ECE 4250
Assignment 4

Due April 27nd 2020, 11:59 pm

General Instructions

The assignment submissions are via Canvas. There are two parts: a problem set and a coding (programming) component. Your answers to the problems can be written up any way you want (e.g., using Latex or scanning your handwritten answers). As long as we can read it, we will grade it. For the programming part, you will code in Python 3, using Jupyter Notebook. Make sure that all cell outputs are clearly visible before saving the notebook. Thus, we recommend that you re-run your code after you are done editing. You should then print this file into a pdf. For the coding component, please submit both the ".pdf" and ".ipynb" files. Please zip up all your work into a single file and submit that.

Any acknowledgments of collaboration, answers to questions, comments to code or other notes should all be included in your problem set answer sheet and/or Jupyter Notebook. General guidelines regarding assignments apply. In other words, you can collaborate at the ideation/brainstorming stage, but whatever you submit should represent your own individual work. You are not allowed to copy/paste others' code/answers or work on same code with someone else. You are not allowed to use functions outside of the Python Standard Library, unless specified otherwise or you have received permission to do so. Your code will be graded on readability, executability, and accuracy. You are strongly advised to use detailed commenting that explains the algorithmic steps in your code. Explain every function and loop. Use piazza to ask questions. Take advantage of TA and Instructor Office Hours to seek help.

Problem Set

[60 pts, 20 per question]

Question 1.

- Two images $x_1(m,n)$ and $x_2(m,n)$ have unnormalized histograms h_1 and h_2. Give a condition under which one can determine these image histograms from the unnormalized histogram of $x_1 + x_2$. Describe the procedure you would implement to compute the individual histograms.
- Give a single integer-valued intensity transformation for spreading the intensities of an input image so lowest intensity is 0 and highest intensity is $L-1$.

Question 2. Consider following 2D image:

$$\cos(2\pi\mu_o m/M + 2\pi\nu_0 n/N),$$

where $m, m_0 \in [M]$ and $n, n_0 \in [N]$. Derive the DFT.

Question 3. Consider a 3×3 spatial mask that averages the four closest neighbors of a pixel (m, n), but excludes the pixel itself from the average.

- Express the filter as a 2D array (in spatial coordinates)
- Derive the 2D discrete Fourier transform of this filter
- Is this a high or low pass filter? Explain your answer.

Programming Questions

1 Histogram Equalization

[25 pts]

- Implement the basic histogram equalization algorithm. Recall that this method applies following intensity transformation:

$$T(x) = (L-1)\int_0^x p_x(w)dw,$$

  where p_x is the normalized histogram of the image, and the intensity values are assumed to be in $[L]$. Apply this algorithm to X.png and visualize result.

- Implement an improved (exact) histogram equalization algorithm based on the paper discussed in lecture (Coltuc, Dinu, Philippe Bolon, and J-M. Chassery. "Exact histogram specification." IEEE Transactions on Image Processing 15.5, 2016). Apply this algorithm to X.png and compare the result to previous one.

2 Spatial Transformations

[25 pts]

For the following, the output image should be the same size as the input image. Outside of boundary values can default to zero. You are allowed to use a readily available interpolation function. Note that the "transformed image" might not have the field view to show the entire original scene.

- Write a function called *Rotate2D* that accepts an input image and a rotation angle (in degrees) and outputs an image that is the input rotated around the center pixel of the image.
- Read the image X.png. Rotate it by 45 degrees, save this as RotatedX.png, and visualize this image.
- Write a function called *Translate2D* that accepts an input image and a translation (shift, displacement) vector (in 2D, pixel units); and outputs an image that is the input image shifted/translated by the provided displacement vector.
- Shift X by $[30, -70]$ pixels, save this as ShiftedX.png, and visualize this image.