CS512 - AS1 - Report

Akshay Jain

Department of Computer Science

Illinois Institute of Technology

February 5, 2022

Abstract

In this assignment, we have performed various transformations on images based on the special keys pressed on the keyboard. To do these operations, we have used the OpenCV library in python and also used Cython to make our program's execution faster.

1. Problem Statement

In this assignment, our aim was to perform various transformations on images like converting to grayscale, rotating the image by some degree, reloading the image, writing the image, cycle through color channels of the image, etc. These image transformations are helpful as we can many times detect some features in the images after transformation which we are not able to identify in the original image. So, we have performed these transformations on images of different sizes and based on the key entered by the user on the keyboard, the operations were performed.

2. Proposed solution

Using the OpenCV library and other supporting libraries like NumPy, we have implemented the program to do transformations on the images. Below are some of the functions that we performed and their working.

A) RGB to Grayscale

We implemented RGB to Grayscale using three different methods:
1) Using the cv2 function COLOR_BGR2GRAY

2) By creating a user defined function and without using any loops - we used the NTSC color encoding formula and took the dot product of the 3D array with the respective values for R, G, B. Below is the formula: - Grayscale = 0.299 * Red + 0.587 * Green + 0.114 * Blue

3) First, we converted the image into a numpy array. Then, with the help of loops, we traversed each pixel in the image based on the length of the image using 2 loops and then multiplied the RGB values with the respective NTSC values as stated in the above formula.

B) Cycle through the color channels of the image showing different color every time the key is pressed Here, using the split() from cv2, we split the image into 3 channels i.e. R, G, B respectively. We also kept a count of the number of times the key was pressed. If the count was 0, Blue color channel of the image would be displayed, if count was 1, Green channel and if count was 2, Red channel. After that, count was reset to 0.

C) Converting the image to grayscale and then rotating the image by some degree using trackbar

We used the createTrackbar() and gave values in the range 0 to 360 representing the angle range. We defined a slide handler function where we calculated the center point of the image and provided it as input to the getRotationMatrix2D(). This function is responsible for rotating the image and we got the value of the angle from the trackbar function. By using the slider, we were able to rotate the image between any value from 0-360 degrees. We also applied affine transformation using function warpAffine() which preserves the points, lines in the image.

D) Some other functions which were implemented were reloading the image, saving the image and disThe image is converted to grayscale and after moving the slider, it gets rotated. So, from the above screenshot, we could see and evaluate that all the transformations associated with the respective input keys worked fine. All the functions worked correctly. 5. References:- ● https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.htmlplaying the description of the program and the keys. We reloaded the image by reading the image again using imread() and saved the image by using imwrite()

3. Implementation details

Some of the problems and design issues which were faced are as mentioned below:

● While trying to install the openCV package in Ubuntu through the terminal, it was not getting installed as there were too many dependent packages which were not getting installed automatically. After trying to install the package directly in anaconda, it worked

● When converting from rgb to grayscale, it was not giving correct output. But, after taking dot product with the NTSC values, we got the expected output, i.e. gray scale image

● When we tried giving keyboard input with the help of waitKey(), the program was going in an infinite loop and not accepting any key as input. Then, after using the input() function, the program was able to take keyboard inputs

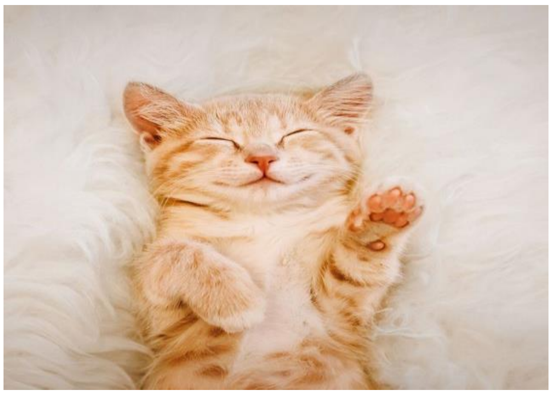
● At first, the program was crashing whenever the imshow() function was called. Later on, after adding waitKey() and destroyAllWindows(), it worked fine

● For the key input ‘G’, as I have used matplotlib, the image will be displayed inline in the command prompt after the execution

