National University of Singapore

Department of Mathematics

02/2023 Semester II MA4268 Mathematics in visual data processing **Project 1**

Goal

Discrete Wavelet Transform (DWT) and its inversion are one essential transform in digital image processing. The operations it involves are also widely used in other algorithms. The goal of this project is to implement two-dimensional Haar-based DWT and its inverse DWT for 2D grayscale images in Python. Recall that the low-pass filters and 3 wavelet high-pass filers of a Haar wavelet transforms are given

$$H = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; G_1 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}; G_2 = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}; G_3 = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Implementation

In this project, the following TWO functions are expected. For simplicity, you can assume that the image is always of size $2^N \times 2^N$, N is an integer. For example, the size of images are 256×256, 512×512, or 1024×1024. The image can be assumed to be a gray-scale image.

haar2d(im,lvl)

Computing 2D discrete Haar wavelet transform of a given ndarray *im*

Parameters: **im: ndarray**
An array representing image
lvl: integer
An integer representing the level of wavelet decomposition

Returns: **out: ndarray**
An array representing Haar wavelet coefficients with *lvl* level. It has the same shape as *im*

ihaar2d(coef,lvl)

Computing an image in the form of ndarray from the ndarray *coef* which represents its DWT coefficients.

Parameters: **coef: ndarray**
An array representing 2D Haar wavelet coefficients
lvl: integer
An integer representing the level of wavelet decomposition

Returns: **out: ndarray**

An array representing the image reconstructed from its Haar wavelet coefficients.

Forbidden functions
You may NOT call any function of discrete wavelet transform or inverse wavelet transform from existing packages such as scipy and pywavelet

Allowed functions
You are allowed to call the routines "convolve", "convolve2d" or similar functions in numpy or scipy, as well as down-sampling and up-sampling functions. If you are unsure whether some function can be called, please ask me in advance.

Remark on implementation

- The function "convolve" or "convolve2d" in existing package often have different setting for **boundary extension**. When calling the existing function, pay attention to its boundary treatment.
- Pay attention to the starting index of down-sampling and up-sampling to avoid **possible mis-alignment** of coefficient matrix.
- Do not use "matplotlib" for reading/writing images, using the package "pillow" for reading/writing images.

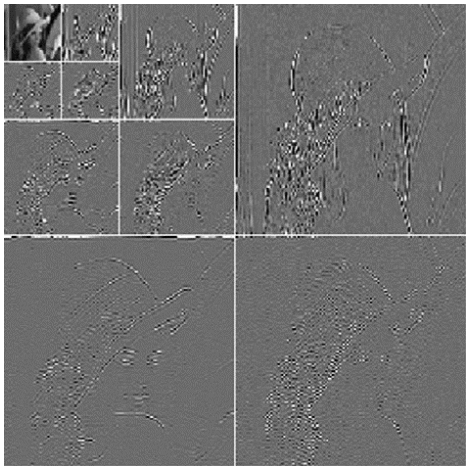
Illustration of the implementation. See Figure 1 for an demonstration of the format of expected wavelet-decomposition.

Image data and experiments
One image is provided: a gray-scale image of size 512×512. You can use it to test your code, as well as other images that you prefer.

Academic integrity
It is an individual project. You are expected to implement the project on your own. You may discuss the ideas with your classmates, but the sharing of code is strictly prohibited. You are not allowed to use any code from online sources or from other classmates' projects, which will be checked via some standard code plagiarism checking system. Any suspicious code identified by the system will lead to further investigation. Anything that breaks these principles will commit a violation of the Honor Code, and will be reported to the university for the corresponding action.

Grading policy

(a) input image



(b) Visualization of 3-level wavelet coefficients

Figure 1: Visualization of 3-level Haar wavelet decomposition of image. Note that different visualization scheme will appear to be different

The grade of the submitted project is based on the following three factors: 1) How well the submission meets the guidelines of the project; 2) Correctness of the output; (3) Computational efficiency in terms of running time, and (4) Quality of your report.

Submissions

You are required to submit the following

1. The python package "main.py" that contains the functions of haar2d, ihaar2d; and other supporting functions if any. A reference implementation of main.py is attached in the zip file. Please modify it accordingly. To facilitate the grading process, do not modify the IO format of "main.py".

   Warning: DO NOT submit a single jupyer notebook, the file with extension .ipynb, that contains everything.

2. A technical report to discuss what you would like to mention about your codes, if any.

Please package all your files in a single zip file named by your student ID and upload the zip file through the assignment section of the module in online CANVAS system. **DO NOT email the files to my email address**. The deadline for submission is **FRIDAY, 10-Mar-2023**. Any submission after the deadline will lead to some penalty on grade.

If you have any question, please contact me by email: matjh@nus.edu.sg for help.