HW4

THE OHIO STATE UNIVERSITY

Preparation

- Please read Chapter 23 about the Gaussian pyramid and Laplacian pyramid
- The Gaussian pyramid down-samples an image sequentially
- The Laplacian pyramid records “information loss” during down-sampling
- By combining them, we can reconstruct the image of the original size

Caution

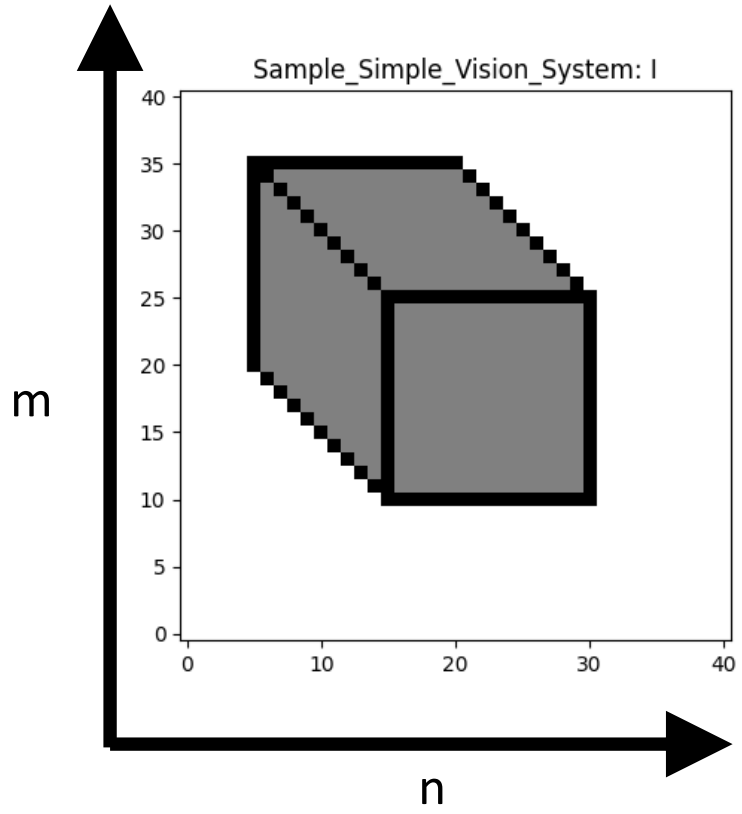
- Please do not import packages (like scikit learn) that are not listed in the provided code.

- In this homework, you are NOT allowed to use NumPy's or other Python libraries' built-in **convolution, filter functions, down-sampling, up-sampling, Gaussian pyramid, and Laplacian pyramid functions.** If you use them, you will get 0 points for the entire assignment.

- This homework uses the same Convolution function and kernel function you saw in HW 2 (but you don't need to re-implement them again).

Convention

- In this homework, given a map (or a matrix), say I
  - I[n, m] means the i-th horizontal index (left-right) and j-th vertical index (bottom-up)
  - n >= 0, m >= 0



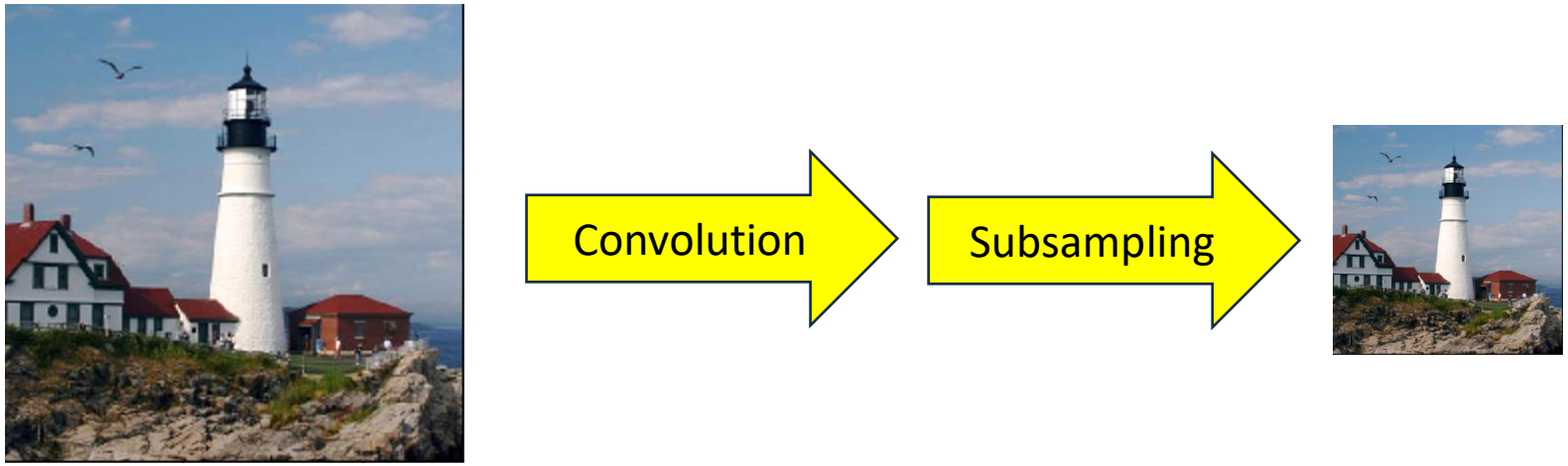
Color images

- Please note that a color image I means that the image I is a 3D tensor. The 3rd dimension corresponds to R, G, and B.

