**HW#08: Convolutional Neural Network for Super-resolution, Reference Software**

**Issued:** May 2 (Tuesday), 2023 **Due:** May 8 (Monday), 2023

**What to turn in**: **Copy the text from your MODIFIED codes and paste it into a document**. If a question asks you to plot or display something to the screen, also include the plot and screen output your code generates. Submit either a \*.doc or \*.pdf file.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 1 (10p): Convolutional Layer**

Implement a function to calculate a convolutional layer in Matlab. Please see the description in the lecture note for details.

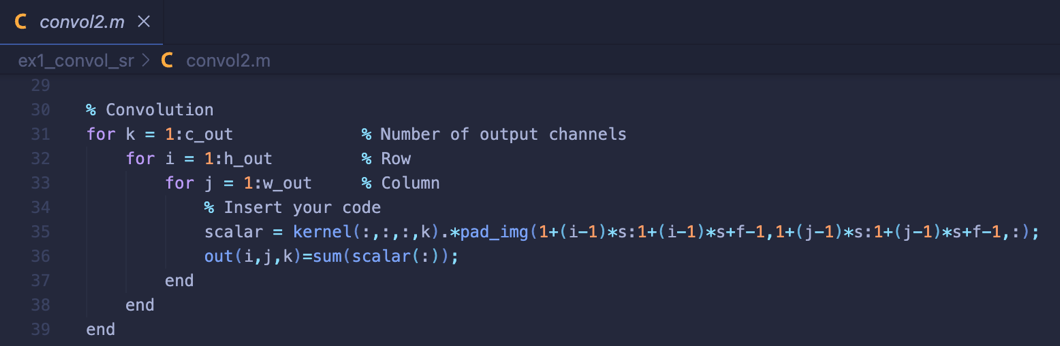
What you have to do:

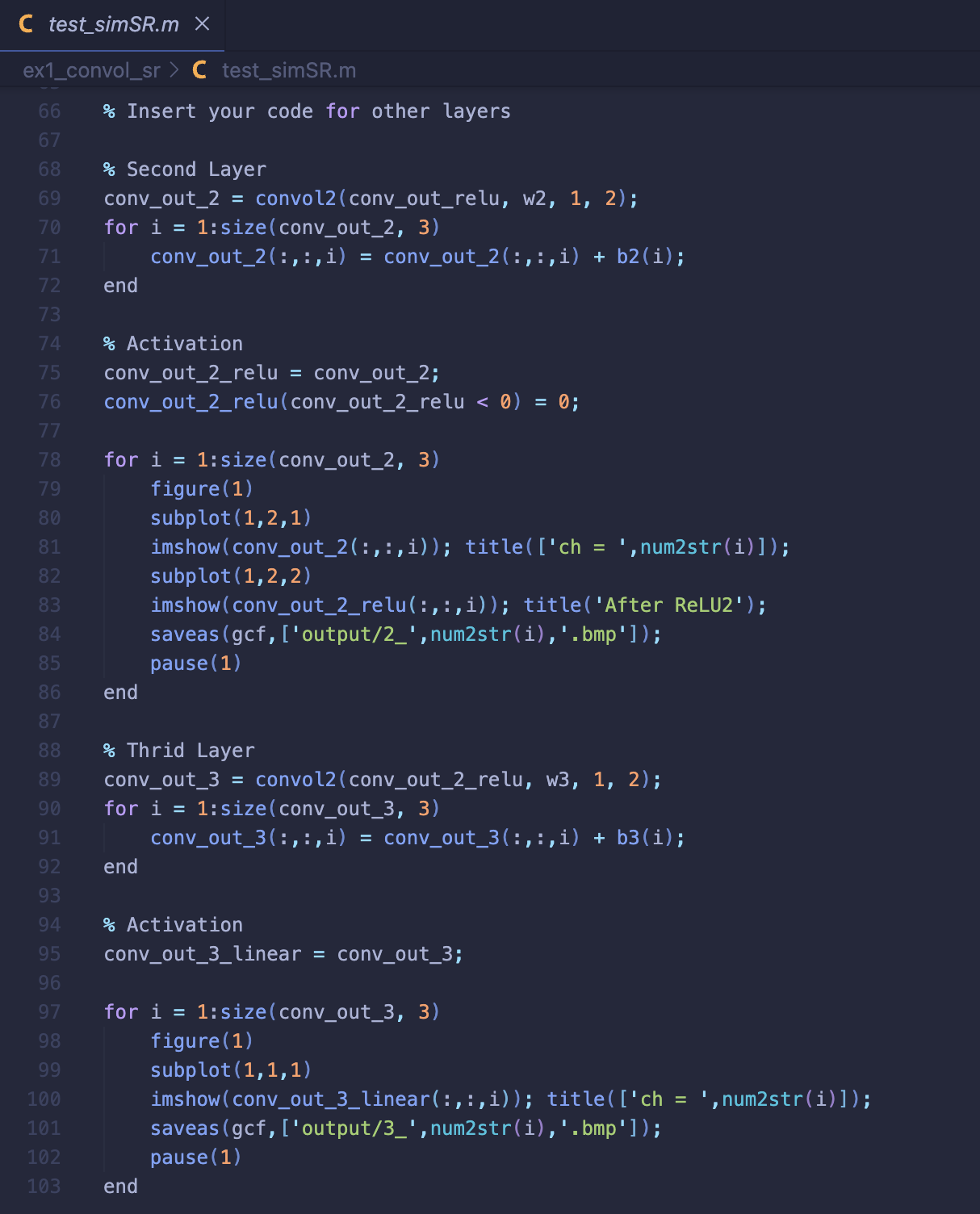
1. Complete the missing codes in convol2.m and test\_SR.m
2. Output the feature maps of all three layers.
3. For each convolutional layer, let num\_ops be the number of multiplication operations to calculate **ONE output pixel**. Complete the following table:

