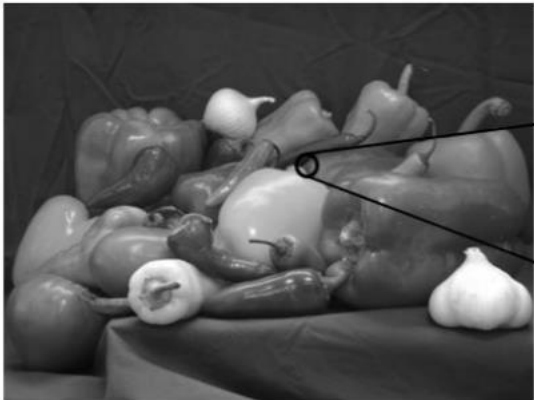


In this project, it is aimed to do some operations on a photograph using Python libraries. The project is a continuation of the intermediate project and aims to add additional modules to the intermediate project. To be able to do the project, we first need to understand how a picture is saved on the computer. In summary, a picture is kept in the form of a 3D matrix in the computer. Each element of the matrix corresponds to the color value of the picture and is called a pixel. The size of this matrix is the size of the picture x width x the depth of the picture. Depth is 3 for color pictures (red, green blue, ie RGB values), and 1 for black and white pictures. In other words, black and white images are stored in the computer in two dimensions. A sample black and white image (matrix size: size of the image x width of the image);



A sample black and white image (matrix size: size of the image x width of the image);



0.1216	0.1255	0.1059	0.1059	0.0941	0.0941
0.1176	0.1176	0.1137	0.1059	0.1059	0.1059
0.1020	0.1020	0.1059	0.1059	0.1059	0.1059
0.2902	0.1490	0.0980	0.0902	0.0941	0.0941
0.6235	0.5020	0.4196	0.2941	0.1608	0.0863
0.6941	0.6392	0.6431	0.6510	0.5294	0.3541
0.7451	0.7255	0.6667	0.6353	0.6510	0.6353
0.6863	0.6824	0.7137	0.6863	0.6353	0.6353
0.6784	0.6784	0.7373	0.7373	0.7020	0.7020
0.6980	0.7176	0.7176	0.7176	0.7098	0.7098
0.7255	0.7216	0.7216	0.7216	0.7216	0.7216

A sample colourful image (matrix size: size of the image x width of the image);



0.1333	0.1451			
0.1529	0.1882	0.2118	0.2314	
0.1647	0.2078	0.2471	0.2588	
0.1529	0.4706	0.4784	0.2706	0.2
0.4863	0.4980	0.5176	0.5333	0.26
0.5020	0.5127	0.5294	0.5412	0.26
0.4980	0.4039	0.4118	0.5569	0.172
0.4157	0.4431	0.4431	0.4511	0.5
0.4392	0.4471	0.4667	0.4824	0.525
0.4471	0.4549	0.4510	0.4549	0.4745
0.4471	0.4549	0.4627	0.4863	0.5137
0.5216	0.5882	0.6627	0.7333	0.7843
0.8314	0.8863	0.9098	0.9373	0.9412
0.9451	0.9647	0.9647	0.9765	0.9686
0.9922	1.0000	1.0000	1.0000	1.0000
1.0000	1.0000	1.0000	1.0000	1.0000
1.0000	1.0000	1.0000	1.0000	1.0000

The photo editor will work in a loop. The expected program flow is as follows.

1- Upload a picture

2- Turn the picture into black and white

3- kirpin the official

4- Apply a filter (convolution)

5- Freeze the picture horizontally

Enter your choice: 1

Enter the name of the image file: istinye.jpg

istinye.jpg image uploaded.

1- Upload a picture

2- Turn the picture into black and white

3- Crop the picture

4- Apply a filter (convolution)

5- Freeze the picture horizontally

Enter your choice: 4 // A square filter will be requested from the user and the filter will be applied on the image.

If no image is uploaded, when user selects other options

the program should give a warning. If there is an image loaded, you can do the operation on the program and

it should show the result as a picture. will be displayed.



