CS512 Assignment 2 - Report

Lin Zhuo

A20379609

Abstract

The report summarized the implementation of simple image manipulations using openCV. In the program the image is loaded either through reading from a file or directly captured from camera. It also provided various options to the users to manipulate the display of image.

1. Problem Statement

The requirements are: - The image should be input from command line, either from the local file or from the camera. If the file name is specified in the command line, the image should be read from it otherwise the program should attempt to capture an image from camera. - The image should be read as 3-channel color image. - The program should be adapt to any size of image. - There are special keys passed to the command line to modify the displayed image.

2. Implementation Details

- Convert the image to gray scale using own implementation of conversion function.

Adopt the formula: $Gray(R, G, B) = 0.299 * R + 0.587 * G + 0.114 * B$

Transform the 3-channel color image with shape $(h, w, 3)$ to gray-scale image with shape (h, w). Note that the opencv requires transforming the data type to `np.unit8` in order to display using `imshow()`.

The code:

```
image_g = image[:,:,0]*0.299 + image[:,:,1]*0.587 + image[:,:,2]*0.114
image_g = image_g.astype(np.uint8)
```

- Cycle through the color channels of the image showing a different channel every time the key is pressed.

Implement this feature by the openCV's function `cv2.split()`. The color channels (blue, green, red) combinations can be obtained. Define a parameter that has the value 0,1,2 to determine the current color channel in cycling. See the code for the details.

- Smooth the image

When the smoothing is applied using openCV's library, the `blur()` function can be used.

Use a N by N matrix with all elements 1 divided by $N * N$ as the smoothing filter for implementing the track bar more easily.

- Downsize the image

Use `resize()` function in openCV.

```
cv2.resize(image, (image.shape[1]/2, image.shape[0]/2))
```

- Derivative and gradient computation

By convolving the x derivative $\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$ and y derivative $\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$ with the image, we get the gradient in x and in y direction.

The gradient can be compute by combining the x and y derivative as:

$Grad(I) = \sqrt{I_x^2 + I_y^2}$

- Draw the gradient vectors

The computed gradients in x and y directions $grad_x$ and $grad_y$ give the information of vector length in each direction. Having the fixed vector length K, it is possible to calculate the length of gradient vector in x and y direction, $grad_x * K/\sqrt{grad_x^2 + grad_y^2}$ and $grad_y * K/\sqrt{grad_x^2 + grad_y^2}$

The function `cv2.line(image, (x1, y1), (x2, y2), (r,g,b), width)` can be used to draw lines on the image object.

3. Result and Discussion

The program can be executed generally in good condition.

4. References

- http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filteri
- http://www.cs.iit.edu/~agam/cs512/share/using-opencv.pdf