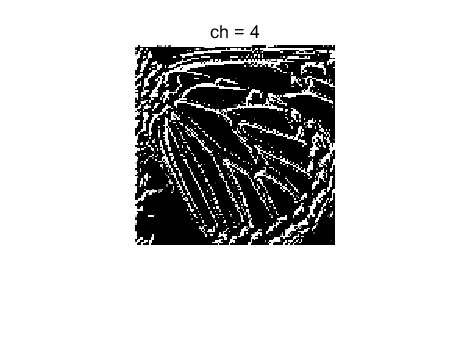
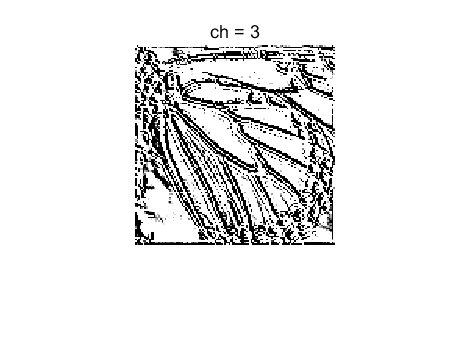
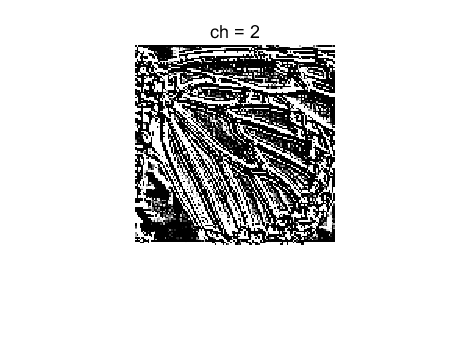
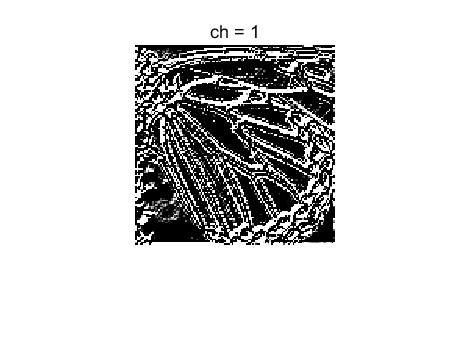
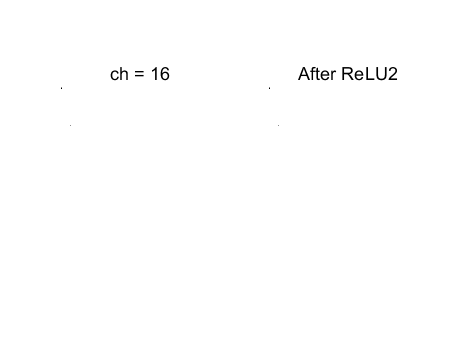
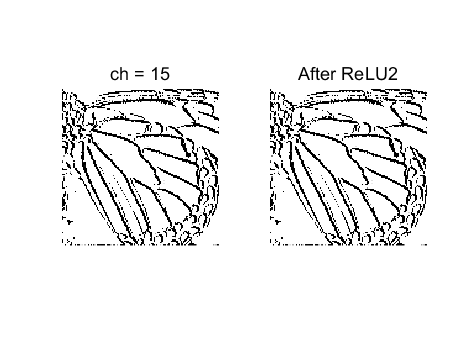
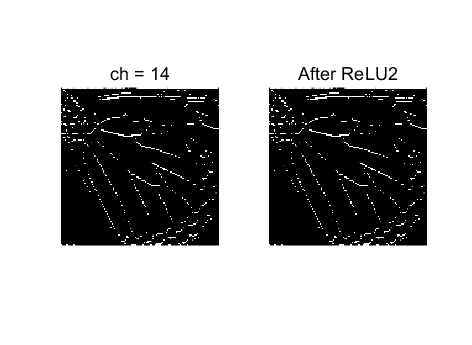
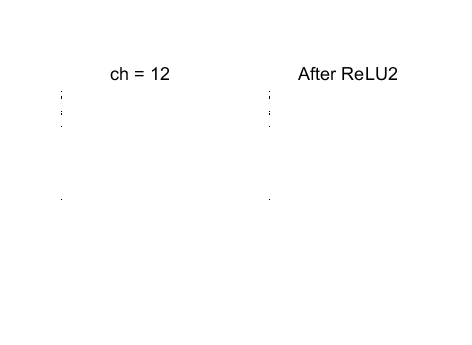
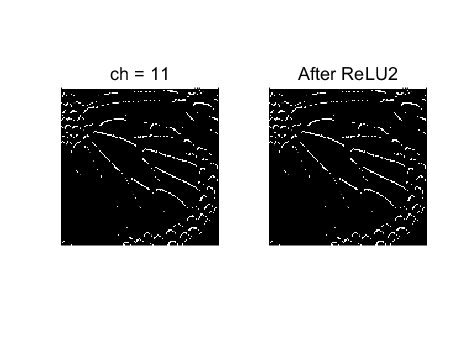
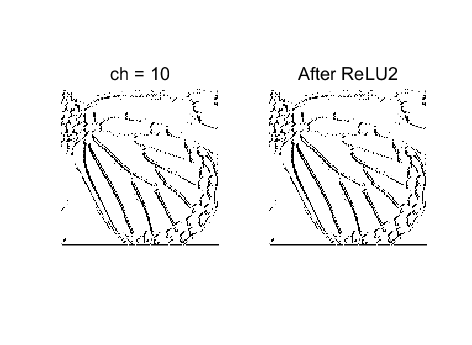
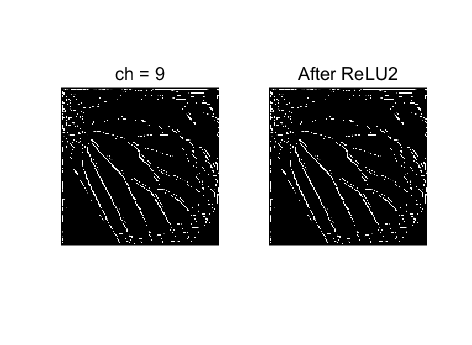
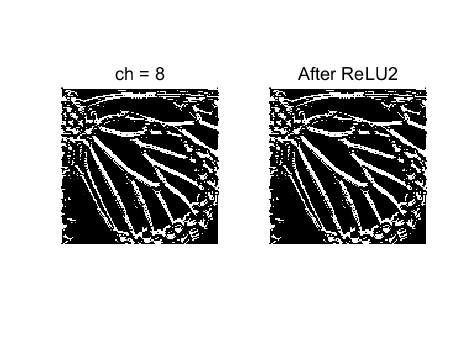
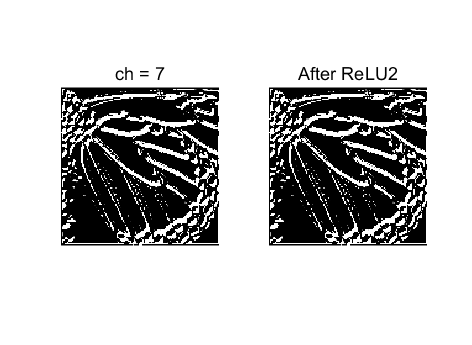
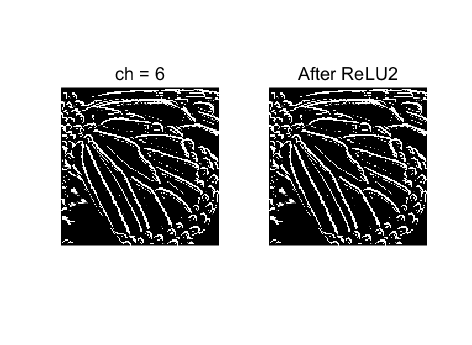
