

BLG 453E Homework - 1

Histogram Matching

Musa Anıl Doğan

150130053

20.10.2018

Introduction	2
Process	2
1. <i>Making an interface for showing image and plots</i>	2
2. <i>Loading image and set into the array</i>	3
3. <i>Making histogram of array and create LUT</i>	3
4. <i>Matching histogram and get result image</i>	5
Conclusion	6

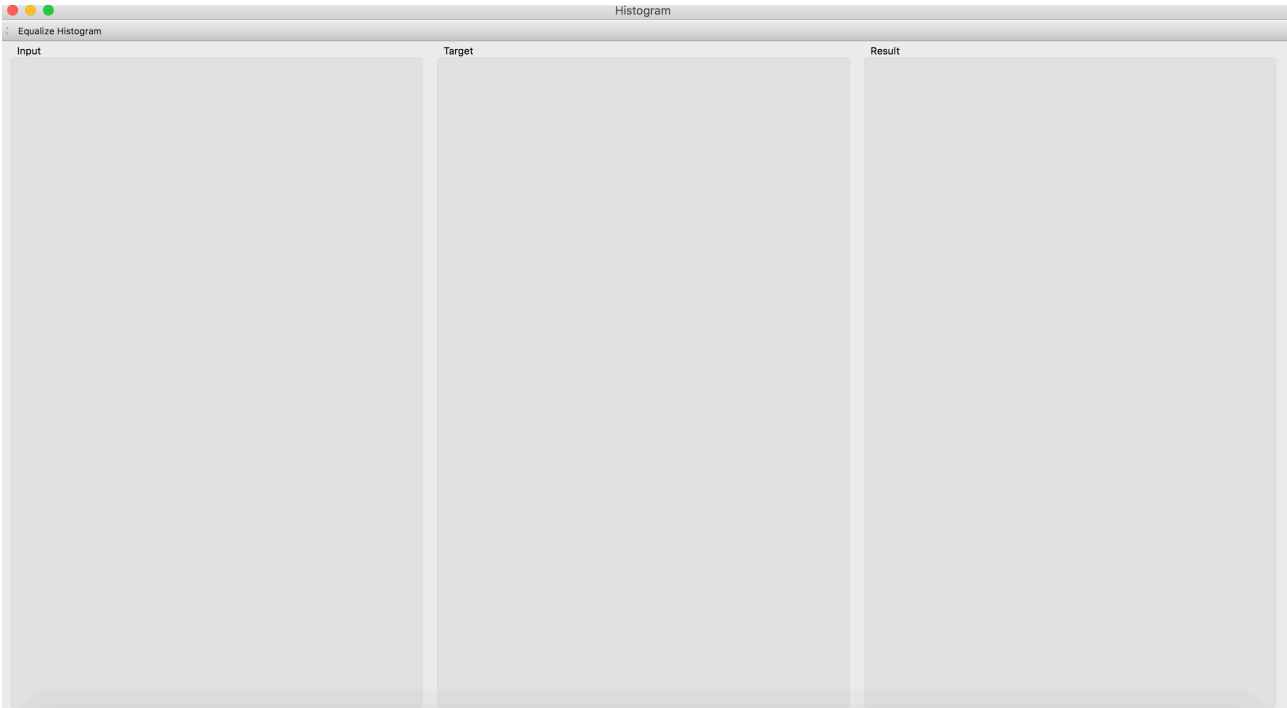
Introduction

First of all, in this project, should understand how to make histogram of image, how to convert pdf histogram to cdf histogram and how to make histogram matching. There are two images; one of them is input image and one of them is target image. Therefore, should match input’s histogram and target’s histogram to reach result remapped image, based on the input image.

Process

Histogram matching process is like that;

1. Making an interface for showing image and plots



INTERFACE OF HISTOGRAM MATCHING

Using Widget function to create widget then create QGroupBox widget to create 3 different box for interface(Input, Target, Result).

In the GroupBox, layout should be set because layout is needed that for upload image and plots to the interface.

