Assignment 1: Brightness and Contrast Adjustments using OpenCV

Elijah Garmon

A report on the creation of a computer vision program using OpenCV
to allow the changing of the brightness and contrast

Department of Computer Science
Florida Polytechnic University, Florida, USA

January 2024

ABSTRACT

In this project, I used the principles I learned in Computer Vision to create a program that adjusts the brightness and contrast of a preview image. It was also required that the original and the preview image appear side by side and that the brightness and contrast be adjusted by sliders whose values would be saved by pressing a specified key. Finally, it was required that two histograms be created, one for the original image and one for the preview image.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

This project started by making a program that allowed the adjusting of the brightness and contrast of the image. Afterwards I created track bars that allowed the live altering of the image, as well as showing a preview of the original image. Next I created a save functionality where at the press of the button it would save the brightness and contrast values for the next launch of the program. Lastly, I created two histograms, one for the original image and one for the newly changed image.

1.1 BRIGHTNESS AND CONTRAST

My first task for this project was to create a program that allowed me to adjust the brightness and the contrast of the image. In order to prevent the contrast or the brightness from causing the image to cycle back round, the image is going to adjust itself by a mod 255 at all times. The contrast is multiplied the image directly and divided by 10. The brightness is being directly added to the image. There is also a maximum value of 100 set for both of these values.

1.2 TRACK BAR

The two trackbars bars are set linked to the same window panel as the images appear in. The variables created previously for brightness and contrast are set to the starting values of the trackbar, and by changing these values it causes an update function to run depending on the variable. Each of these functions runs the changeImage function in order to change the value of the second image. The orignal image and the preview image is set to be stacked next to each other horizontally. Both the trackbars have maximum values of 100, set by a maximum value variable that can be adjusted in the declared variable section.

1.3 SAVING

In order to create the save functionality, I have it pull the values into a file called ”Data.txt”. If Data.txt does not exist, it will create Data.txt with the default values of 10 for the contrast and 0 for the brightness. Using the waitKey(0) and setting it equal to a variable ”key”, if the key value

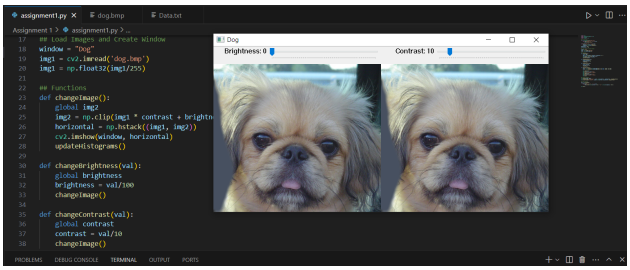

Figure 1.1. Basic layout of the program

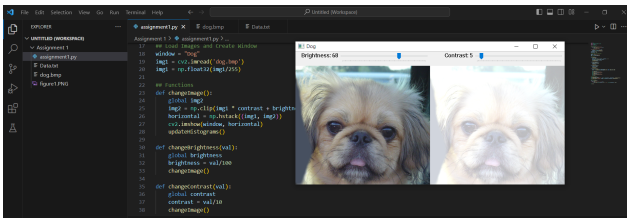

Figure 1.2. Changing of the Contrast and the Brightness

equals 115 (or the letter 's'), it will write over the Data.txt with the current values of brightness and contrast. The next time the file is openned, it will use this new brightness and contrast value. There were several issues with the variable types when doing this part of the project. The original reading of the file casts the variables from a string to a integer for use in the actual program. In order to save the brightness and contrast values, the value is casted as an integer in order to remove the decimal point and then recasted as a string.

1.4 HISTOGRAM

The most difficult aspect of this project was the inclusion of the histograms. Unfortunately I could not get this program to alter the histograms live, however after the brightness and contrasts values are saved or closed, it will show the histograms and the preview however altering the trackbar will cause the program to crash. The histogram will show the values for both the images however they are just incapable of altering it live.

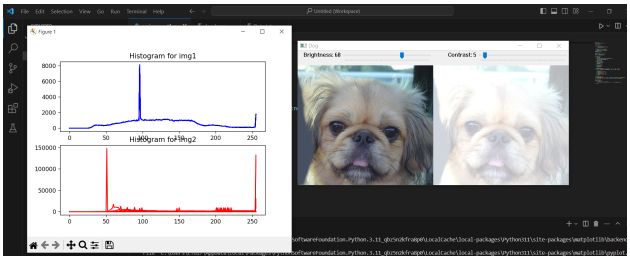

Figure 1.3. Histograms