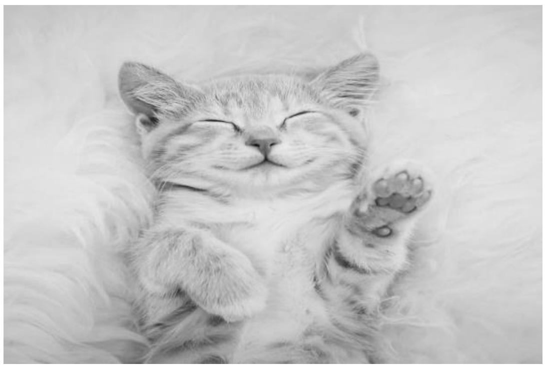
4. Results and Discussions

● Evaluation:

Original image

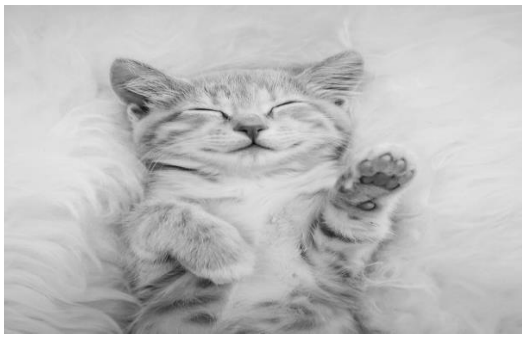


After pressing key ‘g’ - convert to grayscale



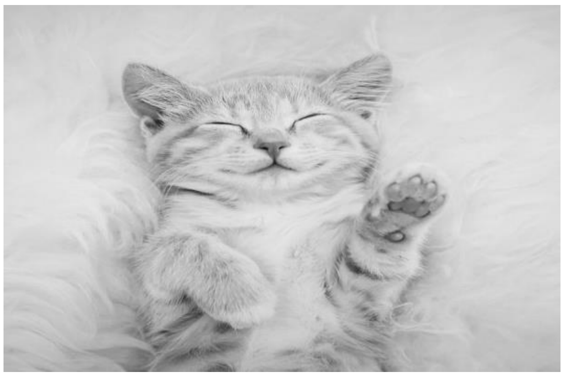
This is implemented using the cv2 function COLOR_BGR2GRAY. The original image was transformed to grayscale successfully.

After pressing key ‘G’ - convert to grayscale without using double loops

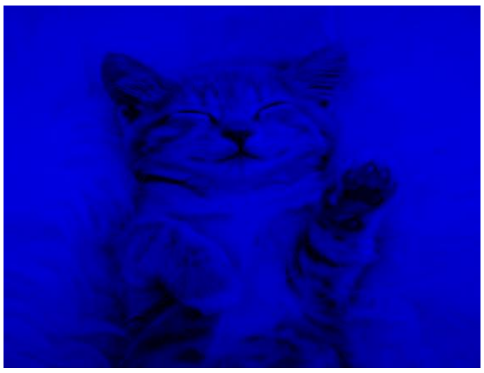


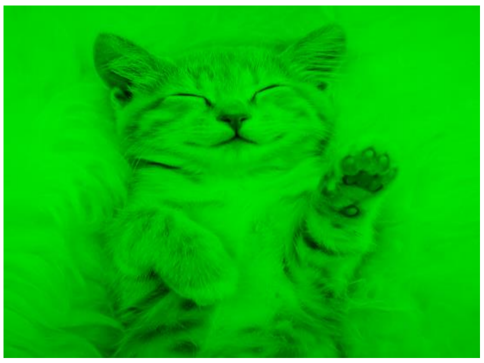
This is implemented using conversion function (gray = 0.2989 * r + 0.5870 * g + 0.1140 * b).

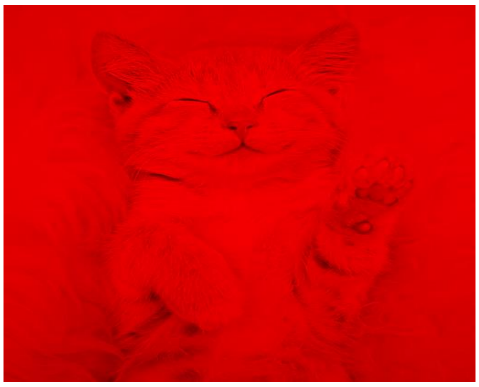
After pressing key 'T' - convert to grayscale using double loops