Finally, set these widget to the main widget

2. Loading image and set into the array

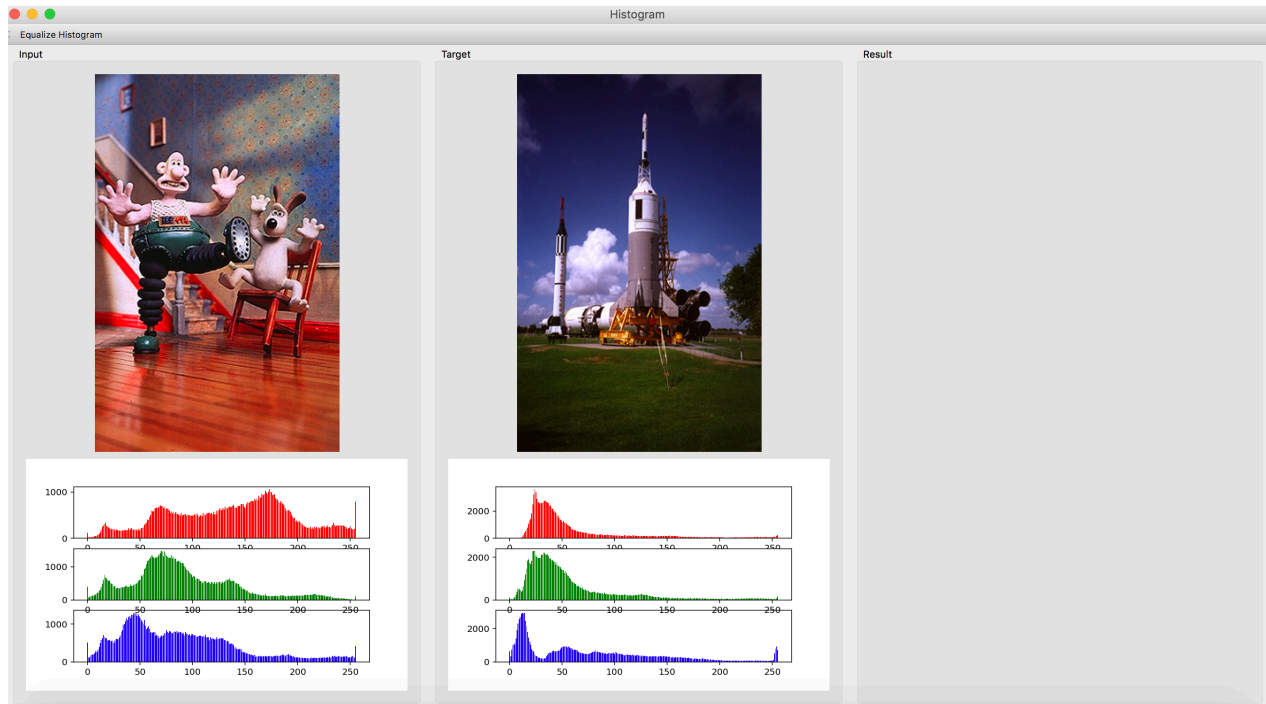


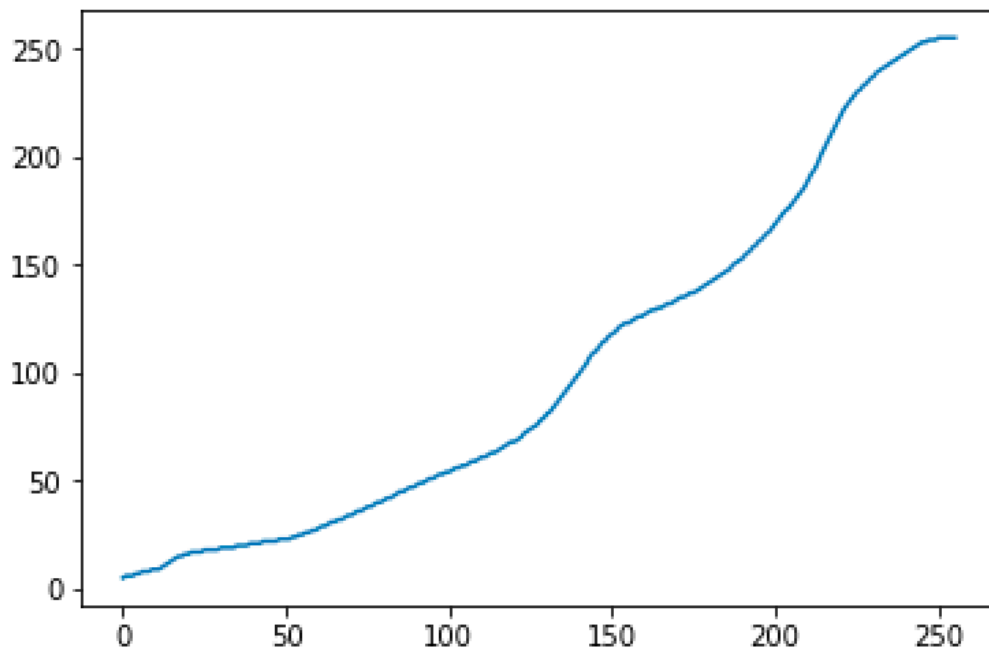
IMAGE AND HISTOGRAM LOADING TO THE SCREEN

Using OpenCV for reading image and then put it into the array for creating histogram of image.

