German University in Cairo

Media Engineering and Technology

Assoc. Prof. Rimon Elias

DMET 901: Computer Vision

Assignment 1, Deadline: 27th of July 2023

In this assignment, the main aim is to exercise practically the concepts of arithmetic and geometric transformations. This target is to be achieved by adopting a series of exercises described below. In each exercise, it is required to visualize the effect of each operation, analyze its effect on different image areas, and provide a brief discussion demonstrating your findings. This assignment is to be implemented over Python, and you will be provided with a starter code, which is discussed by the end of the document.

Required Functions:

1. mergeImages:
    - Input: two images of the same size (I', I''), and a number (s).
    - Output: one images: $\left(I = I' + \frac{I''}{s}\right)$
    - Description: the main aim is to decrease the color range of image I'', that after adding to I', I & I' would be nearly the same visually.
2. retrieveImage:
    - Input: two images of the same size (I, I'), and a number (s).
    - Output: one image ($I'' = s * (I - I')$).
    - Description: the main aim is to retrieve I'' back.
3. scaling42:
    - Input: one image.
    - Output: one image after performing the transformations.
    - Description: you are asked to perform the transformations stated in Lecture 4, Slide 42. Instead of working with a 512x512 image (centering it at 256x256), you will use a 256x256 image (centering it at 128x128), you are asked to provide a feedback about the change of location of a certain feature of your choice.
4. ImagesXOR:
    - Input: two images of the same size (I, I').
    - Output: one image ($I'' = I\ XOR\ I'$).
    - Description: you are asked to perform the XOR operation on the input images (I, I') on the ***grayscale*** level as discussed in class.

Assignment Requirements:

1. This assignment is to be implemented using Python over Google Colab.
2. Each of the previously mentioned functions are to be implemented as discussed.
3. You are allowed to use predefined functions in the openCv library.

4. The four test scenarios (on Functions 1 and 2) are to be performed (using image Input1.PNG and Input2.PNG), with displaying all output images. The four scenarios are: $s = \{10, 50, 100, 200\}$.
5. Function 3 is to be tested using either of the input images.
6. Function 4 is to be tested twice, the first test is performed using image Input1.jpg and Input2.jpg (producing output.jpg for instance), the second test is performed using the output of the previos test (output.PNG) and Input2.PNG.
7. Input images are loaded and operated on in the grayscale color range.
8. A brief discussion of each output demonstrating your analysis (visually) and findings performing these tests. This discussion is to be provided as a comment on your code or as a text cell.
9. Assignment is to be implemented individually, cheating detection will be applied.

Starter Code:

1. Imports:
   - `import cv2`: to import the library of openCv in order to use its pre-defined functions.
   - `from matplotlib import pyplot as plt`: used for the image display function demonstrated later.
2. Image Display:

```
def displayImages(imgs):
  r = len(imgs)//2
  if len(imgs) % 2 != 0:
    r+=1
  plt.subplots(nrows=r, ncols=2, figsize=(20, 20))
  for i in range (0, len(imgs)):
    plt.subplot(r,2,i+1)
    plt.imshow(imgs[i], cmap="gray")
```

This function is used to display a set of images in a grid form of dimensions $(\frac{\#imgs}{2}, 2)$. For each of the required tests, each generated output image is to be generated then added to a list, which this method is to be invoked on.

Input1.jpg

Input2.jpg