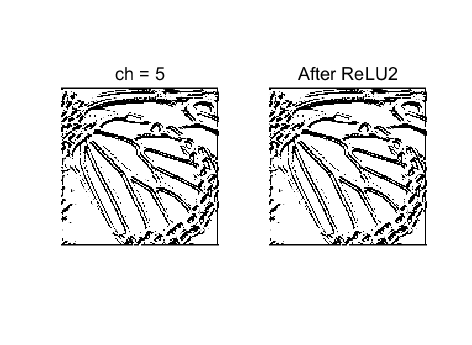
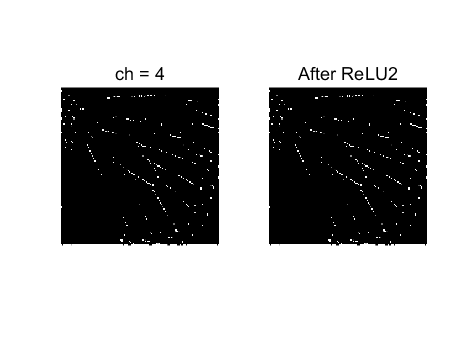
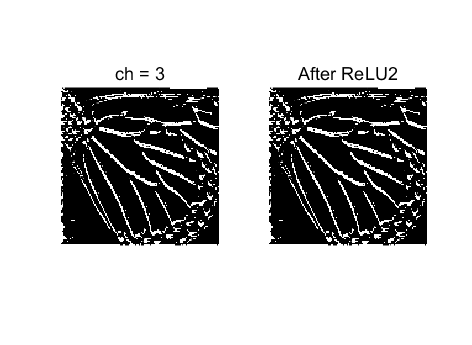
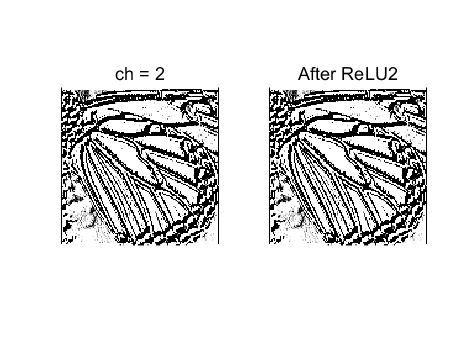
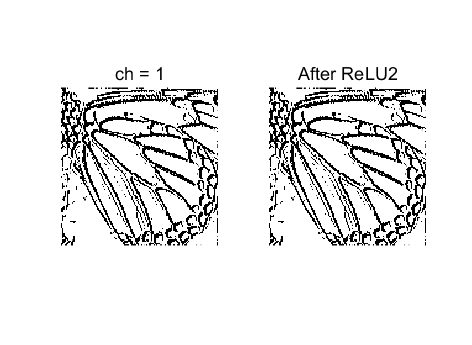
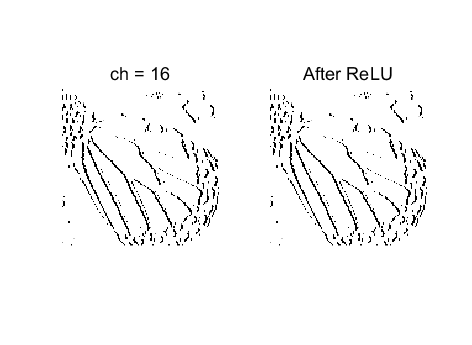
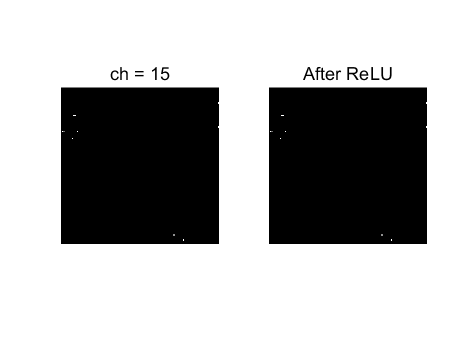
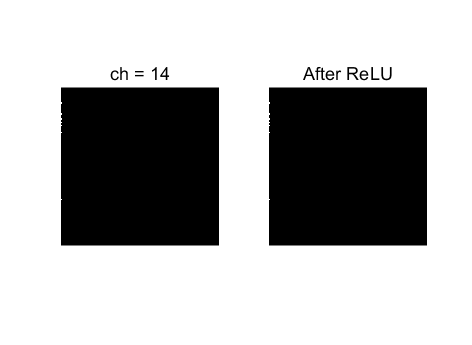
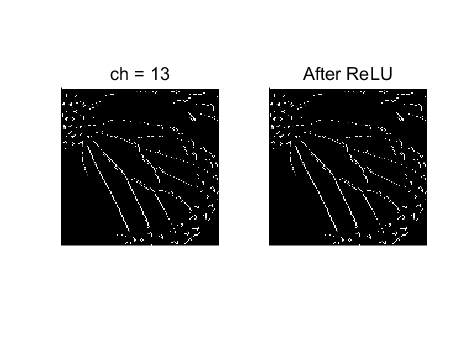
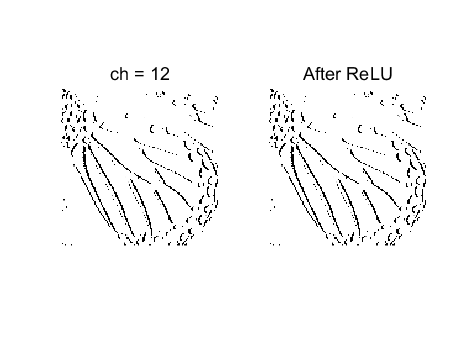
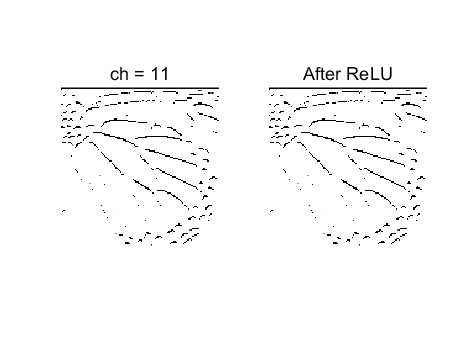
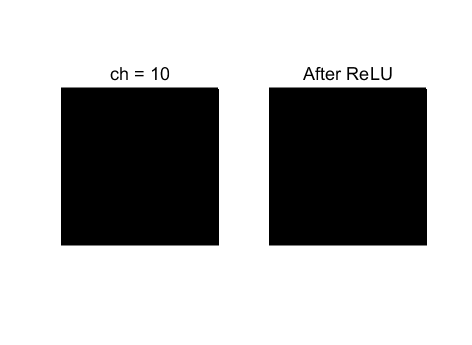
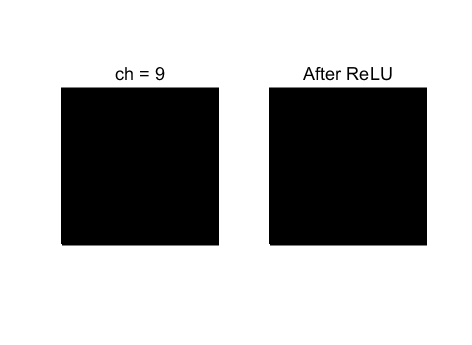
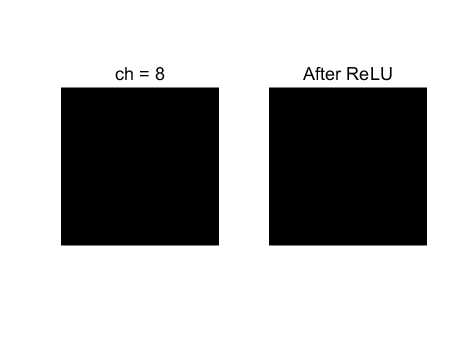
