Indian Institute of Technology Kanpur

Department of Electrical Engineering

Image Processing : EE604A

Photo Editor

Prepared By:

Sagar Chand 14579

Supervisor:

Professor Tanaya Guha

1 Introduction

Two functionalities namely red eye correction and auto-enhancement of image have been implemented in python. TkInter library has been used to create GUI for the project.

2 GUI in action

The GUI has been created in Python using TkInter library which allows us to perform basic functionalities. Our GUI starts with a basic pop like small window with 2 options. One of RedEye and another of Auto Enhance depicting the 2 features implemented in this project.

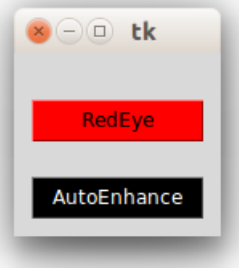

Figure 1: GUI 1

The colors are chosen such that RedEye is background filled with Red color In autoenhance, the color of the button changes (when mouse hovers over it) from extreme black and white to Grayish shade depicting contrast stretching as can be seen below.

Figure 2: GUI 2

When any of the button is pressed, a new window opens which asks the user to select the image on which we have to perform operations.

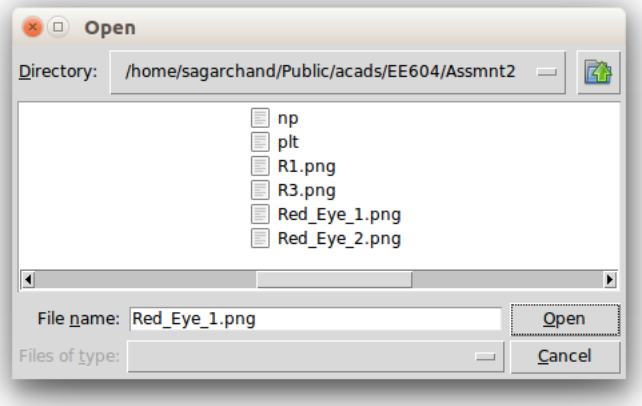

Figure 3: GUI 3

After image has been selected, and open button is pressed, this window closes and a new window opens up with original image on the left and the transformed image on the right.
There is also option to perform transforms on another image with the help of buttons given at the bottom of GUI as can be seen below.

