

COMPUTER VISION

ISTANBUL TECHNICAL UNIVERSITY



**HOMEWORK-0 REPORT**

SEPTEMBER 22, 2017

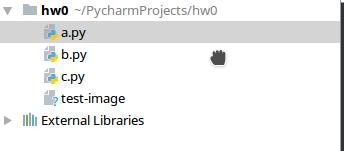
Name: EMRE KÖSE

ID: 150130037

Computer Vision Homework-0

Report

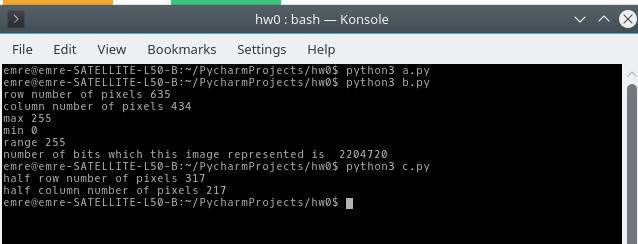
The homework has three parts and the image, which is colored-RGB form was used for 1st and 3rd ones. In the 2nd (b) part of homework grayscale version of image was used.



Firstly, python project was created and named as 'hw0' and three python files are created for every part of project and image named 'test-image.jpeg' was added to project. I runned these .py files from project but for homework I tried from konsole with these commands:

**python3 ‘filename’** and worked for me. I

also added to zip file the **test-image.jpeg** file which is used in these python files.

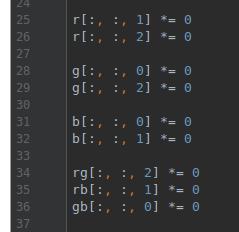


**Part a)**

In this part, the image was opened with **Image** library, which is in **PIL** and then it converted to array with the help of **numpy** library to do necessary operations with the below code:

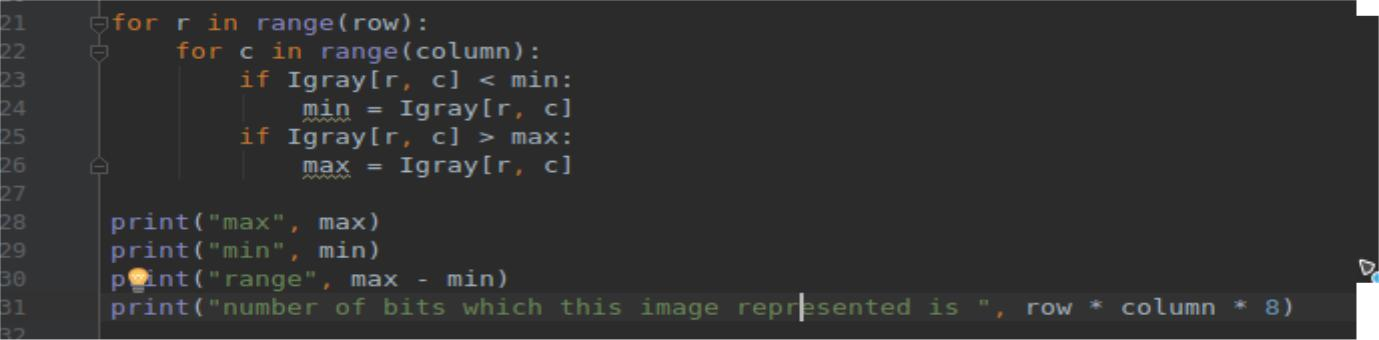
**Icolor = Image.open("test-image.jpeg")**

**r = np.array(Icolor)**

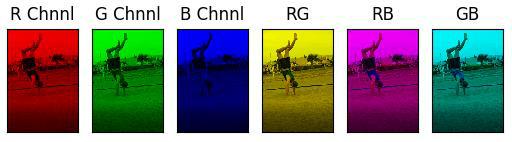


After that to display the only one band of this RGB image, other two bands' intensity values are assigned to 0 value. For example, in line 25 and 26, second and third matrices of RGB image, which are green and blue matrices' values were assigned to 0 value and the red channel was left only with its values. Then other arrays were arranged with same logic to get intended bands of array of image.

1



After these operations, arrays were transformed back to images to display with **imshow** function which is in **matplotlib** library. At last, all images displayed in one figure which is given below.



**Part b)**

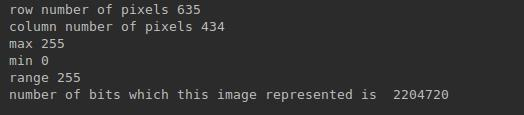
In this part, the original image was read with **cv2** library function **imread** and showed with **matplotlib** library function **imshow**, then image's size was taken by **numpy** library function whichname is **size** and assigned row and column values.



The Igray image was already read with formed in array and we know this is one band and has two indexes which are row number and column number of pixels. To find the range of the image minimum and maximum valued pixels were found by two for loops structure that is shown below:

Then range was calculated with max – min values and lastly number of bit which represents the image was found with taking products of row ,column and 8 (8 comes from grayscale image range value (0-255) and others to find pixel number) . For my image, results were shown also in code:

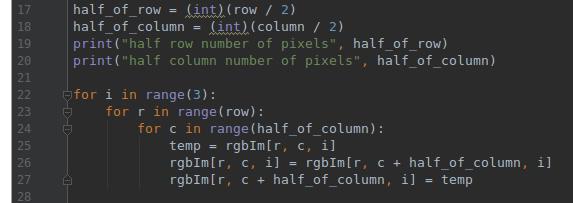
2



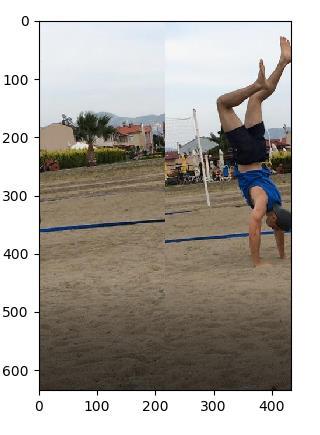
**Part c)**

At last part of the homework, flipping operations were handled with arranging the three bands RGB image. This was done with using three for loops because first two parameters were to obtain pixel and last one was the band index for R, G or B bands.

To explain of one example issue for flipping with code screenshot below:



Firstly, the half of the row and columns numbers calculated in integer values to play with pixel values. First for loop was used for bands from 0 to 2 indexes (0: R, 1: G, 2: B). Second and third loops were used for to find the value of pixels and then to change with necessary one.



At last, results were good, for this example, the image, which was flipped with horizontally obtained like right side:

3

Then as intended in homework, the image was flipped with vertically and at last was flipped the cross pixels and original image produced again.

To sum up, this homework was good for me to remind python again and to learn some basics of image processing operations.

4