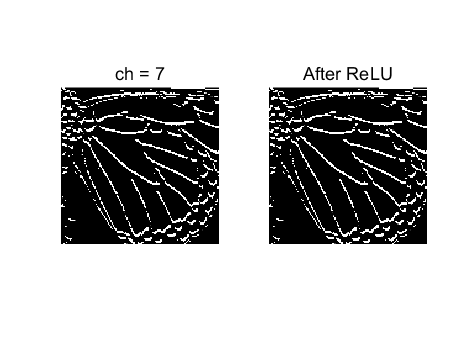
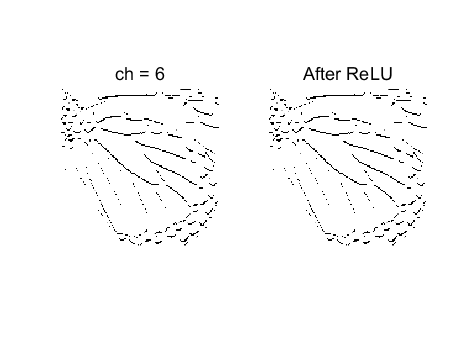
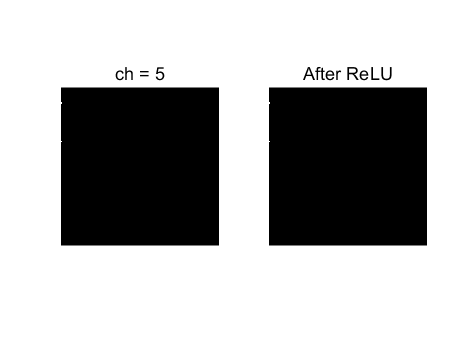
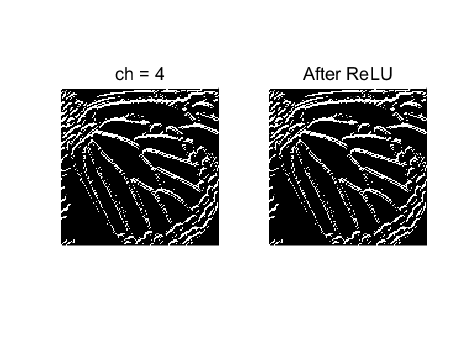
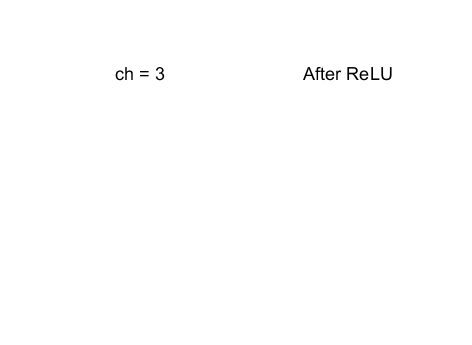
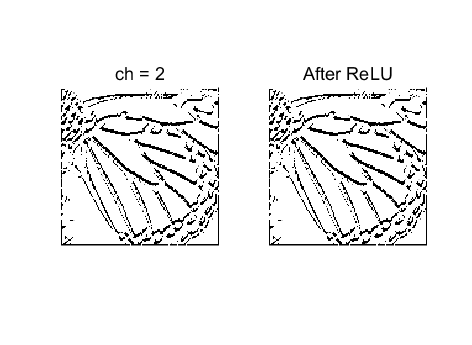
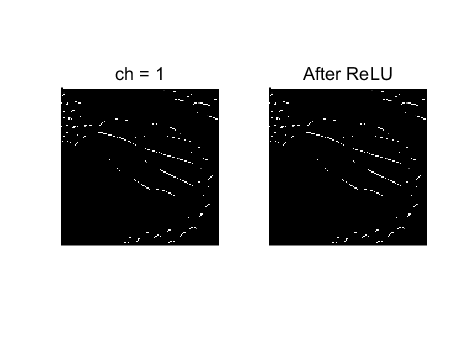
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Layer | Filter size | Number of input channels | Number of output channels | Input | Output | num\_ops |
| 1 | 3x3 | 1 | 16 | 128x128x1 | 128x128x16 | 9 |
| 2 | 3x3 | 16 | 16 | 128x128x16 | 128x128x16 | 144 |
| 3 | 3x3 | 16 | 4 | 128x128x16 | 128x128x4 | 144 |

