

Homework 5 – Convolution Accelerator

Handout: 2025/11/17

Due: 2025/12/15

- (VGG-16) Fig. 1 shows the VGG-16 Convolutional Neural Network (CNN) model with a total of 16 layers, including 13 convolutional layers and 3 fully connected layers. The final softmax generates the probability of every class. Table 1 shows the information of the first two layers.

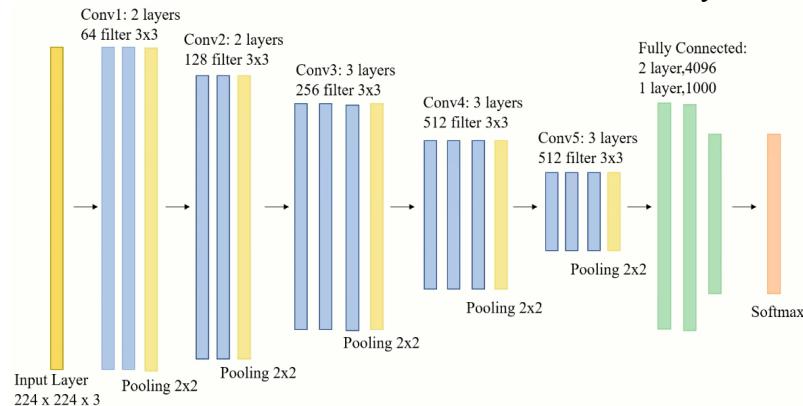


Fig. 1. VGG-16 CNN model.

Table 1. Information for the first two layers in VGG-16

# of input channels	# of output channels	Input feature map size (before padding)	Filter Kernel size	Stride
3	64	224x224	3x3	1
64	64	224x224	3x3	1

- (DNN Architecture) Design a deep neural network (DNN) hardware accelerator similar to that shown in Fig. 2 to speed up the convolution operations in the first two VGG-16 layers. You can determine the ICP and OCP for your own design. Calculate the total number of cycles and amount of memory accesses for executing the first layer and the second layer. Note that the execution cycles and memory access depend on ICP and OCP

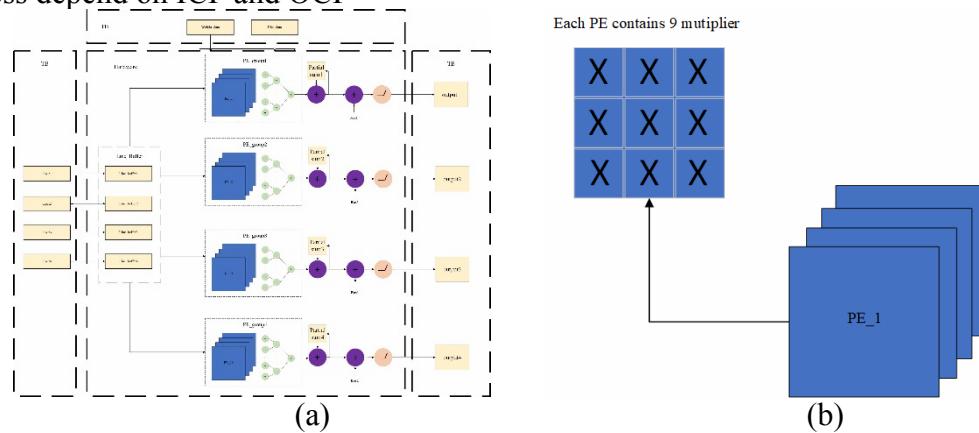


Fig. 2. (a) Overall architecture of the DNN accelerator (ICP=4, KWP=9, OCP=4). (b) Architecture of PE.

3. (Padding) First, pad the input images as shown in Fig. 3 to obtain the output feature maps of the same size after 3x3 filter kernel convolution. Note that after padding, the featuremap size is 226x226.

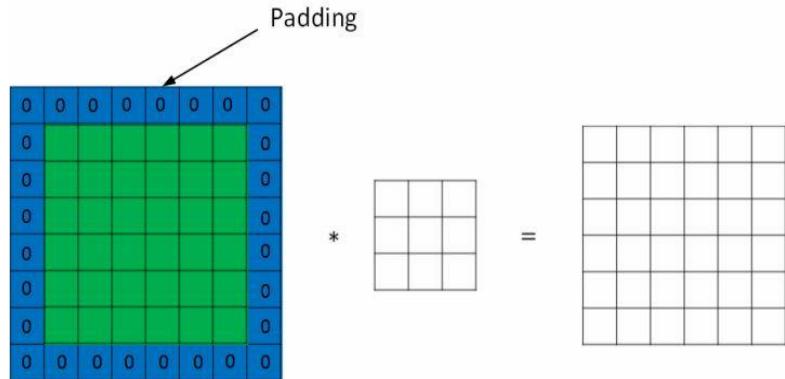


Fig. 3. Padding.