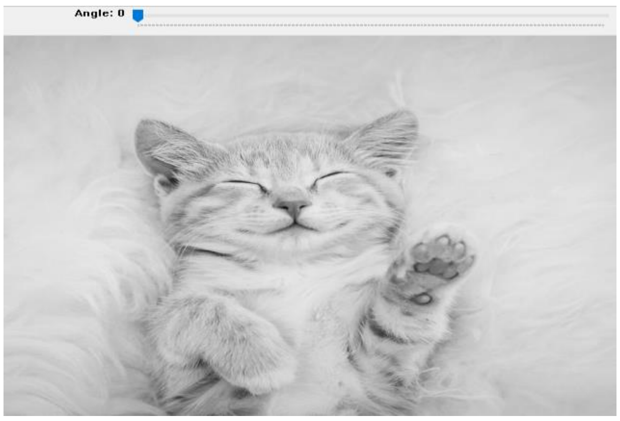
After pressing key 'c' – cycle through the color channels of image showing different channel every time the key is pressed.





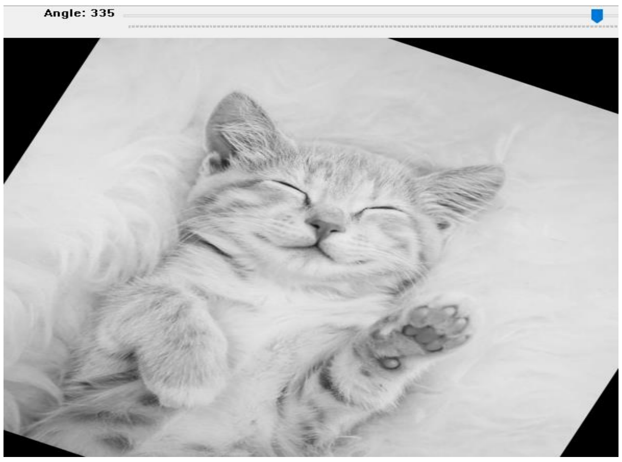


The image was successfully transformed to grayscale. After pressing key ‘r’ - converting to grayscale and then rotating the image



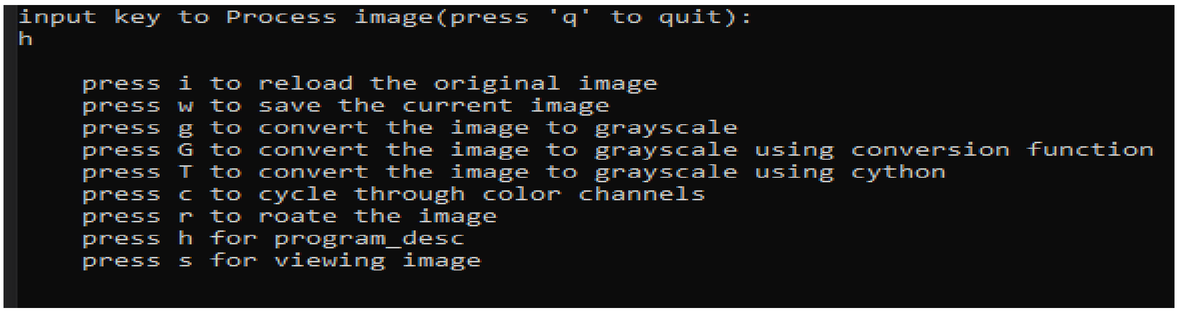
Here, we can see that the image was successfully converted to grayscale and the slider given is to rotate the image

After moving the slider –



The image was rotated at an angle of 335 degree using the slider.

Display a short description of the program



```
input key to Process image(press 'q' to quit):
h

    press i to reload the original image
    press w to save the current image
    press g to convert the image to grayscale
    press G to convert the image to grayscale using conversion function
    press T to convert the image to grayscale using cython
    press c to cycle through color channels
    press r to roate the image
    press h for program_desc
    press s for viewing image
```

So, from the above screenshot, we could see and evaluate that all the transformations associated with the respective input keys worked fine. All the functions worked correctly.

5. References: -

- https://towardsdatascience.com/use-cython-to-get-more-than-30x-speedup-on-your-python-code-f6cb337919b6
- https://docs.opencv.org/3.4/da/d6a/tutorial_trackbar.html