for C implementation, what I need is just comparison, so you do not need to load image in C. Just use a randomly initialized array for input image (use the same resolution provided in the homework) and report the values for it

As I told in class, use OpenCV or PIL to load images in Python and preferably use Python3 or greater

In PyTorch it is c x h x w

Use PIL to load images

number of operation is multiplication and addition both

1) Both images are RGB. This is referenced by in\_channel being 3

2) I don't think we need to do any padding for this assignment

3) in channel=3 means number of channels in input (RGB here) out\_channel means how many channels will the output of your convolution have

4) int is the number of operations carried out by the convolution and 3D float tensor is the output of the convolution

5) we have to write convolution code from scratch

the same kernel will be repeated along all 3 channels

write convolution from scratch using numpy

Output of the conv2d.forward function is a 3D tensor, so output should include channels, height and width. Each channel is then supposed to be separately saved as a grayscale image (this will be 2D) and you can use matplotlib as far as I know.

An output channel is generated when you convolve your input channel with a single kernel(filter).

Let’s say you have 1 input image of size [720, 1280, 3]; and 2 kernels (K1 and K2), each of size [5, 5].  
On a side note the kernel that you will actually use will be [5, 5, 3].

When you convolve your input image with K1, you will get an output channel (say o1). o1 will be of size [716, 1276, 1].

Similarly, if you convolve your input image with K2, you will get another output channel o2. o2 will also be of size [716, 1276, 1].

you are supposed to use the kernels mentioned in the last row. So, for task2, it will be k4, k5 and for task3 it will be k1, k2 and k3

to get output channel=1, 2, 2^i... you will need 1, 2, 2^i... number of kernels

the values so that they lie between 0 and 1 if you are using FloatTensor to represent your image and perform convolutions on it. So, once you convert it back to a PIL image, it will have the correct values **(checked)**

You should have 2 loops, one each for the rows and columns of your output image. A convolution on the 1280 x 760 x 3 image takes me less than 10 seconds, running on a MacBook Pro

Say for example you have an image with 3 channels (RGB), rather than convolve each channel individually with a kernel and add the result (which would require 3 convolutions, 1 for each channel); why not add the RGB channels first and then perform just one convolution operation.

3 Additions + 1 Convolution is better than 3 Convolutions + 1 Addition

Please embed/show your plots and results in the README.md file. Do NOT put the plots in different folder and mention that those plots are located there. Your README.md file is proper report.

Store image values as torch.FloatTensors. This tensor will be input to your forward function and you can access its size by variable\_name.size()