**Project 1 --- Color Project**

***Attention:***

According to the requirements of the project, I wrote 9 required programs and an assistant function library: myModule.py in python. I implemented linear stretching and class histogram equalization functions in myModule.py. You should keep all 10 programs in the same directory.

***Overview:***

My code firstly reads the target image, and then selects the target region. Use the interface provided by openCV to convert every pixel in target region from BGR to Lab, Luv or XYZ. Then it uses linear stretching, histogram equalization, or class histogram equalization to stretch the illumination and finally transfers back to BGR. Discuss the pros and cons of the nine solutions by comparing the results.

A picture containing sky, indoor

Description automatically generated

A sky view looking up at the camera

Description automatically generated

It can be clearly seen that these illumination stretching functions do increase the visibility of the target region. But the degrees of improvements are different, and the color of images are varied lightly or heavily.

***Questions:***

1. Describe strange behavior when colors appear to be changing. If this occurs it is an indication that the OpenCV code does not handle out of range values properly. You should report it and show an image where this occurs.



You can see that the part circled in red that should be white in the picture becomes black or pink, and these exceptions only occurs in the picture that uses class histogram equalization in white region.

1) Assume it was caused by out-of-range.

I implemented an out-of-range check function before and after color space change:

*# Convert back to BGR*check\_range(image3)*# check range*image4 = cv2.cvtColor(image3, cv2.COLOR\_Luv2BGR)  
check\_range(image4)*# check range*

A screenshot of a cell phone

Description automatically generated

But our-of range behaviors did not appear.

2) Assume it was caused by cv2.cvtColor()

I made an inverse control group which is an invert image of original image by photoshop. Imply all three class histogram equalization functions then.



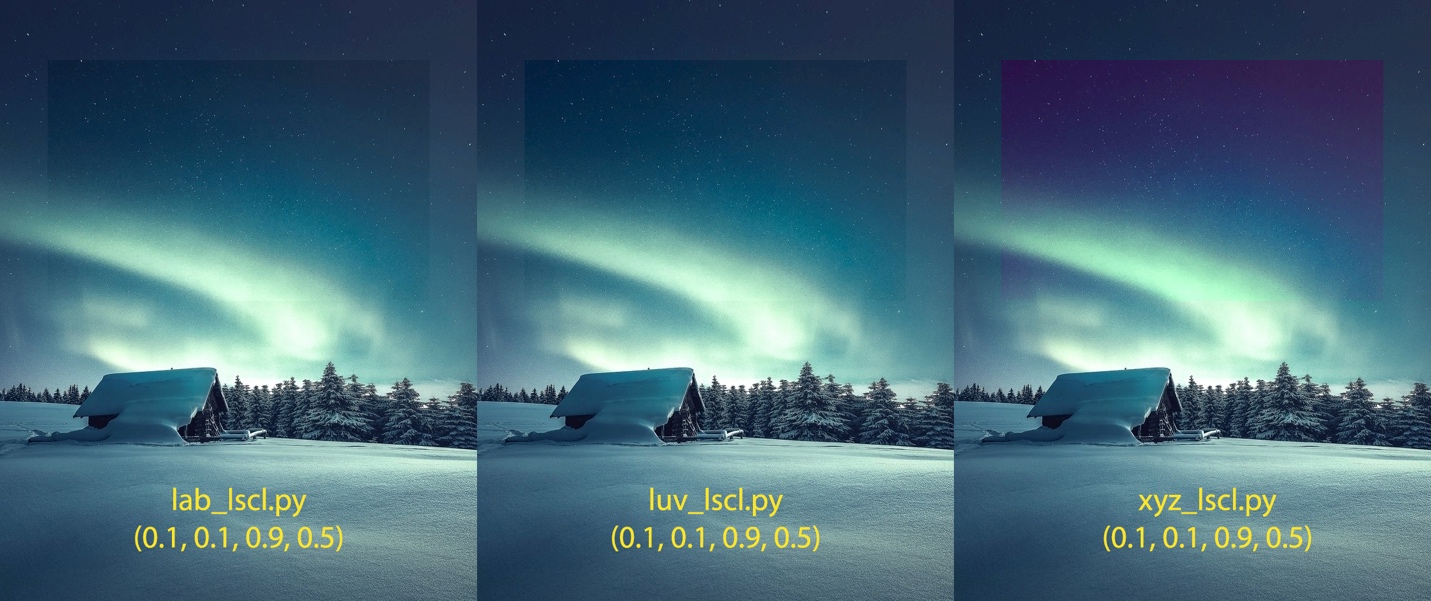
It can be seen that the exceptions only appear in the center of white, and the out-of-range during color space change may cause these exceptions. I think that the reason why the histogram equalization provided by openCV does not generate exceptions is because some possible optimization for out-of-range might be implied already.

2. Among the 3 lscl programs decide which one is the best and which one is the worst. Show images that support your conclusion.

A sky view looking up at the camera

Description automatically generated

White (255, 255, 255) and black (0, 0, 0) are almost always present in natural landscape photos. So, when the selection range is large enough, linear stretching does not change any value. Narrow down the selection range.



A picture containing nature

Description automatically generated

As the size of selection range decreases, the range of color decreases, and linear stretching begins to take effect.

From the results I got, lab\_lscl.py is the best for the contrast is the biggest, and xyz\_lscl.py will change the color to pink which is the worst.

3. Among the 6 histeq programs decide which one is the best and which one is the worst. Show images that support your conclusion.

A picture containing sky, indoor

Description automatically generated

OpenCV histeqs are better than class histeqs, because there is no exception in OpenCV histeqs. Among openCV histeqs, the lab\_histeq.py is the best, because the contrast is highest among all. Among class histeqs, xyz\_classhisteq.py is the worst, because there is an exception in white region and the color is changed to green.