PIDGEOT IMAGE EDITOR

Muhammet KORKMAZ
Gazi University Engineering Faculty
Computer Engineering
Eti, Yükseliş Sk. No. 5, 06570
Çankaya/Ankara
+90 511 129 00 30
muhammet.korkmaz@gazi.edu.tr

Candan DOĞAN
Gazi University Engineering Faculty
Computer Engineering
Eti, Yükseliş Sk. No. 5, 06570
Çankaya/Ankara
+90 535 047 22 75
candan.dogan@gazi.edu.tr

Ufuk SÜNGÜ
Gazi University Engineering Faculty
Computer Engineering
Eti, Yükseliş Sk. No. 5, 06570
Çankaya/Ankara
+90 541 230 92 97
ufuk.sungu@gazi.edu.tr

ABSTRACT

Image processing is one of the most important fields of study in computer science. Image processing techniques are used to extract useful information by interpreting images. Iris, face and fingerprint recognition processes are performed using image processing methods. Images to be processed can be obtained with the help of technological devices such as cameras and scanners. After the images to be processed are obtained, they are digitized and digitized and by applying different processes, meaningful interpretable results are obtained. In order for these images to be interpreted, pre-processes must be applied on them. There are many libraries developed with programming languages on this subject. OpenCV (Open Source Computer Vision Library), which is an open source library, is one of the image processing libraries preferred in this study. OpenCV functions were used extensively in this project. In the project, there are simple operators such as rotation, cropping, mirroring, manually increasing and decreasing brightness and contrast, inverse, as well as automatic image enhancement methods such as histogram equalization and contrast enhancement. And in addition to all these, 29 filters and effects are available to the user in the interface.

Keywords

Image Processing, OpenCV, Automatic Image Enhancement, Basic Operations, Effects and Filters.

1. INTRODUCTION

The methods that enable performing various operations on images obtained by means of a technological device are called image processing techniques [1]. In order to perform operations on any image, the image must first be digitized. Expressing the colors in the picture with numerical value is called digitization. There are preliminary preparations to be done before processing the image. Some of these are operations such as reducing the noise on the image, rotating it and applying filters [2]. The operations performed in image processing techniques have 3 levels. These are low, medium and high level transactions. Low-level processing is also achieved by filtering the authenticity of the input and output images. In the middle level processes, division and recognition processes are performed in the recognition and classification of objects in the images. High level operations involve analyzing images to recognize objects in images [3].

In this study, simple image processing techniques were applied on the logo on the cover page. These are mirroring, cropping, rotation, brightness increasing and decreasing, contrast increasing and decreasing, inverse, histogram equalization operations, contrast enhancement operations, filters and effects applications. The application is presented to the user with a simple interface.

2. METHODS and MATERIALS

Jupyter Notebook and Pycharm platforms were used while developing this project. The project is written in Python language. While developing the project, OpenCV, Pillow, Numpy, Imageio, Scipy libraries were frequently used. The application and the interface offered to the user are presented through the Pycharm platform. Libraries and versions required to run this project are as shown in the figure.

Package	Version	Latest version
Pillow	8.0.1	8.0.1
PyQt5	5.15.2	5.15.2
PyQt5-sip	12.8.1	12.8.1
imageio	2.9.0	2.9.0
numpy	1.19.4	1.19.4
opencv-python	4.4.0.46	▲ 4.5.1.48
pip	20.3.3	20.3.3
scipy	1.6.0	1.6.0
setuptools	51.1.1	51.1.1

Figure 2.1 Project Libraries

3. BASIC OPERATIONS

3.1 Rotate Image

This operation takes two arguments as image and angle. It simply rotates image to given angle. Then, return the image. It uses PIL library functions.

3.2 Crop Image

This operation takes five arguments as image, top, bottom, left and right. It crops image to given values and return it. It uses PIL library functions.

3.3 Flip Image

This operation takes two arguments as image and flip direction. It flips the image to horizontal or vertical then return it. It uses PIL library functions.

3.4 Image Brightness

This operation takes two arguments as image and rate. It increases image brightness to rate value. It uses PIL library functions.

3.5 Image Contrast

This operation takes two arguments as image and rate. It increases image contrast to rate value. It uses PIL library functions.

3.6 Invert Image

This operation takes one argument as image. Firstly, it changes image to grayscale. Then, it subtracts image colors from 255. Finally, it returns the inverting image. It uses CV2 library functions.

3.7 Histogram Equation

The operation takes one argument as image. Firstly, it changes image to grayscale. Then, it applies histogram equalize and return it. It uses CV2 library functions.

3.8 CLACHE

The operation takes one argument as image. Firstly, it changes image to grayscale. Then, it applies Contrast Limited Adaptive Histogram Equalization and return it. It uses CV2 library functions.