For example, if we assume that the Usable image is loaded into memory, this image

In order of the actions to be taken on it;

2. Convert the picture to black and white: During this process, the program loaded in memory will turn the picture (3-dimensional) into two-dimensional. The action for this is the RGB doing the following between the returns and translating the result into a new 2-dimensional matrix. is to write. It will then be shown that this matrix will be used.

$$0.2989 * R + 0.5870 * G + 0.1140 * B$$



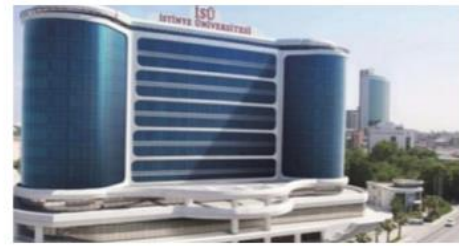
3. Crochet the picture: For this operation, the program takes the area of the picture from the user to crop 2.

will select one as the (x, y) coordinate. The first coordinate is the upper left corner of the place to be cropped,

The second coordinate will correspond to the lower right corner. Program trimming is possible

It should also check whether it is present (for example, the lower right corner is outside the border of the picture).

it should show the result as a picture. will be displayed.

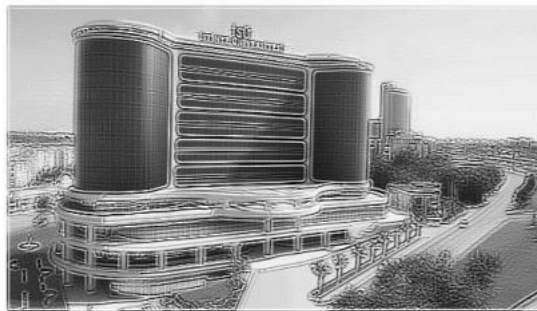


4. Apply a filter (convolution): During this process, a square filter entered by the user is black.

The white image will be applied and the processed image will be printed on the screen. If user 4. Process

If it chooses, the program should ask the user for the filter size and the values of the filter. The sample should be shown as a filter. will be displayed. sample filter result:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -7 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



5. Freeze the picture horizontally: The color picture will be frozen horizontally during this process.

. sample filter result:



NOTE 1: After each process, the transaction result must be written in the same folder. For example

If the uploaded image name is istinye.jpg, the result of the operations numbered 2, 3, 4, 5, respectively

**istinye_sb.jpg, istinye_kirpilmis.jpg,
istinye_filtre.jpg, istinye_yatay.jpg
must be.**

NOTE 3: You can use libraries such as pillow, numpy. But each desired

**The operation will be explicitly coded as a function. Ready to do these operations
the function cannot be used.**

The intermediate project mentioned is below The codes I sent for the intermediate project are also included.



Simple filtering on a photo using some Python libraries in this project

operations are intended to be performed. First, how to use a picture on the computer we need to understand that it was recorded. In summary, a 3D matrix on a picture computer it is kept in shape. Each element of the Matrix corresponds to the color value of the image, and the Pixel

is referred to as. The size of this matrix is the length of the image x the width of the image x the depth of the image

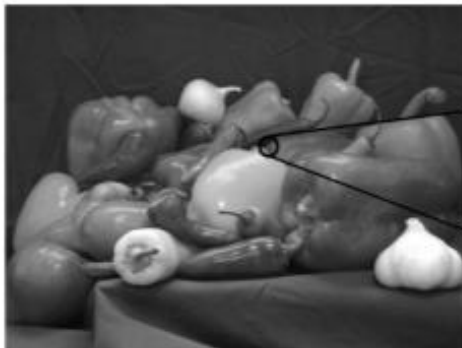
references 3 for depth color images (red, green blue, i.e. RGB values), black

1 for white pictures. So black and white images are two-dimensional on a computer

he hides.