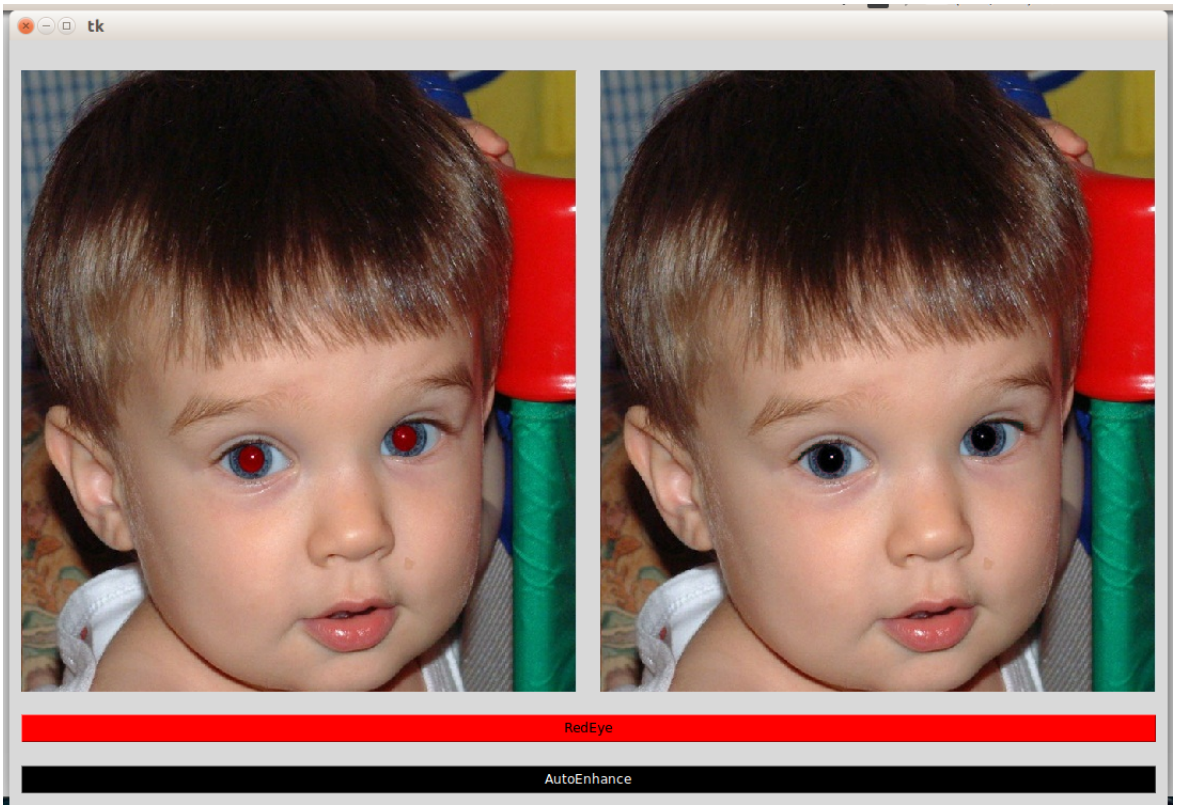

Figure 4: GUI 4

3 METHOD USED

3.1 Red Eye Correction

Using dlib's trained model [1] which detects 68 landmarks points in human face. These are points on the face such as the corners of the mouth, along the eyebrows, on the eyes, and so forth. There are 6 points for each eye as well. After detecting these points, I have made a rectangle around it and then redness in this area is spotted and subsequently removed.

Figure 5: 68 landmark points in human face.

See in image above that feature points from 36 to 41 are for left eye and from 42 to 47 are for right eye.

The beauty of the code is that is can do red eye correction for any number of images and faces can be of any size and red eye error need not be circular.

3.2 Auto Enhancement

For this method, a lot of filters like Wiener filter, NL means and bilateral filters were tried with variation in their hyper-parameters. Bilateral filters with σs=15 and σr=0.1 were found suitable For contrast stretching histogram equalization and adaptive histogram equalization(by varying its clip limit) were tried. Adaptive histogram equalization with clip limit 0.03 was found suitable. Only middle 96% of the pixel values were used for contrast stretching.