- In this homework, you will process each channel separately. You can extract each by I[:, :, c], where c is between 0 and 2.

Question 1: Down-sampling

- There are many variants, but please follow the formula below.
  - Given an image I, we will first convolve it with a 2D binomial filter to obtain I_convolved
    - I and I_convolved have the same size
  - We will then sub-sample the convolved image I_convolved by a factor of two
    - If I_convolved is 128 x 128, the subsampled image I_down is 64 x 64 by keeping pixels at "even" Python indices. That is, I_down[n, m, :] = I_convolved[2n, 2m, :]



Question 2: Gaussian pyramid

- Create a sequence of sub-sampled images

Question 3: Up-sampling

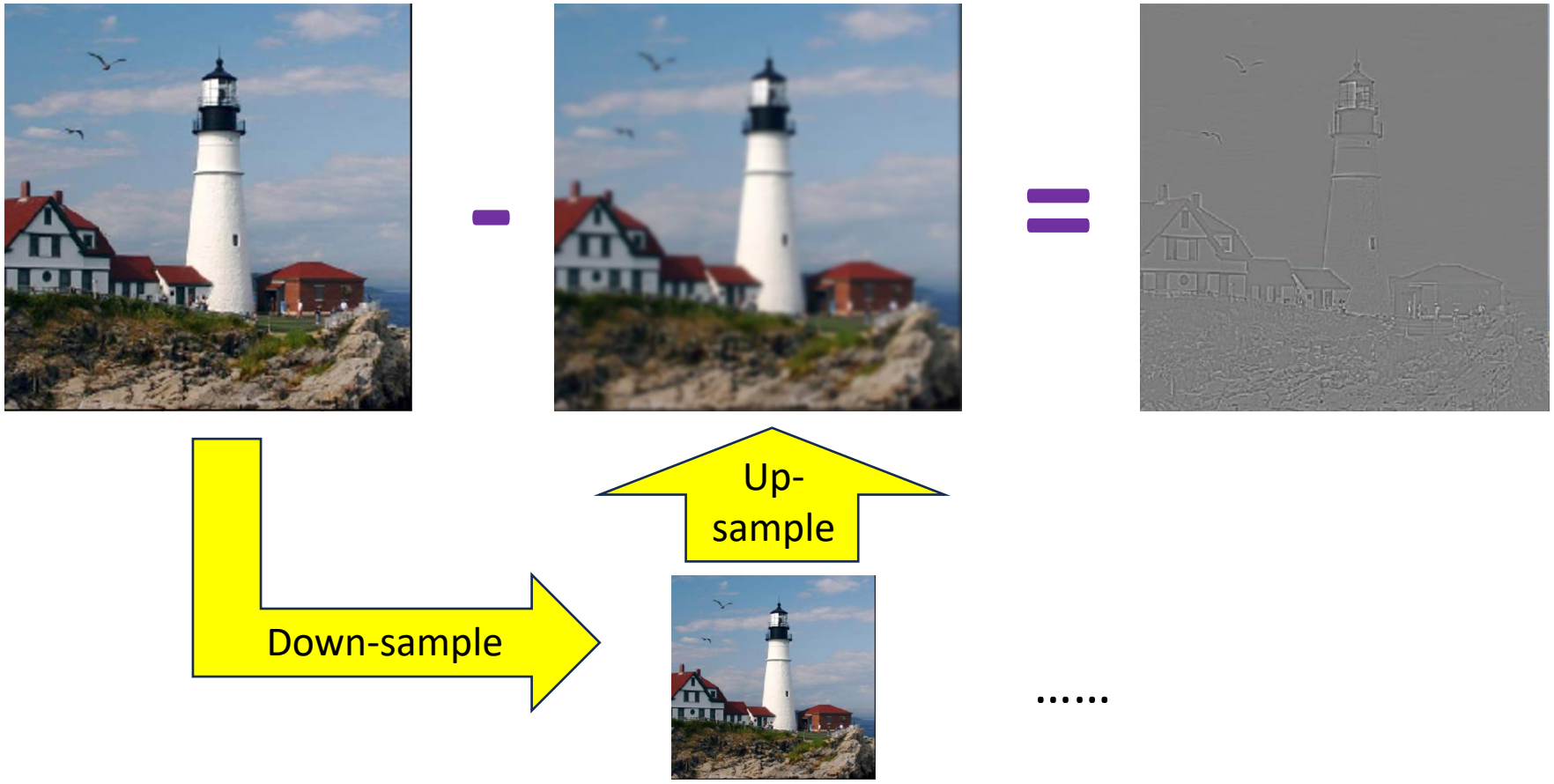
- There are many variants, but please follow the formula below.
  - Given an image I, we will first double its size in each dimension to obtain I_up
    - If I is 64 x 64, I_up is 128.
    - I_up[n, m, :] = 4 * I[n/2, m/2, :] when n and m are "even" Python indices
      - **Please make sure you multiply the pixel values by 4**
    - I_up[n, m, :] = 0 when n and m are "odd" Python indices
  - We will then convolve I_up with a 2D binomial filter to obtain I_convolved
    - I_up and I_convolved have the same size

Double-size

Convolution

Question 4: Laplacian pyramid

- Create a sequence of residual images



Question 5: Image reconstruction