Calculate the total number of multiplication operations.

**Solution 1 : Convolutional Layer**

****

****

****

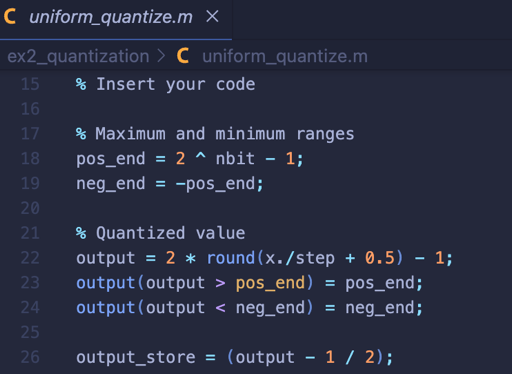
**Problem 2 (10p): Quantization**

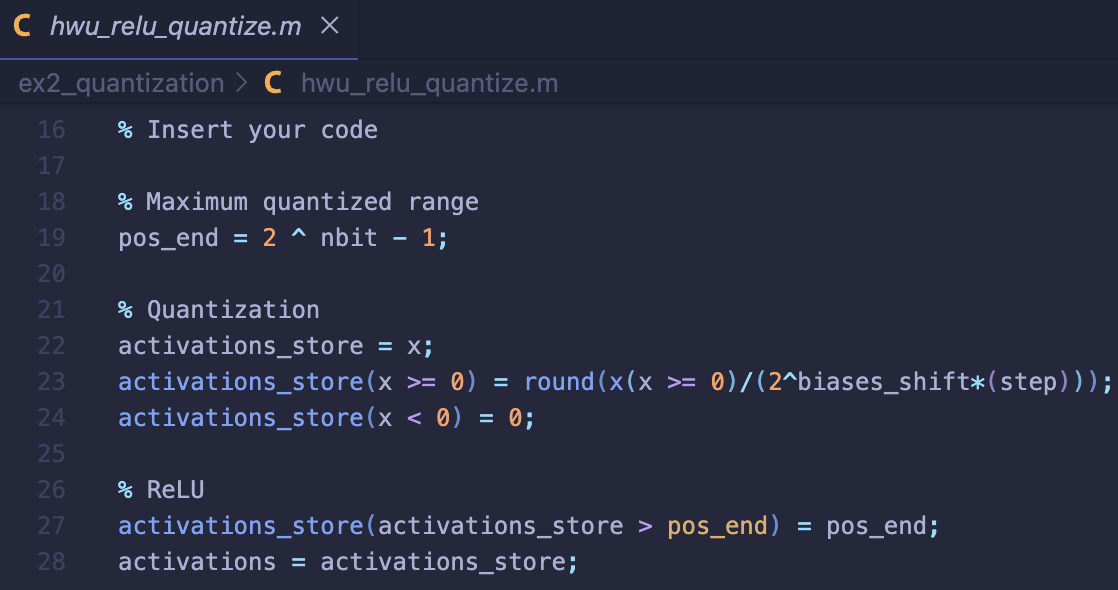
Implement a function to do weight and activation quantization in Matlab. Please see the description in the lecture note for details.

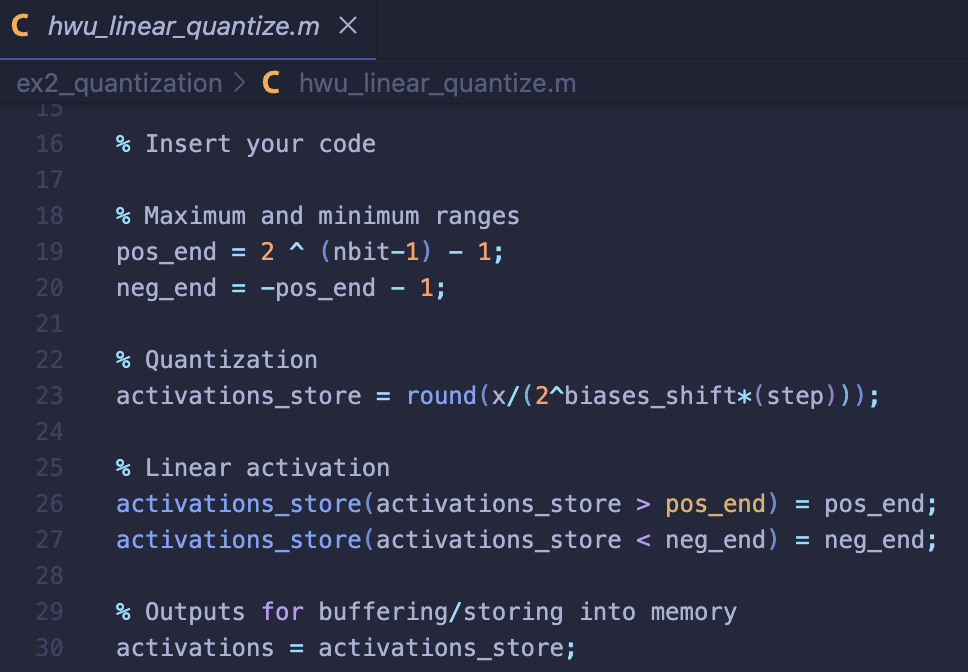
What you have to do:

