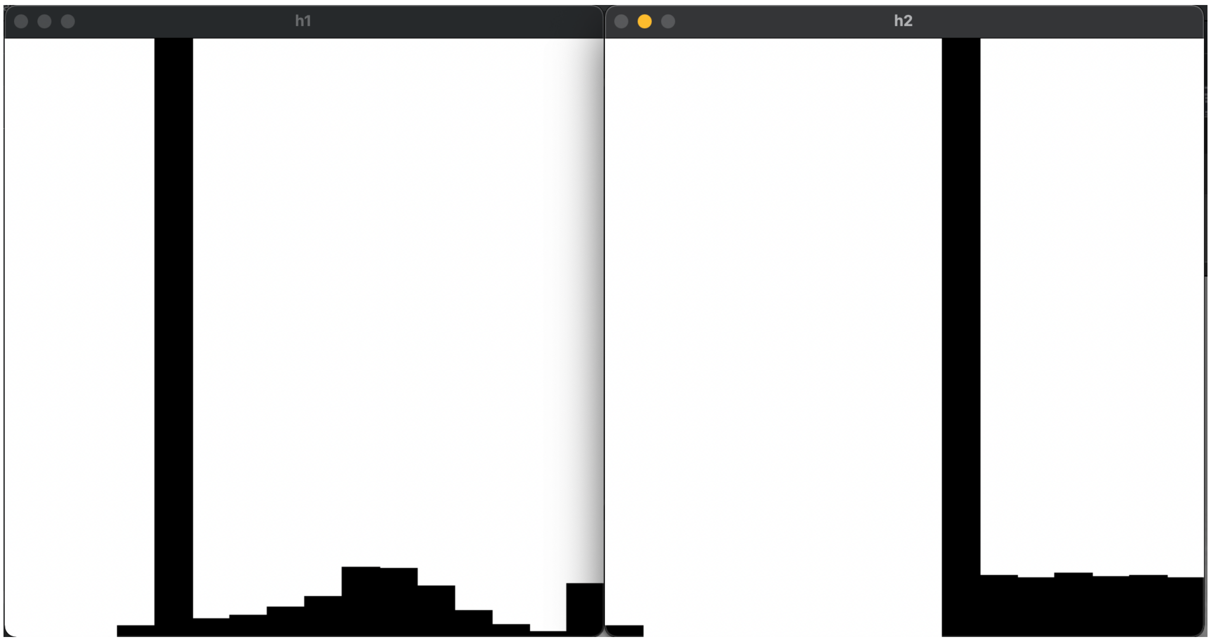
- Result



```
bin 1 : 0.000000
bin 2 : 0.011517
bin 3 : 0.608859
bin 4 : 0.052756
bin 5 : 0.110042
bin 6 : 0.118778
bin 7 : 0.038589
bin 8 : 0.059459
```



```
bin 1 : 0.012436
bin 2 : 0.000000
bin 3 : 0.000000
bin 4 : 0.000000
bin 5 : 0.611809
bin 6 : 0.123300
bin 7 : 0.127566
bin 8 : 0.124890
```



- Code

```
#include "opencv2/opencv.hpp"
#include <iostream>
using namespace std;
using namespace cv;

Mat drawHistogram(Mat src){
    Mat hist, histImage;
    int hist_w,hist_h,bin_w,histSize;
    float range[] = {0,256};
    const float∗ histRange = {range};
    hist_w = 512;
    hist_h = 512;
    histSize = 16;
    bin_w = cvRound((double)hist_w/histSize);
    histImage = Mat(hist_h, hist_w, CV_8UC3, Scalar(255, 255,
255));
    calcHist(&src, 1, 0, Mat(), hist, 1, &histSize, &histRange
);
    normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, −1,
Mat());
    for(int i = 0;i <histSize;i++){
        rectangle(histImage, Point(bin_w ∗ i, hist_h), Point(bi
n_w ∗ i+hist_w/histSize, hist_h − cvRound(hist.at<float>(i)))
, Scalar(0, 0, 0), −1);
    }
    return histImage;
}
```

```
Mat draw_component_number(Mat src){
   Mat hist, histImage;
   int hist_w,hist_h,bin_w,histSize;
   float range[] = {0,256};
   const float* histRange = {range};
   hist_w = 512;
   hist_h = 512;
   histSize = 8;
   bin_w = cvRound((double)hist_w/histSize);

   histImage = Mat(hist_h, hist_w, CV_8UC3, Scalar(255, 255,
255));
   calcHist(&src,1, 0, Mat(), hist, 1, &histSize, &histRange);
   int plus = 0;
   for(int i = 0;i <histSize;i++)
   {
      plus+=cvRound(hist.at<float>(i));
   }
   for(int i = 0;i <histSize;i++){
   putText(src,format("bin %d : %f",i+1,hist.at<float>(i)/pl
us),
Point(10,30+25*i),FONT_HERSHEY_SIMPLEX,0.5,Scalar(0,200,200)
,2);
   }
   return src;
}
int main() {
   Mat image;
   Mat hist_equalized_image;
   Mat hist_graph;
   Mat hist_equalized_graph;

   image = imread("moon.png",0);
   if (!image.data) exit(1);
   equalizeHist(image, hist_equalized_image);
   hist_graph = drawHistogram(image);
   hist_equalized_graph=drawHistogram(hist_equalized_image);
   image = draw_component_number(image);
   hist_equalized_image=draw_component_number(hist_equalized
_image);
   imshow("before", image);
   imshow("after", hist_equalized_image);
   imshow("h1", hist_graph);
   imshow("h2", hist_equalized_graph);
   waitKey(0);
   return 0;
}
```

- References

https://docs.opencv.org/master/