Assignment 1 Report

Course Code: CS 512
Course Name: Computer Vision
Course Instructor: Dr. Gady Agam

AS1
Prepared by: Khalid Saifullah
Student ID: A20423546
Date: 6th February, 2023

1. Problem Statement

In this programming assignment, we will be performing simple image manipulations with OpenCV. OpenCV is an open-source library of programming functions mainly for real-time computer vision.

2. Proposed Solution

The execution time is slow when double loops are used. This is solved by adding cython to the workflow. To test for generalizability of the code, it is run on many images.

3. Implementation Details

At first, a random image of choice was read using “cv.imread” function as a 3 channel color image regardless of what it is. Then, the color image is converted to grayscale image using OpenCV conversion function “cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)”. Both the original and the converted images are displayed (Fig 1). The execution time measured was 0.00235 seconds.

Fig 1

This rgb to grayscale conversion is done again but this time without using the built-in function and without any double loop. It is simply done using matrix multiplications. The execution time measured was 0.01937 seconds.

Next, the same thing is done just using double loop, without using built-in functions, where we scan all the pixels of the image. The execution time measured was 20.374 seconds. This was then accelerated using Cython. The new execution time measured was 0.35933 seconds.

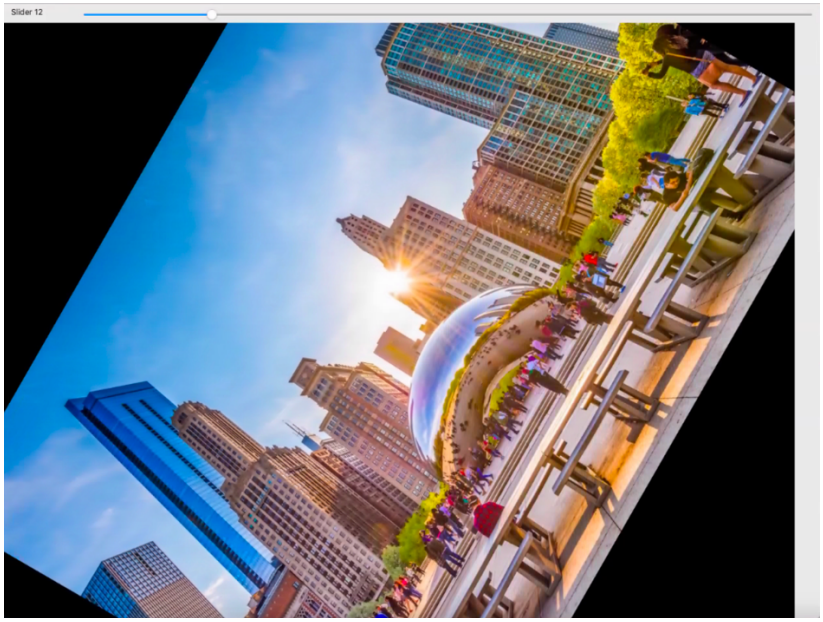
In the next part, the rgb channels are shown separately as 3 different images (Fig 2).

Fig 2

Then the image is rotated at an angle of choice between 0 and 360 degrees. It is implemented real-time using track bar. Note that the rotation is done using inverse map so that there are no holes in the rotated image. The 2 functions used are “cv2.getRotationMatrix2D” and “cv2.warpAffine”.

The next step was to replace the “cv2.getRotationMatrix2D” function by an user defined function that generates the rotation matrix.

Then, finally, both the functions “cv2.getRotationMatrix2D” and “cv2.warpAffine” were replaced by user defined functions. To carry this out, the output image was traversed, and the corresponding pixel of the input image was found using the inverse transformation matrix. In doing so, the holes became visible (Fig 3). The measured execution time was 2.749 seconds.



Note that, the size of the images was reduced to half to make the images look nicer when rotation takes place.