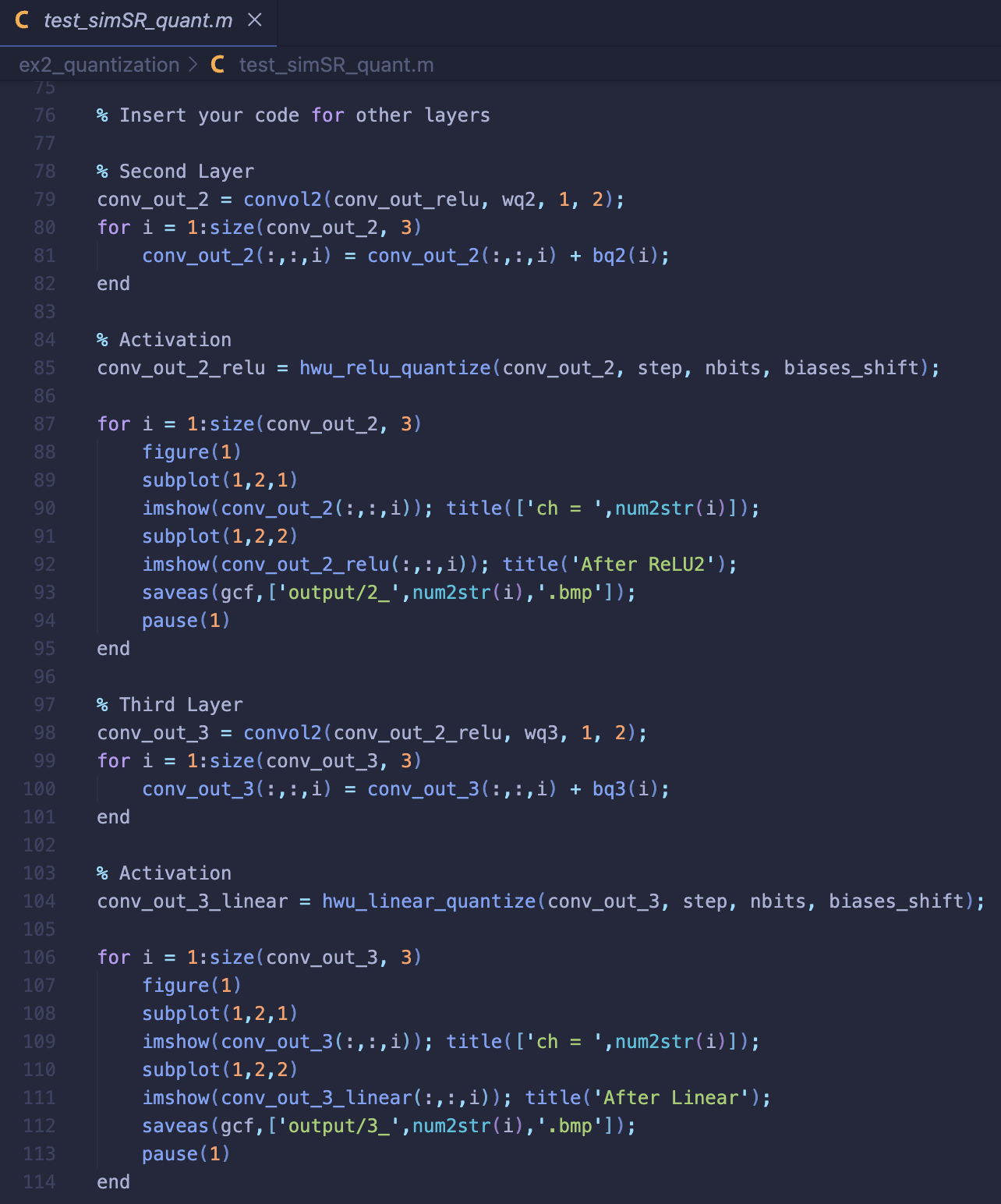
1. Reuse convol2.m in Problem 1.
2. Complete the missing codes in those files: uniform\_quantize.m, hwu\_relu\_quantize.v, hwu\_linear\_quantize and and test\_SR\_quant.m.
3. Output the feature maps of all three layers.

