CSE 398/498 - Assignment 1

Image Enhancement

Logistics

The submission should be made on Course site in the form of a zip file named as **<your_full_name>_cv_21.zip** by **September 10, 2021 5:00pm (EST).**

Contents of the compressed folder should be organized in the following manner:

- **data/** - folder containing input images
- **output/** - folder containing output images
- **src/** - containing your python scripts, one file for each task
- **readme.md/readme.txt** - containing verbose description of the steps required to run your code and other observations/comments you may have for your submission. Treat this as a short report. Include any links you may have referred to in this document.
- **environment.yml** - exported conda environment file. Contains information about libraries and their versions.

Credit: 5% of course total

Due date: September 10, 2021, 5:00 pm

Pre-assignment

Prepare your laptop. Virtual environments are a great way to ensure stability of your code execution and also improve its reproducibility. I suggest everyone to use the **Anaconda** package manager for your projects and assignments. If you have used it before, you already know. If you haven't used it before, it will make installations easier and project management better. Also, a lot of other people's code uses it so it's good to learn.

Quick tutorial for conda: https://conda.io/projects/conda/en/latest/user-guide/getting-started.html

Please contact the instructor if you need help with conda ASAP

General steps of image processing (with links for reference) -

1. Read the image file - https://www.geeksforgeeks.org/reading-images-in-python/

2. Store it as a matrix - https://www.oreilly.com/library/view/programming-computer-vision/9781449341916/ch01.html
3. Apply operations to the matrix - https://www.pyimagesearch.com/2016/07/25/convolutions-with-opencv-and-python/
4. Display the matrix to check the output of your operation - Using Matplotlib and OpenCV (https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html), Using Pillow (https://pythonexamples.org/python-pillow-show-display-image/)
5. Save the output as an image file - Using Matplotlib (https://pythonspot.com/matplotlib-save-figure-to-image-file/), Using OpenCV (https://www.geeksforgeeks.org/python-opencv-cv2-imwrite-method/), Using Pillow (https://www.geeksforgeeks.org/python-pil-image-save-method/)

The basic image reading, drawing and writing can be done using 3 libraries -

- OpenCV
- Pillow
- Matplotlib

NOTE: For this assignment, it is suggested to use OpenCV for image loading, showing and saving but all the following tasks have to be completed without use of any specialized built-in functions. You have to implement the matrix operations for image filtering for both the tasks. You may reuse the function you have created for one task for the other one in order to reduce redundant work.

Task 1 (2 points): Image Sharpening

A sharpening operation enhances the major edges in an image. It can be implemented using the following steps:

Consider input image is I

Step 1: $I - G(I) = D(I)$, here $G(I)$ is the blurred image using a Gaussian kernel/filter of your choice and $D(I)$ is the residual image containing edge details.

Step 2: $I + D(I) = I_{enhanced}$ adding the details back to image reinforces the edge information creating a sharper image.

$I_{enhanced}$ is the final sharpened image and the above method is called '*unsharp masking*'.

Display image after each step along with the original image and the final output. Save the output images in the output folder.

***CSE 498 - For graduate students: Identify a filter that can sharpen the image in one step and explain why it works. No code required. (-0.5 if not attempted or wrong)

Task 2 (3 points): Image Denoising

Background - Real world signals usually contain departures from the ideal signal that would be produced by our model of the signal production process. Such departures are referred to as *noise*. Noise arises as a result of processes going on in the production and capture of the real signal. It is not part of the ideal signal and may be caused by a wide range of sources, *e.g.* variations in the detector sensitivity, environmental variations, the discrete nature of radiation, transmission or quantization errors, *etc.* It is also possible to treat irrelevant scene details as if they are image noise (*e.g.* surface reflectance textures). The characteristics of noise depend on its source, as does the operator which best reduces its effects.

Noise can generally be grouped into two classes:

- independent noise.
- noise which is dependent on the image data.

Image independent noise can often be described by an additive noise model, where the recorded image *f(i,j)* is the sum of the *true* image *s(i,j)* and the noise *n(i,j)*:

$$f(i,j) = s(i,j) + n(i,j)$$

In many cases, additive noise is evenly distributed over the frequency domain (*i.e. white noise*), whereas an image contains mostly low frequency information. Hence, the noise is dominant for high frequencies and its effects can be reduced using some kind of lowpass filter. This can be done either with a frequency filter or with a spatial filter.