4. ARTISTIC FILTERS

4.1 Yıng-Yang

In the Yin Yang filter, first the picture sent as a parameter is grayed out, then a black and white image is obtained by applying a certain threshold value. Gauss blur is applied to the final output.

4.2 Corrupted Gray

In the Corrupted Gray filter, a manual value is assigned to the seed value. Then this value is sent to the 'dilated' function together with the image as a parameter. The resulting image is grayed out and the final output is created.

4.3 Broken Television

In the Broken Television filter, while creating this filter, the maximum values of the BGR channels were taken and combined.

4.4 Anime Isn't Cartoon

The filter is an Anime Isn't Cartoon filter. For this filter, the image is grayed out first. Then Median blur was applied to the grayed image. Finally, the image with the Median blur applied was sent to the adaptive Threshold as a parameter and the result image was obtained.

4.5 The Maklube

In the Maklube filter, for this, the size of the image was expanded by using the resize function. Later, the 'row' and 'col' variables were configured according to the new dimensions. Later, a vignette mask was created for "X_resultant_kernel" and "Y_resultant_kernel" by using get Gaussian Kernel function. Then, the result kernel matrix was created by multiplying these values. This matrix was normalized by dividing the numpy library into the linalg function and the resulting output was applied to the BGR channels, resulting in output.

4.6 Back to Black

The filter is the Back to Black filter. The row and column values were determined by converting the image to the matrix. 'Kernel_x' and 'kernel_y' values were created by sending these values to the get Gaussian Kernel function as parameters. These values were combined and assigned to the "filter" variable by applying the normalization process. Finally, the final output was obtained by applying a filter to each BGR (Blue Green Red) channel.

4.7 Breaking Reality

Our another filter is the Breaking Reality filter. In this filter, Canny and bitwise_not functions are used for 'edge1' value. Then, the image has been grayed out with the 'gray' value. Median blur was applied to the grayed image. Edge protection filter has been applied to the obtained image. The outputs obtained with the bitwise_and functions are combined and output is created.

4.8 Roses are Red, Violets are Blue, Sugar is Sweet

Our another filters are the Roses Are Red, Violets Are Blue, Sugar İs Sweet filters. For these filters, we have 2 function definitions named "exponential_function" and "tone". 'Exponential_function', the first function we created, takes 2 parameters. These parameters are 'channel' and 'exp' parameters. The 'channel' parameter indicates the BGR channels. 'Exp' is the exponent value. In this function, firstly, a table is created for the exponent with the numpy array. Then the 'channel' value taken as parameters and the 'table' value created in the function are sent as parameters to the LUT function and the output obtained is returned. 'Tone', the second function we have defined, takes 2 parameters. These parameters are 'img' and 'number' parameters. The 'img' parameter is the current view, the 'number' parameter takes the values 0,1 and 2. When the value 0 is entered, the blue channel is sent as the parameter, the green channel when the value 1 is entered, and the red channel (BGR - Blue Green Red) when the value 2 is entered. Thanks to the if-else control statement defined in the "tone" function, the image in the tone desired by the user is returned.

4.9 Gouache Natürmort

In the Gouache Natürmort filter, for this, the pencil Sketch function has been used and the dst_color value has been returned. The pencil drawing obtained is colored. Then, bilateral blur was applied to the picture obtained and the noises in the picture were removed.

4.10 Fifty Shades of Gray

Another filter is Fifty Shades of Gray filter. The first function created while creating this filter is the 'hsv' function. This function takes 3 parameters as 'img', 'l' and 'u'. The 'l' value contains the smallest (lower) HSV value, the 'u' value contains the highest (upper) HSV value. Then, in Range function is used and a new parameter named 'mask' is obtained and returned. The second function defined for the splash filter is the "splash" function. In this function, firstly a blank mask named 'res' is created for the result. The zeros function of the numpy library is used for this. Then we give the lowest and highest tone values we want to the 'l' and 'u' parameters. In the next step, we call the hsv function that we defined first. In the next step, we inverse the new 'mask' value we have obtained in this step by using the bitwise_not function. In the next step, we are graying out the first picture we take as a parameter. We define the borders of the grayed and colored picture with 'res1' and 'res2' variables. Finally, by using the bitwise_or function, we get the result output by combining 'res1' and 'res2' variables.

4.11 Martini with an Olive

In the Martini With An Olive Filter, "increaseLookupTable" and "decreaseLookupTable" parameters have been defined first. The image was then divided into BGR (Blue-Green-Red) channels.

Then, the LUT function was applied to each divided channel. The purpose of these operations is to warm the picture by increasing the presence of the red channel in the picture and reducing the presence of the blue channel. Finally, the new channels obtained were combined using the Merge function.

4.12 White Russian

Another filter is a White Russian filter. The same operations performed in the Martini Waith An Olive Filter filter are also performed in this filter. The only difference is that the aim is to make the picture cooler by increasing the presence of the blue channel and reducing the presence of the red channel. For this reason, this filters in our project work opposite to each other.