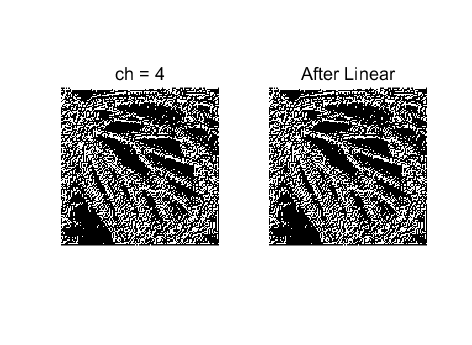
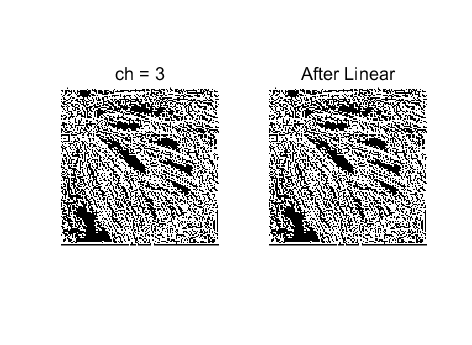
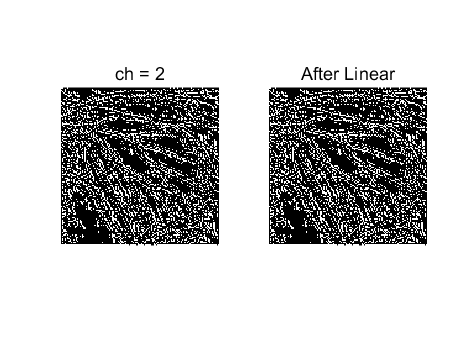
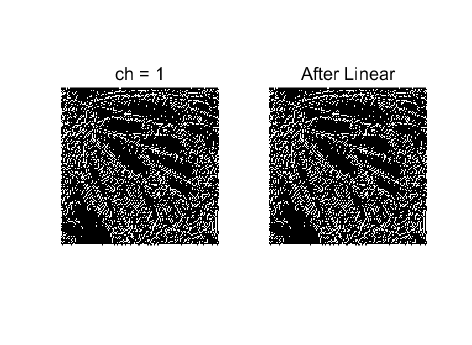
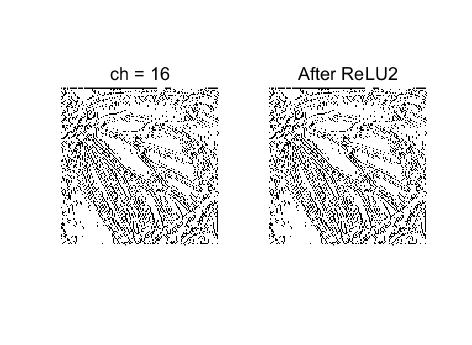
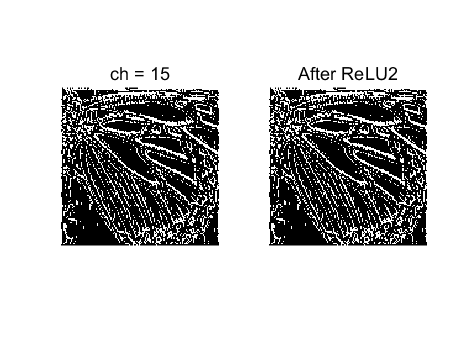
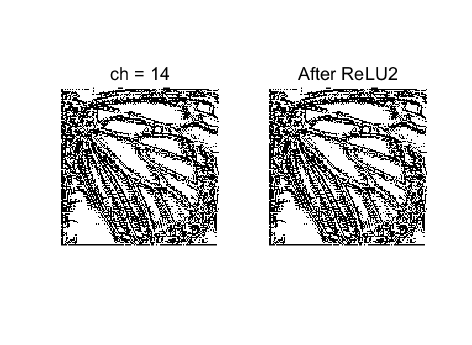
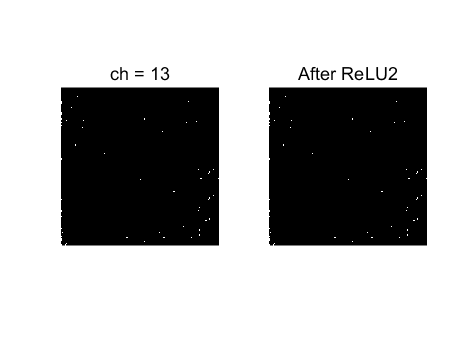
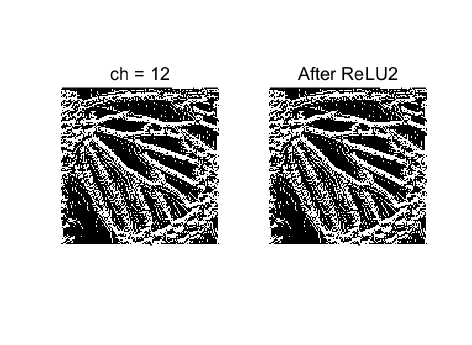
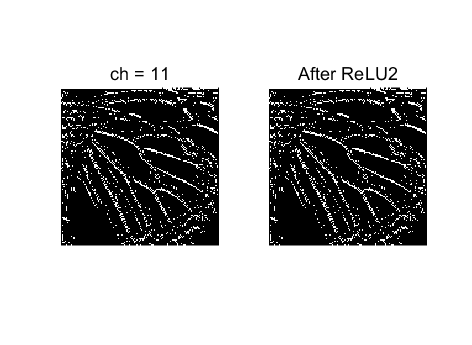
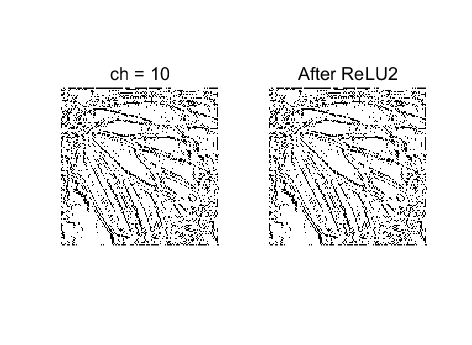
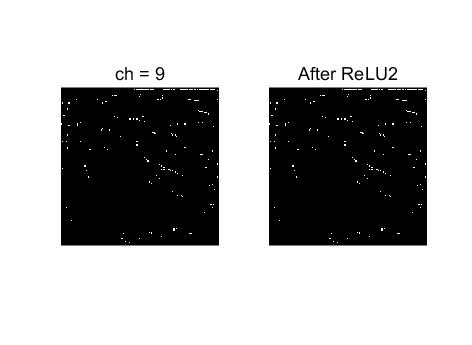
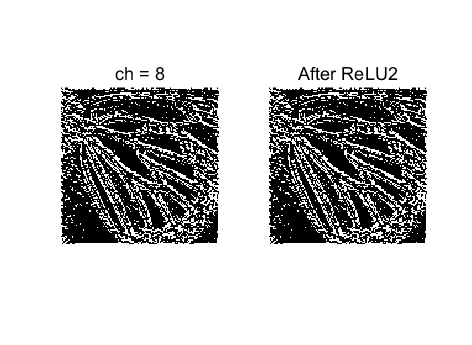
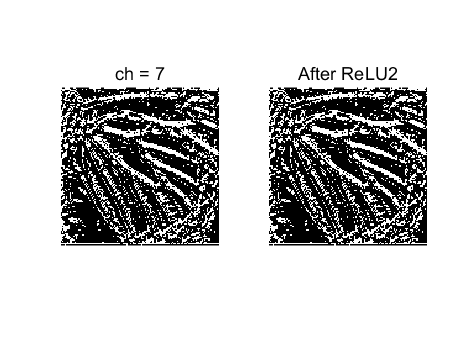
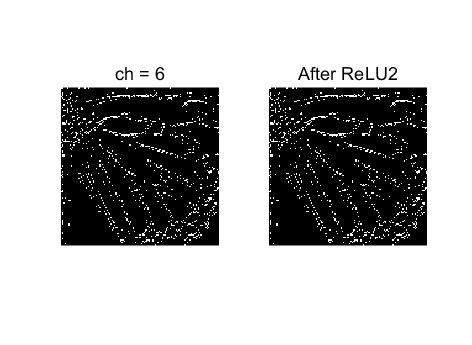