In the second case of *data-dependent noise* (*e.g.* arising when monochromatic radiation is scattered from a surface whose roughness is of the order of a wavelength, causing wave interference which results in image *speckle*), it is possible to model noise with a multiplicative, or non-linear, model. These models are mathematically more complicated; hence, if possible, the noise is assumed to be data independent.

Detector Noise

One kind of noise which occurs in all recorded images to a certain extent is *detector noise*. This kind of noise is due to the discrete nature of radiation, *i.e.* the fact that each imaging system is recording an image by counting photons. Allowing some assumptions (which are valid for many applications) this noise can be modeled with an independent, additive model, where the noise *n(i,j)* has a zero-mean Gaussian distribution described by its standard deviation (σ), or variance. This means that each pixel in the noisy image is the sum of the true pixel value and a random, Gaussian distributed noise value.

Salt and Pepper Noise

Another common form of noise is *data drop-out* noise (commonly referred to as *intensity spikes*, *speckle* or *salt and pepper noise*). Here, the noise is caused by errors in the data transmission. The corrupted pixels are either set to the maximum value (which looks like snow in the image) or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a `salt and pepper' like appearance. Unaffected pixels always remain unchanged. The noise is usually quantified by the percentage of pixels which are corrupted.

Implement the following three denoising techniques using image filtering:

1. Mean Filtering - Apply an averaging kernel on the given input image.
2. Gaussian Smoothing - Apply a Gaussian blur kernel on the given input image.
3. Median Filtering - Apply median filtering to the given image.

Display the input and output images for each. Save the output images in the output folder.

Analysis - Which method works best for each image and why?

Rubric

Task 1 : 1 point for correct implementation of each step and displaying intermediate output. A penalty of 0.5 for graduate students if the extra component is missing or incorrectly answered.

Task 2 : 1 point for correct implementation of each of denoising operations and displaying a denoised image.

Deadline Extensions and Late Submissions:

- First deadline extension request will be granted with a 3-day extension automatically and you won't be penalized. ***Applicable to Assignments and Projects***
- For subsequent late submissions, you will lose 10% for each day late for the programming projects. This means anytime within the first 24 hours after the due date count as 1 full day, up to 48 hours is two and 72 for the third late day. Beyond that, your submission will not be graded. ***Applicable to only Assignments***

Please clarify this with the instructor if the policy is not clear before you submit late.

Statement on Academic Integrity

University -We, the Lehigh University Student Senate, as the standing representative body of all undergraduates, reaffirm the duty and obligation of students to meet and uphold the highest principles and values of personal, moral and ethical conduct. As partners in our educational community, both students and faculty share the responsibility for promoting and helping to ensure an environment of academic integrity. As such, each student is expected to complete all academic course work in accordance to the standards set forth by the faculty and in compliance with the University's Code of Conduct.

Course - The work you do in this course must be your own. This means that you must be aware when you are building on someone else's ideas—including the ideas of your classmates, your professor, and the authors you read—and explicitly acknowledge when you are doing so. Feel free to build on, react to, criticize, and analyze the ideas of others but, when you do, make it known whose ideas you are working with. If you ever have questions about drawing the line between others' work and your own, ask me and I will give you clear guidance or you may visit Lehigh Library's 'Proper Use of Information' page at http://libraryguides.lehigh.edu/plagiarism

Grade Specific - Zero assigned to the Quiz/Assignment for first offense and the student will Fail the class on second offense.

For assignments, you can discuss with peers but the code should be your own. For quizzes, no consultation with a living or non-living entity is allowed.

University COVID Policy:

To meet the challenge of teaching and learning during the COVID-19 pandemic, Lehigh instructors and students will be adopting new forms of instruction and interaction; following new guidelines around classroom behaviors; enhancing communications; and doing our best to be patient, flexible, and accommodating with each other. In remote synchronous meetings, students are expected to attend just as they would any other Lehigh class. Zoom classes work best when all students come to class ready to participate and follow the instructor's guidelines regarding use of web-cameras. You may be asked to turn your camera on during active learning sessions in Zoom. If you have a strong preference not to do so, please contact your instructor to let them know. Students should respect the in-classroom privacy of their instructors and fellow students by not taking screenshots or recording class sessions. Some instructors will record Zoom sessions; however, any recorded live sessions will be shared only with students in the class and will be deleted at the end of the semester.

In our physical classrooms, Lehigh has established a policy requiring everyone to wear face coverings when in public spaces inside buildings on our campus and to maintain social distance. This policy applies to our physical classroom. Thank you in advance for following this rule. Students who do not wear a face covering during in-class meetings will be reminded to put their face covering on. If they do not do so, they will be asked once again to do so or leave the classroom.