Homework 1

ECE 253
Digital Image Processing

October 4, 2023

Make sure you follow these instructions carefully during submission:

- Homework 1 is due by 11:59 PM, October 22, 2023.
- All problems are to be solved using Python.
- You should avoid using loops in your code unless you are explicitly permitted to do so.
- Submit your homework electronically by following the two steps listed below:

1. Upload a pdf file with your write-up on Gradescope. This should include your answers to each question and **relevant code snippet**. Make sure the report mentions your full name and PID. You **must** use the Gradescope page selection tool to indicate the relevant pages of your report for each problem; failure to do so may result in no points. Finally, carefully read and include the following sentences at the top of your report:

   Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.

   By including this in my report, I agree to abide by the Academic Integrity Policy mentioned above.

   Reports missing any of the above elements will receive no credit.
2. Upload a zip file with all of your scripts and files on Gradescope. Name this file: **ECE_253_hw1_lastname_studentid.zip**. This should include all files necessary to run your code out of the box.

Problem 1. Basics (3 points)

$$\text{Input } A = \begin{bmatrix} 3 & 9 & 5 & 1 \\ 4 & 25 & 4 & 3 \\ 63 & 13 & 23 & 9 \\ 6 & 32 & 77 & 0 \\ 12 & 8 & 6 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

(i) Point-wise multiply A with B and set it to C.

(ii) Calculate the inner product of the 2nd and 3rd row of C.

(iii) Find the minimum and maximum values and their corresponding row and column indices in matrix C. If there are multiple min/max values, you must list all their indices.

In your report include all the outputs generated by your code.

Problem 2. Preliminary image manipulation (5 points)

(i) Download any color image from the Internet or use one of the given images. Read this image and call it **A**.

(ii) Transform the color image to gray-scale. Verify the values are between 0 and 255. If not, please normalize your image from 0 to 255. Call this image **B**.

(iii) Add 15 to each value of image B. Set all pixel values greater than 255 to 255. Call this image **C**.

(iv) Flip image B along both the horizontal and vertical axis. Call this image **D**.

(v) Calculate the median of all values in image B. Next, threshold image B by the median value you just calculated i.e. set all values greater than median to 0 and set all values less than or equal to the median to 1. Name this binary image **E**.

Include all images A-E in your report. Try to fit at least 4 images in a page.

Problem 3. Noise Cleaning (6 points) Load the images sungod, noisy1, and noisy2. For each step (ii)-(iv), include the resulting filtered images.

(i) Which image has Gaussian noise, and which image has Salt & Pepper noise?

(ii) Use a 1x3 median filter to clean both images. What are the resulting MSEs?

(iii) Use a 3x3 median filter to clean both images. What are the resulting MSEs?

(iv) Use a 3x3 mean filter to clean both images. What are the resulting MSEs?

(v) Discuss your results; which filter(s) worked best for which images, and why?

Problem 4. Binary Morphology (10 points)

In this problem, you will be separating out various objects in binary images using morphology, followed by connected component analysis to gather more information about objects of interest.

(i) For the binary image *circles_lines.jpg*, your aim is to separate out the circles in the image, and calculate certain attributes corresponding to these circles.

- The first step is to remove the lines from the image. This can be achieved by performing the *opening* operation with an appropriate structuring element. You may find the following functions to be of use.
  Python: *skimage.morphology.opening*()
  Feel free to try other functions as well for this question.
- Once we have a binary image with just the circles, the individual regions need to be labeled to represent distinct objects in the image i.e. *connected component labeling*. This can be done in
  Python using *scipy.ndimage.measurements.label*().
  Feel free to try other functions as well for this question.
- We now have access to individual regions in the image, each of which can be analyzed separately. For each labeled circular region, calculate its *centroid* and *area*. Tabulate these values for each connected component. You may use loops for this part of the problem.
  Note : This step must be performed entirely by your code without making use of Python's inbuilt functions (like *regionprops()*) for connected component analysis. You can still use simple utility functions like *sum()*, *mean()*, *find()* etc.

(ii) In this part, we are interested in performing similar operations on the binary image *lines.jpg*. Your aim now is to separate out the vertical lines from the horizontal ones in the image, and then calculate certain attributes corresponding to these vertical lines.