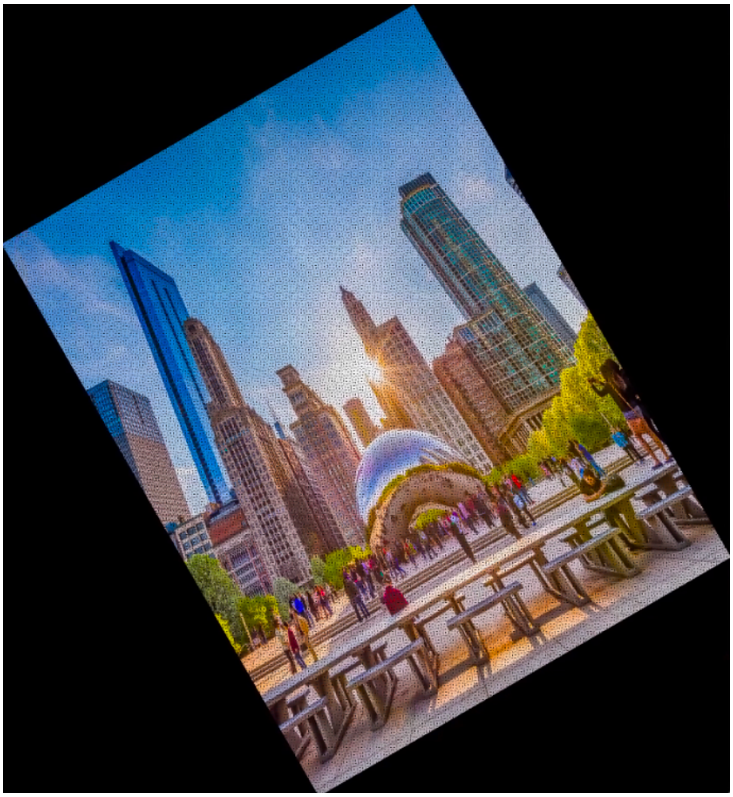


Fig 3

Overall, there were not much issues with the programming except that it was a little bit difficult to setup OpenCV and Cython in the computer.

4. Results and Discussion

Image Name	Size	Using OpenCV (seconds)	Using Matrix (seconds)	Using Double Loop (seconds)	Using Cython (seconds)
Chicago1	(1200, 1600, 3)	0.002	0.019	20.374	0.359
Chicago2	(1365, 2048, 3)	0.003	0.032	30.521	0.528
TulipGarden1	(800, 1280, 3)	0.002	0.006	11.299	0.200
CoxsBazar1	(720, 1280, 3)	0.001	0.008	10.067	0.174
CoxsBazar2	(1536, 2048, 3)	0.003	0.043	33.595	0.600
CoxsBazar3	(1080, 1920, 3)	0.002	0.019	22.595	0.393

Using OpenCV, the execution time is very small – almost instantaneous. But with matrix operation, it is around 10 times more time consuming. With double loops it is 10,000 more time expensive than the OpenCV functions. Incorporating Cython, this can be brought down to a point where is it is only around 200 times more time expensive than the OpenCV functions. We have tried 6 different images with varying sizes to test the execution time as can be seen from the table above.

5. References

1. https://cython.readthedocs.io/en/latest/src/quickstart/install.html
2. https://www.geeksforgeeks.org/how-to-install-opencv-4-on-macos/
3. https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/
4. https://stackoverflow.com/questions/14494101/using-other-keys-for-the-waitkey-function-of-opencv
5. https://answers.opencv.org/question/11959/program-termination/
6. https://www.geeksforgeeks.org/python-opencv-waitkey-function/
7. https://www.geeksforgeeks.org/python-opencv-cv2-imread-method/
8. https://www.geeksforgeeks.org/how-to-display-multiple-images-in-one-window-using-opencv-python/
9. https://stackoverflow.com/questions/60373943/how-to-determine-the-number-of-channels-in-image
10. https://www.geeksforgeeks.org/python-opencv-getrotationmatrix2d-function/
11. https://learnopencv.com/image-rotation-and-translation-using-opencv/
12. https://gautamnagrawal.medium.com/rotating-image-by-any-angle-shear-transformation-using-only-numpy-d28d16eb5076