4. (Line Buffer) Use the line buffer shown in Fig. 4 to reduce repeated accesses of the same data.



Fig. 4. Line buffer and the operation.

5. (ReLU) Non-linear activation function Rectified Linear Unit (ReLU) shown in Fig. 5 is used after convolution.

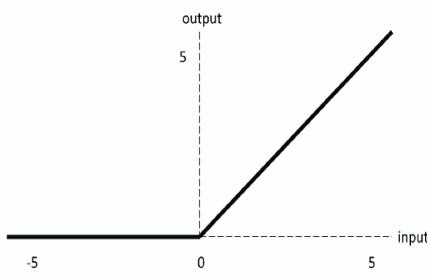
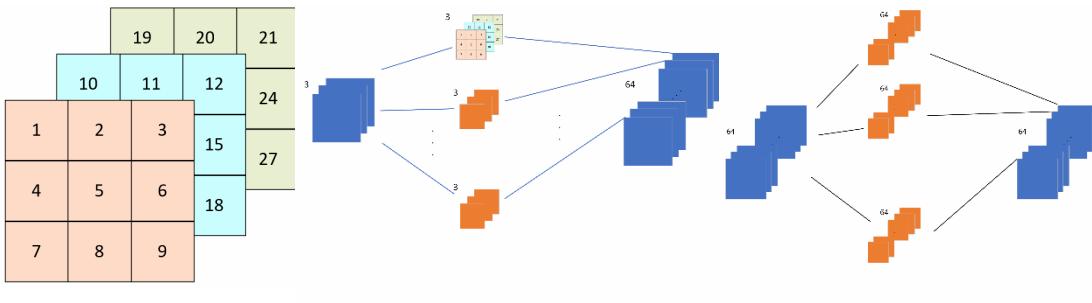


Fig. 5. ReLU.

6. (Provided Data) The following image data, the weights, and provided. The data structure is

- (a) input
- (b) filter weights (conv1_kernel_hex.txt, conv2_kernel_hex.txt)
- (c) biases (conv1_bias_hex.txt, conv2_bias_hex.txt)

data (.txt), including initial input biases of the first two layers, are shown in Fig. 6.
image(cat224.bmp)



(a)

(b)

(c)

Fig. 6. (a)order of the provided weights and biases in the text files.
(b)First layer of VGG-16. (c)Second layer of VGG-16.

7. (Featuremaps) Fig. 7 shows the initial input image and the first output featuremaps in the 1st and 2nd layers. Fig. 8 and Fig. 9 respectively show all the output featuremaps in the 1st and 2nd layers.

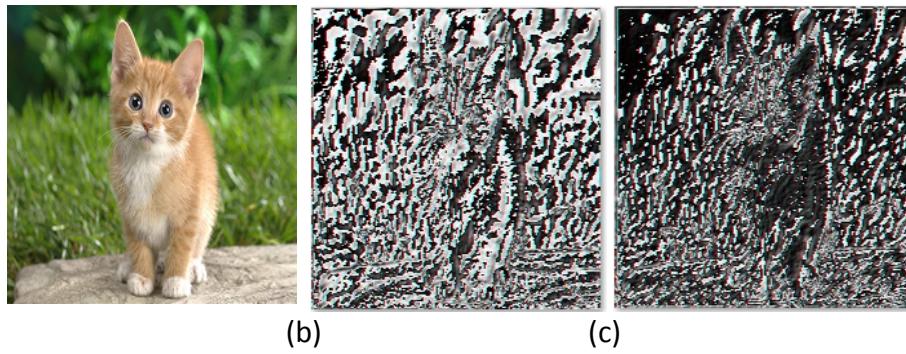


Fig. 7. (a) Input image. (b) 1st output featuremap of the 1st layer. (c) 1st output featuremap of the 2nd layer.



Fig. 8. Output featuremaps of the 1st layer.



Fig. 9. Output featuremaps of the 2nd layer.

8. Report Requirement

檔案類須含有:

- I. Testbench (20%)
- II. Verilog RTL code & Gate-level code
 - LineBuffer (5%)
 - PE (5%)
 - AdderTree (5%)
 - ReLU (5%)
- III. Image
 - Conv1 image *64(15%)
 - Conv2 image *64(15%)

word報告類須含有:

- IV. 硬體架構圖解釋(10%)
- V. Area資訊和critical path資訊，不需要做optimization，合成出來即可(10%)
- VI. 心得(10%)

以上打包成HDL_HW5_MXXXXXXXXX.zip壓縮檔並繳交