- In a similar manner to part (i) above, use the opening operation with a suitable structuring element to remove the horizontal lines in the image. (Remove the diagonal line as well).
- Next, perform connected component labeling on the resulting image to identify and label each unique vertical line.
- Finally, for each labeled region, calculate the length (the longer dimension) and the centroid. Tabulate these values for each connected component. You may use loops for this part of the problem.
  Note : This step must be performed entirely by your code without making use of Python's inbuilt functions (like *regionprops()*) for connected component analysis. You can still use simple utility functions like *sum()*, *mean()*, *find()* etc.

Things to turn in:

- Part (i) & (ii) : The original image, the image after *opening*, the image after *connected component labeling* (plot with colorbar), and a table with the desired values for each component.

- The structuring element *type* and *size* that was used to perform the opening operation (for both parts).
- Code for the problem.

Problem 5. PCB Inspection (6 points)

The image pcb contains an image of a PCB (Printed Circuit Board) that needs to be inspected for the number of holes (in this example, 4) and their horizontal diameters. Using binary morphology and other operations, design a system which filters the image, leaving only the holes. Describe your solution, and show all intermediate filtered images as well as the resulting image. Finally, using the resulting image, write a function which returns the center and radius of each hole and provide the results.

Problem 6. Histograms (6 points)

Histograms[1] are a great statistical tool to analyze the distribution of intensity values in an image. In this problem, you have to write a Python function with the following specifications:

- Write a function named *compute_norm_rgb_histogram* that computes the RGB color histogram.
- Use 32 bins for each color channel (i.e. Red, Green and Blue), spaced equally between 0 and 255. This should result in a 32-length vector for each channel.
- One input (RGB/color image) and one output (1 x 96 vector).
- Normalize the histogram of each color channel (so that it sums to 1), then concatenate the three histograms together (in the order R, G, B) to make a combined histogram of length 3x32 = 96.
- Do not use Python inbuilt histogram function. You may use loops if necessary.
- Call the function and plot the final combined, normalized histogram for the image *geisel.jpg*. Make sure the plot is labeled correctly. Show your plot in the report.

Problem 7. Chroma Keying (6 points)

Chroma keying is used for extracting the foreground from images, with the background typically being a green screen. In this problem, you have been provided 2 images: *travolta.jpg* and *dog.jpg*. Write a script to extract the foreground from either image and overlay the foreground on a different background of your choice. In your report you should include for each of the images:

(i) A binary image showing the foreground mask, i.e., all foreground pixels set to 1 and all background pixels set to 0.

(ii) An image with the background pixels set to 0 and the foreground pixels set to their original values.

[1] You can read more about Histograms here.

(iii) An image with the foreground overlayed on a background of your choice.

Problem 8. Upsampling and downsampling (10 points)

Sampling is a technique that enables you to resize the image to desired resolution. Different interpolation techniques can be used for sampling. In this question, you will perform experiments on different interpolation methods for upsampling and downsampling (Hint: you can use resize function for both upsampling and downsampling).

(i) List (and describe in a short paragraph) 3 interpolation methods.

(ii) Select 2 color images and downsample using the different methods with the 3 ratios below. What differences do you observe? Which interpolation method do you think works best?

- Dowsampling ratio: 0.3, 0.5, 0.7
- Please include the images in the report. Crop small regions to compare if necessary. You should have 18 images (2 images $*$ 3 methods $*$ 3 ratios).

(iii) Repeat the previous step, but this time use upsampling with the ratios below. What differences do you observe? What interpolation method works best?

- Upsampling ratio: 1.5, 1.7, 2
- Please include the images in the report. Crop small regions to compare if necessary.

(iv) Using 2 color images, downsample the images with scale 0.1 and upsample back to the original size, using all combinations of the three chosen interpolation methods. Which interpolation combination do you think works best to reconstruct the original image?

Problem 9. PyTorch tutorial and questions (5 points)

In this problem, we'll try some basic practice of the machine learning framework, PyTorch. Please follow the tutorial for cifar10 classifier (cifar10 tutorial) and answer the following questions.

(i) Login to the server for CPU/GPU resources. (0 point)

- https://datahub.ucsd.edu/ (Jupyterhub)
- Select environment (with or without GPU. You don't need GPUs in this homework.)
- Launch environment.

(ii) How many images and batches are used to train the network?

(iii) What does it mean to 'normalize' the images before training, and why do we do this?

(iv) The losses are dropping! Can you plot out the training loss?

(v) Now the network is done training. Can you check some successful cases and some failure cases (show some images classified by the network)?

(vi) Can you visualize the output of the 1st layer of CNN using one image from the training set?

References:

- deep learning 60min blitz
- pytorch with examples