E9 241 Digital Image Processing

Assignment 01

Due Date: August 31, 2025 - 11:59 pm **Total Marks:** 55

Instructions:

For all the questions, write your own functions. Use library functions for comparison only.

- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run all your functions and obtain results. Your code should be self-contained i.e., one should be able to run your code as is without any modifications.
- For Python, if you use any libraries other than numpy, scipy, scikit-image, opencv, pillow, matplotlib, pandas, and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the **results** (images or numbers) and **inferences** (very important: you may not be explicitly asked to give inferences in each question. You should always include your inferences from what you have observed). Include answers to subjective questions, if any.
- Put all your files (code files and a report PDF) into a **single zip file** and submit the zip file. Name the zip file with your name.

---

1. **Histogram Computation:** Compute the histogram of the image `coins.png`, by finding the frequency of pixels for each intensity level $\{0, 1, \ldots, 255\}$. Show the histogram by plotting frequencies w.r.t. intensity levels. Comment on what you observe. Also, find the average intensity of the image using this histogram. Verify the result with the actual average intensity.

Function	Histogram
Input	Grayscale image
Output	Frequencies at each intensity level ((a list/vector of size 256))

(5 Marks)

2. **Otsu's Binarization:** In the class, we showed that $\sigma_w^2(t) + \sigma_b^2(t) = \sigma_T^2$, where t is the threshold for binarization.

   (a) Binarize the image `coins.png` by finding the optimal threshold in t by minimizing the within class variance $\sigma_w^2(t)$ over t.

   (b) Create a image by adding constant positive offset 20 to every pixel in the image `coins.png`. Binarize this new image by finding the optimal threshold in t by maximizing the between class variance $\sigma_b^2(t)$ over t.

   Analyze the results and describe how the computed threshold values differ between the two cases. Explain why this difference occurs.

Function	Within class variance	**Function**	Between class variance
Input	Grayscale image, threshold	Input	Grayscale image, threshold
Output	Within class variance	Output	Between class variance

(10 Marks)

3. **Adaptive Binarization:** Divide the image `sudoku.png` into overlapping blocks of specified sizes. Apply Otsu's binarization independently to each block. After processing, reconstruct the full image by stitching the binarized blocks back together. In overlapping regions, determine the final pixel value using a majority vote (i.e., select the most frequent binary value , 0 or 1, at each pixel location across all overlapping blocks).

   Perform adaptive binarization with the following block, making sure each block overlaps the next one by 20%.

   (For example, if you use a 10x10 block, a 20% overlap means each block will share 2 pixels (20% of 10) with the next block.)

   (a) 5×5

   (b) 10×10

   (c) 25×25

   (d) 50×50

   (e) The full image (i.e., global binarization)

   In which situations does a smaller block size perform better than a larger one, and vice versa? Briefly comment on the visual differences and the advantages or disadvantages of using smaller versus larger blocks for adaptive binarization.

Function	Adaptive binarization
Input	Image, N
Output	Binarized image

(20 Marks)

4. **Connected Components:** Binarize the image `quote.png` and use connected component analysis to extract characters, excluding punctuation marks. Identify the largest connected component corresponding to a character, color it red, and return a modified image where this component is highlighted in red over the original image.

   In the class, we have seen connected component analysis using 4-neighbour connectivity. Formulate the logic to implement and find the connected components using 8-neighbour connectivity of pixels. The 8-neighbours of a pixel (x, y) is defined as:

$$N_8(x, y) = \{(x-1, y), (x-1, y-1), (x, y-1), (x+1, y-1), \\ (x+1, y), (x+1, y+1), (x, y+1), (x-1, y+1)\}$$

   Note: You are only allowed to use a maximum of two for-loops that scan through all pixels in the image (horizontal and vertical directions). The rest of the code that may require further for-loops are to be written in a vectorized manner. Non-vectorized codes will be penalized. (Refer to Teams `Files -> Programing Tutorial` and the internet to get resources on vectorization)

Function	Get Largest Character
Input	Image
Output	Largest connected component character

(20 Marks)