Project 1

CSE 473/573 (Fall 2019)

Due Date: Sep. 25, (Wed.) 3 P.M., 2019

1 Image Filtering [5 points]

The goal of this task is to experiment with image filtering, and familiarize you with tricks, e.g., padding, commonly used by computer vision practitioners. Specifically, the task is to perform image smoothing and image sharpening using a low-pass filter and a high-pass filter shown in Table 1. You are required to implement all the functions that are labelled with “ `#TODO`” in “`task1.py`”. In “`task1.py`”, we not only provide hints to you, but also provide utility functions that could be used as building blocks for you to complete this task. In order to make it easier for you to validate the correctness of your code, we provide two examples for image smoothing and image sharpening respectively. The examples are store in “`./results/`” and are named as “`low-pass.jpg`” and “`high-pass.jpg`”. Your code should be able to generate images that are identical to those stored in “`./results/`” (Note that images in “`./results/`” are provide to you for reference only. At test time, a different image will be used as input and the correct output images will be different from those already stored in “`./results/`”). Detailed information and image data can be found at our course website on Piazza.

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

-1/9	-1/9	-1/9
-1/9	17/9	-1/9
-1/9	-1/9	-1/9

Table 1: The low-pass filter (left) and the high-pass filter (right) used in this project.

2 Template Matching [5 points]

The goal of this task is to experiment with template matching algorithms, i.e., normalized cross correlation (NCC). Specifically, the task is to locate a given image patch in a large image. You are required to implement a function named “match” in “`task2.py`”, which detects a patch in an image. The function “match” takes an image and a patch as inputs and returns the coordinates that the patch appears. Detailed information and image data can be found at our course website on Piazza.

3 Guidelines

Your code will be graded **automatically** using a program which evaluates the differences between the outputs of your code and the correct outputs. For graduate students whose code raise “`RuntimeError`”, your grades will be 0. For undergraduate students whose code raise “`RuntimeError`”, we will manually grade your projects.

- Compress the three python files, i.e., “`task1.py`”, “`task2.py`” and “`utils.py`” into a **zip** file, name it as “`UBID.zip`” (replace “`UBID`” with your eight-digit UBID, e.g., 51399256) and upload it to **UBLearns** before the due date. The zip file you upload should **not** contain files other than the three aforementioned python files.
- Do **not** modify the code provided to you.
- Do **not** use any API provided by opencv (cv2) and numpy (np) in your code (except “`np.sqrt()`”).
- Do **not** import any library (function, module, etc.).
- Ask for the TA’s approval if you would like to use a programming language other than Python, C, C++, C# and Java.