4.13 Piña colada, Long Island Iced Tea

Our another filters are Pina Colada filter and Long Island Iced Tea filter. These filters have been applied separately to the X and Y axes. It has been obtained using the Sobel function. Then, Laplacian was applied to the picture obtained and the noises in the images were removed.

4.14 Faded Leaf

Firstly, faded leaf filter converts image to RGB format. Then, fill the image to (250,100,50) RGB color and blending these two images by using blending difference method. Lastly, it decreases blending image brightness, contrast and increases saturate.

4.15 Lightbringer

Firstly, lightbringer filter converts image to RGB format. Then, fill the image to (66,181,233) RGB color and blending these two images by using blending hard light method. Lastly, it applies %30 grayscale.

4.16 Eternal Sunshine of The Spotless Mind

Firstly, Eternal sunshine of the spotless mind filter converts image to RGB format. Then create three different RGB colors and combined they with a radial color transition different rate. After that, it blends thee two images by using blending screen method. Lastly, it increases image brightness and decrease contrast, saturate.

4.17 Winter is Coming

Firstly, winter is coming filter converts image to RGB format. Then, fill the image to (57,183,217) RGB color and blending these two images in a ratio 0.54. Finally, it decreases image brightness and contrast.

4.18 Yellow Hot Chick

Firstly, yellow hot chick filter converts image to RGB format. Then create two different RGB colors and combined they with a radial color transition. Finally, it blends these two images by using blending lighten method.

4.19 Emotion Yellow Rock

Emotion yellow rock follows same steps with yellow hot chick. It only uses blending soft light method.

4.20 Is so Cute

Firstly, is so cute filter converts image to RGB format. Then, fills original image to two different RGB colors called cs1, cs2. Then create radial gradient mask for highlighting a area in image. Secondly, it composites cs1, cs2 and radial gradient mask and assign it cm variable. Thirdly, it fills original image to (201,177,150) RGB color then blends it with original image and assign cm3 variable. Fourthly, it creates another radial gradient mask and composites cm3, cm, itself. Then it blends these composited images cm and cm2. Finally, it increases brightness and decreases sepia, contrast.

4.21 Color Passion

Firstly, color passion filter converts image to RGB format. Then, fill the image to (75,75,75) RGB color and blending these two images by using blending overlay method. Finally, it increases brightness and hue.

4.22 Limbo Travelers

Limbo travelers follows same steps with color passion. It only uses blending darken method.

4.23 Mordor, Gondor, Matrix, Eric Clapton

These four effects use color map. It applies hot color map to original image in Mordor effect, cıvıdıs color in Gondor effect, deepgreen in Matrix affect, twilight shifted in Eric Clapton effect.

5. GUI

PyQt is a Python link to the Qt library written in C ++ for cross-platform application development. It is not a programming language. It allows us to create graphical user interface programs with Python. There are two versions of PyQt. These; It is PyQt4 created for Qt 4.x and Qt 5.x, and PyQt5 to be used only for 5.x. We will use the PyQt5 version for the operations in the project. Interface programming with Qt is built around signals and slots for communication between objects.

Programs GUI consist of four parts. There are three buttons as select, save, save as on top of screen. There are two image field in middle of main screen. One of image field shows us original image, other image field shows us processed image. In right side of screen, there are seven button and two slider. Buttons are for basic operations and sliders are for brightness, contrast. Bottom side of screen has an area for all artistic filters and a slider for rotating image.

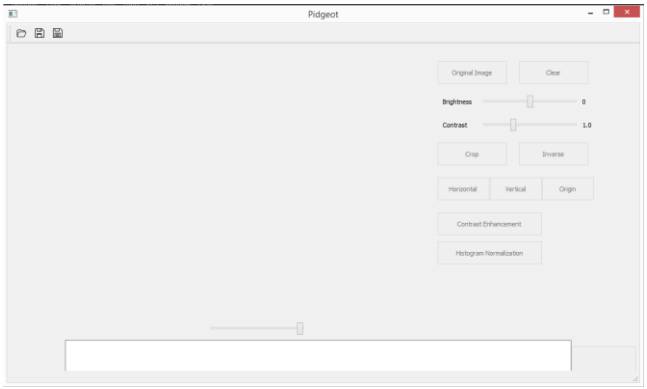

Figure 5.1 GUI

6. ACKNOWLEDGMENTS

Our thanks to Duygu Sarıkaya for teaching us to all image processing techniques.

7. REFERENCES

[1] Internet: https://dergipark.org.tr/tr/download/article-file/391087

[2] Internet: https://polen.itu.edu.tr/bitstream/11527/13574/1/10065111.pdf

[3] Internet: http://www.acarindex.com/dosyalar/makale/acarindex-1423877680.pdf.

Columns on Last Page Should Be Made As Close As Possible to Equal Length