

# Homework-01

## Implement 2D convolution

### Background

Convolutions are important. They are the building block of convolutional neural networks. This homework is designed to give you an introduction to the tools used in our course.

### Deliverables

- Two input images of size 1280x720, 1920x1080
- Total 12 output images (preferably rescale them to smaller size) (**Part A**)
- Charts comparing the performance (**Part B, C, and D**)
- Source code: `main.c`, `conv.py`, `main.py`.
- Write a small report and put your code in `<your_purdue_username>_HW<0X>.pdf`

### API

Python:

```
Conv2D(in_channel, o_channel, kernel_size, stride, mode)
[int, 3D FloatTensor] Conv2D.forward(input_image)
```

`Conv2D` is a class and it has a `forward` function as one of its method (apart from its constructor).

C:

```
[int] c_conv(in_channel, o_channel, kernel_size, stride)
```

### Instructions

1. Class `Conv2D` must be defined in `conv.py` file with `__init__()` and `forward()` functions in it.

#### Part A

2. Initialize `Conv2D` in `main.py` (`conv2d = Conv2D(*args)`) for one of the task.
3. Call `conv2d.forward()` with your input image `[3D FloatTensor]`. The `forward()` function must return output `[int, 3D FloatTensor]`.
4. Save each channel of output tensor separately as a grayscale image in your main repository.
5. Repeat 2-4 for all the three tasks.

#### Part B

6. Initialize `Conv2D` using values of Task 1 and set `o_channel` to  $2^i$  ( $i = 0, 1, \dots, 10$ ) and `mode='rand'`.
7. Plot the time taken for performing each `forward()` pass as a function of  $i$ .

## Part C

- Initialize `Conv2D` using values of Task 2 with `kernel_size=3, 5, ..., 11` and `mode='rand'`.
- Plot number of operations (`int` returned by `forward()`) used to perform convolution as a function of `kernel_size`.

## Part D

- Repeat **Part B** using `c_conv(*args)` function which will be C implementation of convolution.

	Task 1	Task 2	Task 3
<code>in_channel</code>	3	3	3
<code>o_channel</code>	1	2	3
<code>kernel_size</code>	3	5	3
<code>stride</code>	1	1	2
<code>mode</code>	'known'	'known'	'known'
<code>Kernel to use</code>	K1	K4, K5	K1, K2, K3

## Kernel definitions

```
K1 = [-1, -1, -1; 0, 0, 0; 1, 1, 1]
K2 = [-1, 0, 1; -1, 0, 1; -1, 0, 1]
K3 = [1, 1, 1; 1, 1, 1; 1, 1, 1]
K4 = [-1, -1, -1, -1, -1; -1, -1, -1, -1, -1; 0, 0, 0, 0, 0; 1, 1, 1, 1, 1; 1, 1, 1, 1, 1]
K5 = [-1, -1, 0, 1, 1; -1, -1, 0, 1, 1; -1, -1, 0, 1, 1; -1, -1, 0, 1, 1; -1, -1, 0, 1, 1]
```

*Known modes* mean use above kernel definitions. When you need to use only one kernel then use K1, for 2 kernels use K1-K2. When `mode='rand'`, initialize kernel with random values i.e. you do NOT need to use K1-K5.

For part B, C, and D; if your system is taking more than 1 hour then you do not need to go beyond that point (no need to further increase output channels or kernel size). Just mention it in your report that your system was taking too long to perform those computations and plot extrapolated results.

## Format

Upload your homework on *Blackboard* as **ZIP** file.

Your folder name should be as following:

`<your_purdue_username>_HW<0X>`

For example, my folder name for first homework will look like `aabhigh_HW01`.

Please refrain from using global variables in your code unless it is specifically mentioned in the homework.

Results need to be described in the `<your_purdue_username>_HW<0X>.pdf` (eg. aabhash\_HW01.pdf). Do **NOT** forget to put all deliverables in your assignment folder.

Keep in mind that **your code might be made public after due date**. In case you have any problem with that then you should notify your instructor/TA beforehand.

Have fun! :D

## NOTE:

My script will run something similar as the following to check whether you adhered to the given instructions.

```
from conv import Conv2D

# input_image = get_image

conv2d = Conv2D(in_channel=3, o_channel=1, kernel_size=3, stride=1)
Number_of_ops, output_image = conv2d.forward(input_image)
Assert output_image.size() == expected_image.size()
```