4 RESULTS

4.1 Red Eye Correction

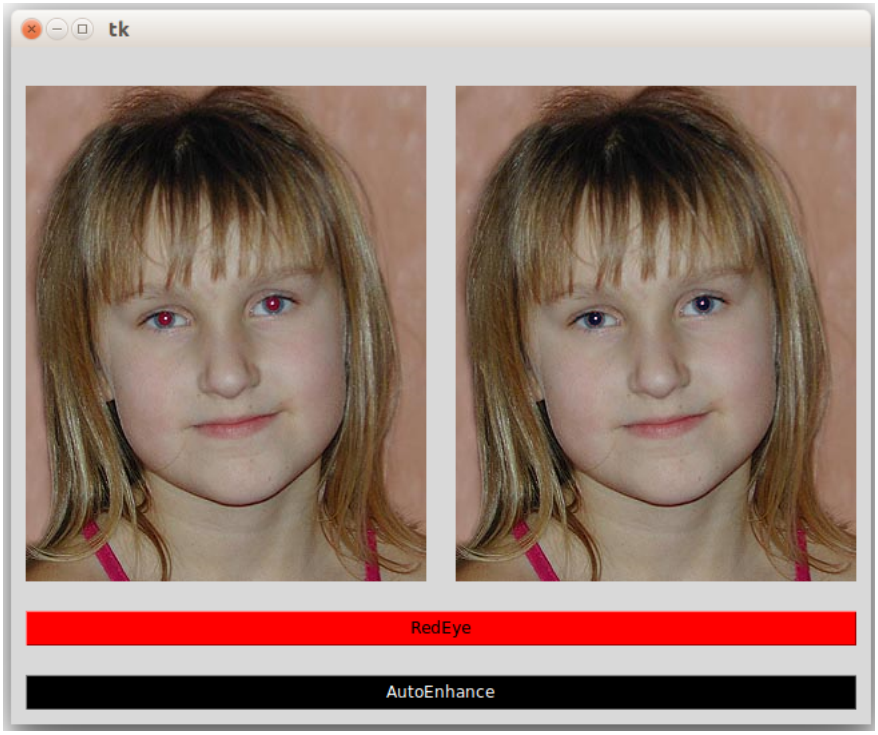

Figure 6: RedEye 1

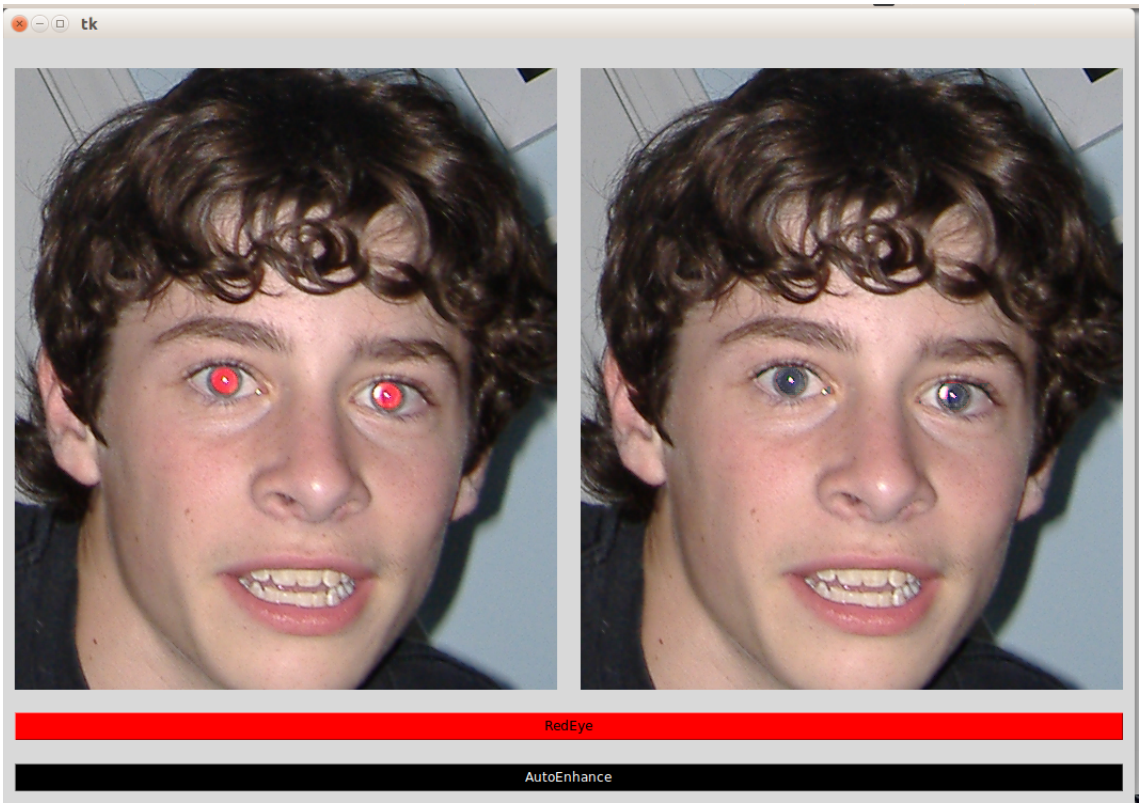

Figure 7: RedEye 2

4.2 Auto Enhancement

Figure 8: AutoEnhance 1

Figure 9: AutoEnhance 2

5 Link to Code

https://github.com/sagarchand9/Photo_Editor

References

[1] Davis E. King(Github:davisking)*dlib-models* 2016