**CV Project 1 Report**

WeiSong wxs170130

Program 1 performs linear scaling in the Luv domain, and writes the scaled image as output.

Program 2 performs histogram equalization in the Luv domain, and writes the scaled image as output.

I made both programs by python OpenCV. There are several decisions that I had to make for both programs.

1. Handle division by 0 in my program. When transferring XYZ to Luv, there are functions u’ = 4X/d, v’ = 9Y/d, if d equals to 0, I directly assign 0 to L, u , v separately. Same situation happens in transforming Luv back to XYZ where L may equals to 0. I assign 0 to X,Y,Z separately to avoid runtime error.
2. After fetching b, g, r from input image, and divide them by 255, there are values not in range [0, 1]. In this situation, I clipped the bad pixels by limiting those whose value is less than 0 to 0 and those whose value is grater than 1 to 1.

Result:

Program 1:

|  |  |
| --- | --- |
|  |  |
| Input image | Output Image |

Output image becomes brighter compare to original image.

Program 2:

|  |  |
| --- | --- |
|  |  |
| Input image | Output Image |

Output image becomes brighter compare to original image.

Bad situation:

If the window in the original image has multiple colors or different luminance while the window locates in an area consists of monotonous color or luminance, then the output image looks pretty bad.

Example: w1 = 0.05, h1 = 0.05, w2 = 0.1, h2 = 0.1

|  |  |
| --- | --- |
|  |  |
| Input image | Output Image |

Description of the machine-readable files:

Files include pj1\_1.py and pj1\_2.py, which are written by PyCharm IDE.