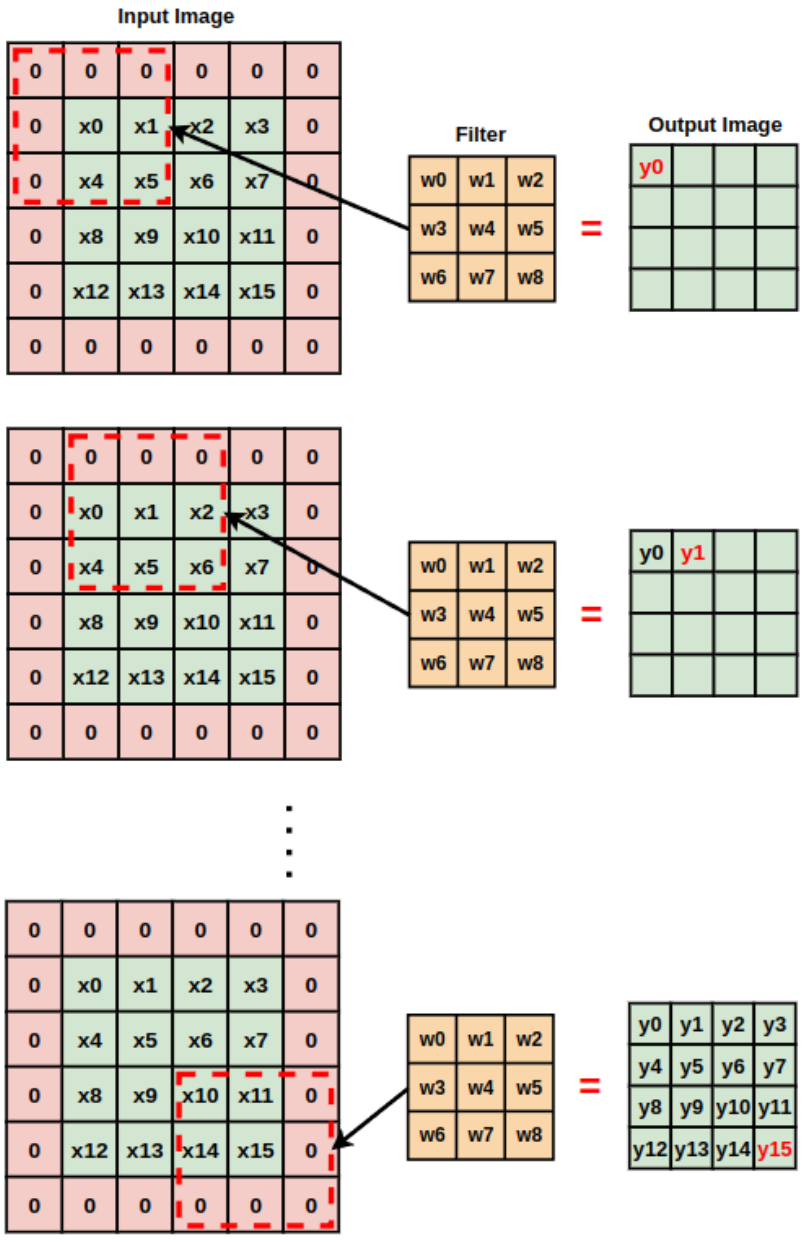
Programming Assignment 1 - 2D Convolution

1. Task Description

We will implement 2D convolution in CUDA. 2D convolution works by sliding a 2D filter(also called kernel) over a 2D input image. As the filter is swept across the input image, the input elements are multiplied by the filter weights, and results are accumulated to the output image.

See the figures below for an example, where the input image size is 4x4, the filter size is 3x3, and vertical/horizontal strides are 1. We start sweeping from the top left of the input image. We sweep horizontally first, then move on to the next row and repeat.



Notice we pad the input image so that the output image size is equal to the input image size. You can do this padding when you first load the input image in your main function, or you can have some checks(if-else) in your CUDA kernel and pad the input as you perform the convolution. For reference, this is called “same” size convolution in MATLAB/Scipy terminology, which Nvidia also seems to use as ground truth for their libraries[1].

You also do not need to flip the filter horizontally/vertically before the actual computation, as opposed to what real convolution implementations do (Matlab, scipy). In other words, you can assume the filters that are given to you are already flipped.

You are tasked with implementing this convolution algorithm in CUDA. You are **not** allowed to use any CUDA libraries(CUTLASS, CuDNN etc.). You also need to write the host code to read the inputs, launch the kernel and print the output. Develop two versions of the host code, one uses `cudaMalloc` and `cudaMemcpy` to move data explicitly and the other uses `cudaMallocManaged` to leverage unified virtual memory to move the data. Finally, prepare a 1 page report briefly describing how your host code and the kernel work. Also report the timing results for both versions.

2. Constraints

Input image height: **H, 64 <= H <= 4096**
Input image width: **W, 64 <= W <= 4096**
Filter height = Filter width: **R, R = {3, 5, 7]**
Horizontal stride = Vertical stride: **U, U = 1**
Number of input images: **N, N =** 1
Number of filters: **K, K =** 1
Both the input and the filter will have **float** values in them.

3. Input format and expected output

The input to your program will be two .txt files; one will contain the input image the other will contain the filter. The first two lines in the input image file will be **H** and **W**. Following lines will be the elements themselves, in row-major order. The first line in the filter file will be **R**. Following lines will be the elements themselves, in row-major order. The names of the input .txt files are not fixed, so your program should just read them as arguments from the command line.

Your program should only print out the output image, one element at each line, in row-major format, **and nothing else. The values should be printed with 3 decimal points precision, not more nor less. Print to stdio and not to a file!**

Ultimately, we will run your program like this,

```
./conv2dV1 input.txt filter.txt > output.txt
```

And simply `diff` the output with the expected output. Therefore, make sure you are not printing anything other than the output image.

We will test your program on the Hydra cluster, so make sure you get the expected outputs there. You will lose points if your code works on your personal computer, but not on the Hydra cluster.