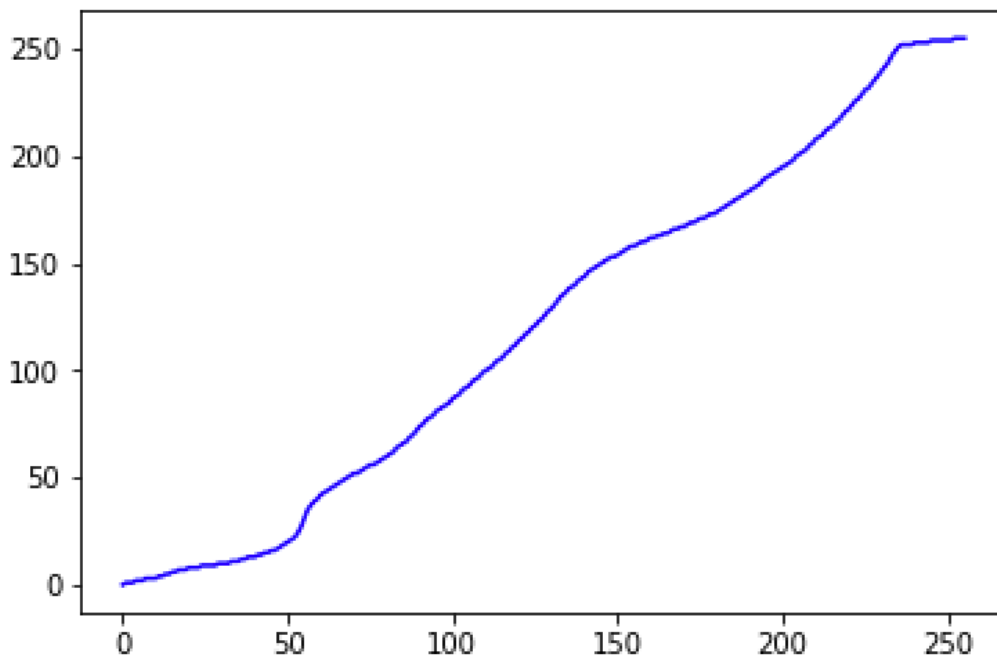
Using imread function to load image and then using QImage class constructor function to convert image to widget for setting image to label.

Finally, Image is set to layout and display on main widget.

3. Making histogram of array and create LUT

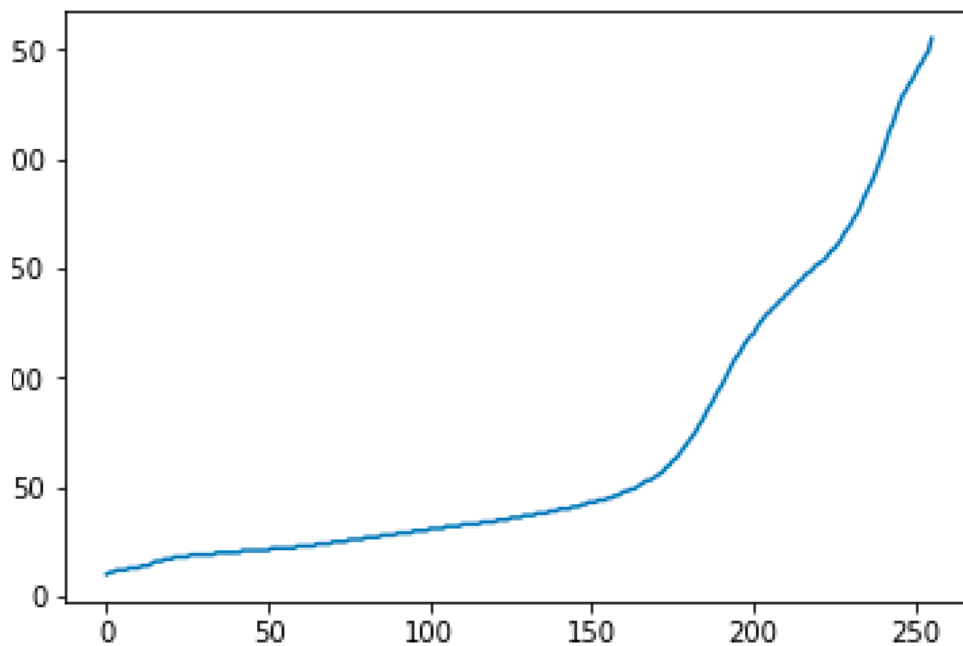


LUT(LOOK UP TABLE) OF CHANNEL GREEN



LUT(LOOK UP TABLE) OF CHANNEL BLUE

Using pseudocode of LUT histogram matching from lecture slide and implementing on the code



LUT(LOOK UP TABLE) OF CHANNEL RED

First, converting all channel's histogram to cdf type and then calculating LUT for every channel.

4. Matching histogram and get result image

Matching histogram is final part of this homework. For matching, first of all, check all item of channel array and use them as index for LUT of channels then equalize with new array. This kind of algorithm is useful for this situation;

```
#####
```

```
h=0
```

```
w=0
```

```
for w in range(0,self.width):
```

```
    h=0
```

```
    for h in range(0,self.height):
```

```
        matchRed[w][h] = self.lookupRed[self.red[w][h]]
```

```
#####
```

- matchRed is a new array
- self.width is a width of red channel array
- self.height is a height of red channel array
- lookupRed is LUT of red channel
- self.red is red channel array
- h and w are index of height and width

Conclusion

The development of my skills such as "python", "opencv", "image processing" and "PYQT5" was very nice for me. At the same time, putting this project into "github" and developing github skills is also an important development for our future business life.