An example is a black and white image (matrix size: image length x image width);;

Örnek bir siyah beyaz resim (matris boyutu: resmin boyu x resmin eni);



0.1216	0.1255
0.1176	0.1137
0.1020	0.1059
0.1490	0.0902
0.5020	0.2941
0.6350	0.6510
0.7255	0.6353
0.6824	0.6863
0.6784	0.7373
0.6980	0.7176
0.7255	0.7216

Örnek bir renkli resim (matris boyutu: resmin boyu x resmin eni x 3);



0.1333	0.1451
0.1529	0.1892
0.1647	0.2471
0.4706	0.4784
0.4863	0.5176
0.5020	0.5412
0.4039	0.4118
0.4157	0.4431
0.4392	0.4471
0.4549	0.4510
0.4545	0.4627
0.5216	0.5882
0.6314	0.9098
0.9647	0.9765
0.9922	1.0000
1.0000	1.0000
1.0000	1.0000



For example, if we assume that the user has uploaded the above image to memory, this image

The procedures to be done on it are as follows;

1-) Converting the picture to black and white: During this process, the program loaded into memory

will convert the colored picture (3 dimensional) to two dimensional What to do for this?

Do the following operation between RGB values and add the result to a new 2-dimensional matrix.

is to write. Then this matrix will be displayed on the screen

$$0.2989 * R + 0.5870 * G + 0.1140 * B$$



2-) Inverting the picture: For this operation, the program will reverse the picture in the memory horizontally.

will dial / frozen. The picture you need to do for this process (3-dimensional matrix)

It is to make a replacement between the lines. For example, reverse the first line of the picture the next line should be the last line of the picture.



The photo editor will work in a loop. The expected program flow is as follows.

1- Upload a picture

2- Turn the picture into black and white

3- Freeze the picture horizontally

Enter your choice: 1

Enter the name of the image file: istinye.jpg

istinye.jpg image uploaded.

1- Upload a picture

2- Turn the picture into black and white

3- Freeze the picture horizontally

Enter your choice: 2 // The screen will show the picture in black and white.

If no images are uploaded, when user selects Option 2 or 3

the program should give a warning. If there is an image loaded, you can do the operation on the program and

it should show the result as a picture.

NOTE 1: What you need to do in this project is 2 or 3 stored in computer memory.

It is to do operations on dimensional matrices. Other than that, your picture

Sample codes and solutions for issues such as loading, projecting on the screen

will be provided. Necessary explanations will be made in our lessons.

NOTE 2: A pillow library is used to do simple image operations on Python.

can be used. The picture can be read via the pillow library or

I can show In addition, the picture read by this library is transferred to the numpy data type.

By turning (matrix), the above operations can easily be done on this matrix.

The codes I submitted for the intermediate Project

```
from PIL import Image
from glob import glob
import numpy as np
files=glob('.png')
files.extend(glob('.jpg'))
files.extend(glob('*.*jpeg'))

processes="""
1-) Upload the picture
2-) Turn the picture into black and white
3-) Rotate the picture horizontally
4-) Exit

"""

def rotate(img):
try:
img2=np.array(img)
for i in range(4):
Image.fromarray(np.rot90(img2,i)).show()

except:
print("NO PICTURE UPLOADED!")
def black_and_white(img):
gray = img.convert('L')

bw = np.asarray(gray).copy()

bw[bw < 128] = 0 # Black
bw[bw >= 128] = 255 # White

imfile = Image.fromarray(bw)
imfile.show()

while True:
print(processes)
choice=input("Enter your choice:")
if(choice == "1"):
try:
image_name=input("Enter the name of the image file:")
img=Image.open(image_name)
print("{} resmi yüklendi".format(image_name))
except:
print("NO PICTURE UPLOADED!")
print("\n")
```

```
print("Pictures you can choose:")
print("\n".join(files))
elif(choice == "2"):
try:
black_and_white(img)
except:
print("NO PICTURE UPLOADED!")
elif(choice == "3"):
rotate(img)
elif(choice == "4"):
break

else:
print("WRONG CHOOSE!")
```