HW1: Fundamentals

DIP Teaching Stuff, *Sun Yat-sen University*

Welcome to your first DIP homework! This homework consists of two parts. In the first part are some simple questions, while the second part requires certain programming works. Please submit a report (in **PDF** format) and all relevant codes as the homework solution. Warning: We encourage discussions among students, but homework solutions should be written and submitted **individually**, without copying existed answers. Plagiarism = Fail. Besides, there may be at least 30% penalty for late homework.

1 Exercises

Answer the following questions in your report.

1.1 Storage ($3 * 3 = 9$ Points)

If we consider an N-bit gray image as being composed of N 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane N all the highest-order bits, then given a 1024×2048, 128-level gray-scale image:

1. How many bit planes are there for this image?
2. Which panel is the most visually significant one?
3. How many bytes are required for storing this image? (Don't consider image headers and compression.)

1.2 Adjacency ($3 * 3 = 9$ Points)

Figure 1 is a 5×5 image. Let $V = \{1, 2, 3\}$ be the set of pixels used to define adjacency. Please report the lengths of the shortest $4-$, $8-$, and $m-$path between p and q. If a particular path does not exist, explain why.

3	4	1	2	0
0	1	0	4	2(q)
2	2	3	1	4
(p)2	0	4	2	1
1	2	0	3	4

Figure 1: *Adjacency.*

1.3 Logical Operations ($3 * 3 = 9$ Points)

Figure 2 are three different results of applying logical operations on sets A, B and C. For each of the result, please write down one logical expression for generating the shaded area. That is, you need to write down three expressions in total.

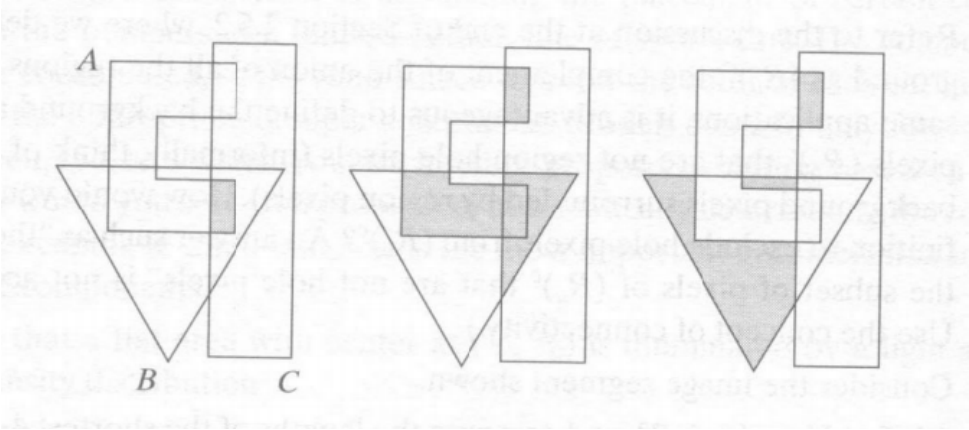

Figure 2: *Logical Operations.*

2 Programming Tasks

Write programs to finish the following two tasks, and answer questions in your report. Don't forget to submit all relevant codes.

2.1 Pre-requirement

Input Please download the archive "hw1.zip", unzip it and choose the image corresponding to the last two digits of your student ID. This image is the initial input of your programming tasks in HW1. For example, if your student ID is "14110563", then you should take "63.png" as your input. You can convert the image format (to BMP, JPEG, ...) via Photoshop if necessary.

Make sure that you have selected the correct image. Misusing images may result in zero scores.

Language Any language is allowed, but you can only use element-wise indexing and scalar operations. Advanced indexing (including but not limited to row-wise/col-wise/range indexing) and vectorization are forbidden. As an example, given a matrix A, if you want to divide all its elements by 2, you should use loops over matrix elements, sequentially performing "A[i][j] /= 2", instead of directly writing "A /= 2" in python with numpy. This "annoying" requirement will be removed in future homework.

Others There remain some issues that you should pay attention to:

1. You can use third-party packages for loading/saving images.
2. Good UX (User Experience) is encouraged, but will only bring you negligible bonuses. Please don't spend too much time on it, since this is not an HCI course.

3. Keep your codes clean and well-documented. Bad coding styles will result in 20% penalty at most.

2.2 Scaling (45 Points)

Write a function that takes a gray image and a target size as input, and generates the scaled image as output. Please use **bi-linear** for interpolation. The function prototype is "**scale(input_img, size)** $\rightarrow$ **output_img**", where "input_img" and "output_img" are two-dimensional matrices storing images, and "size" is a tuple of (width, height) defining the spatial resolution of output. You can modify the prototype if necessary.

For the report, please load your input image and use your "scale" function to:

1. Down-scale to 192×128 (width: 192, height: 128), 96×64, 48×32, 24×16 and 12×8, then manually paste your results on the report. (10 Points)
2. Down-scale to 300×200, then paste your result. (5 Points)
3. Up-scale to 450×300, then paste your result. (5 Points)
4. Scale to 500×200, then paste your result. (5 Points)
5. Detailedly discuss how you implement the scaling operation, i.e., the "scale" function, in **less** than 2 pages. Please focus on the algorithm part. If you have found interesting phenomenons in your scaling results, analyses and discussions on them are strongly welcomed and may bring you bonuses. But please don't widely copy/paste your codes in the report, since your codes are also submitted. (20 Points)

Attention: you should not rescale your scaling results in your report, unless they exceed the page height/width.

2.3 Quantization (28 Points)

Write a function that takes a gray image and a target number of gray levels as input, and generates the quantized image as output. The function prototype is "**quantize(input_img, level)** $\rightarrow$ **output_img**", where "level" is an integer in $[1, 256]$ defining the number of gray levels of output. You can modify the prototype if necessary.

For the report, please load your input image and use your "quantize" function to:

1. Reduce gray level resolution to $128, 32, 8, 4$ and 2 levels, then paste your results respectively. Note that, in practice computers always represent "white" via the pixel value of 255, so you should also follow this rule. For example, when the gray level resolution is reduced to 4 levels, the resulting image should contain pixels of $0, 85, 170, 255$, instead of $0, 1, 2, 3$. (8 Points)
2. Detailedly discuss how you implement the quantization operation, i.e., the "quantize" function, in **less** than 2 pages. Again, please focus on the algorithm part. Analyzing and discussing interesting experiment results are also welcomed, but please don't widely copy/paste your codes in the report (20 Points).