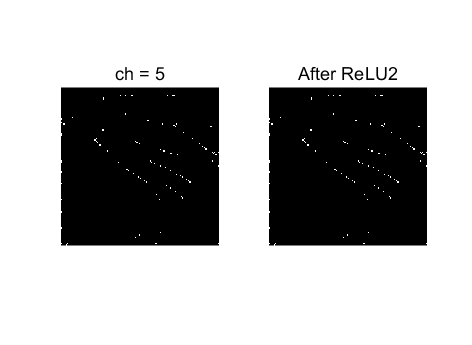
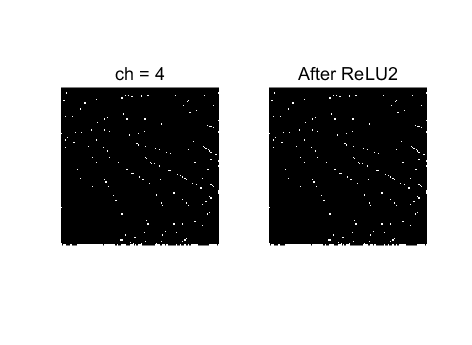
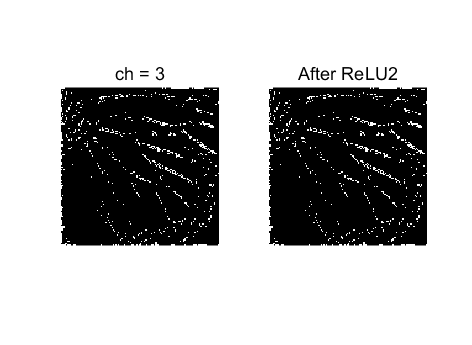
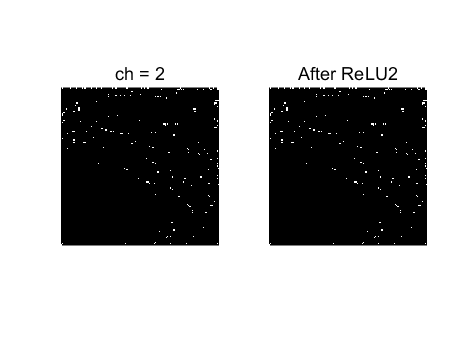
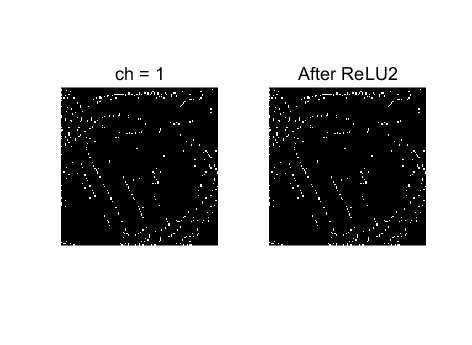
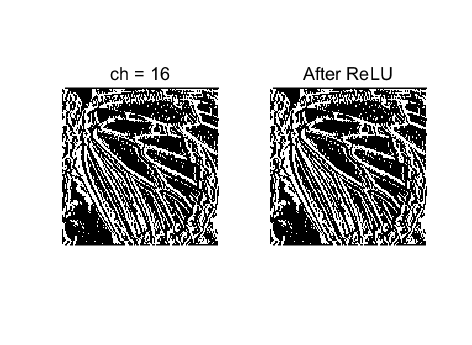
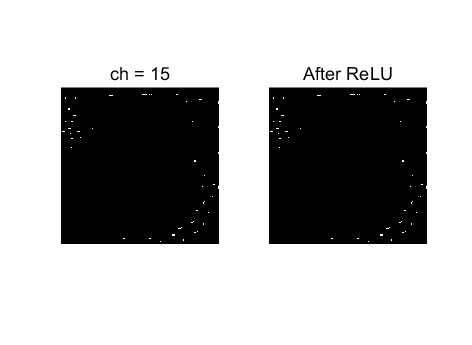
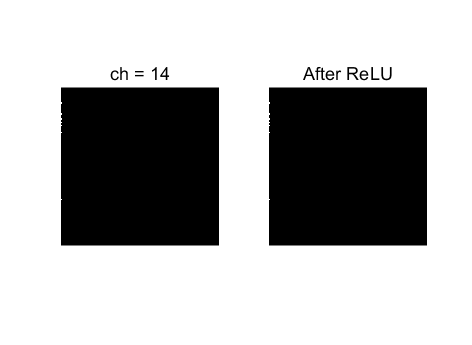
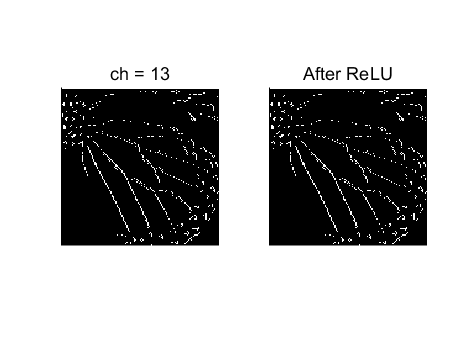
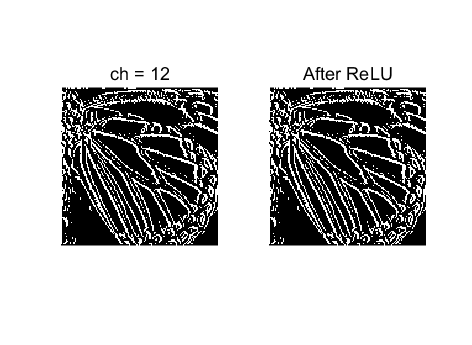
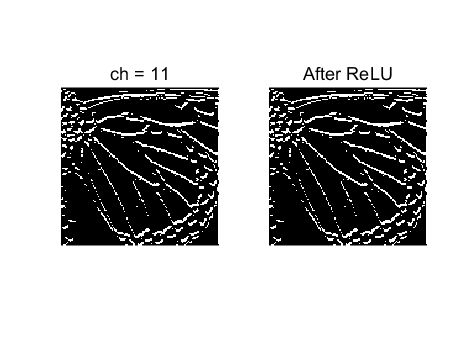
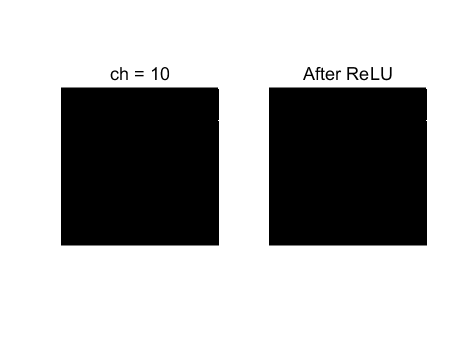
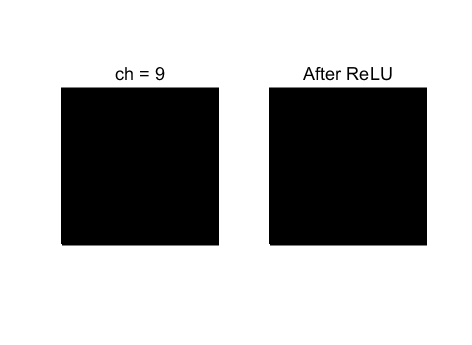
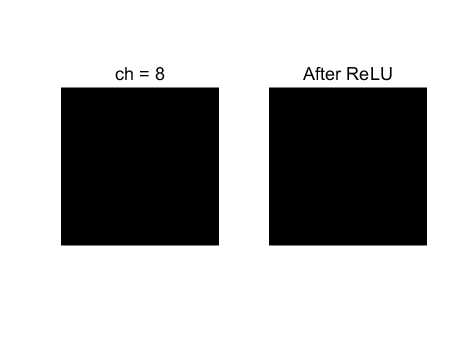
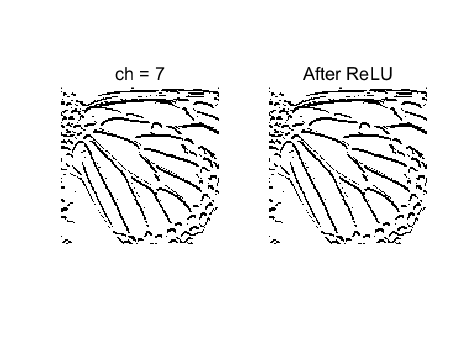
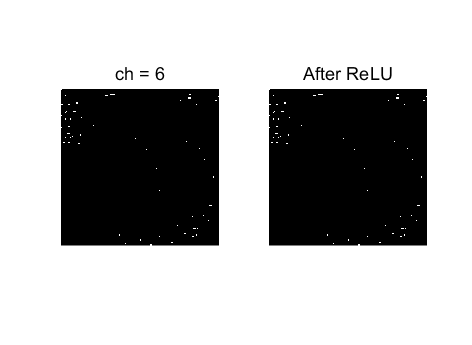
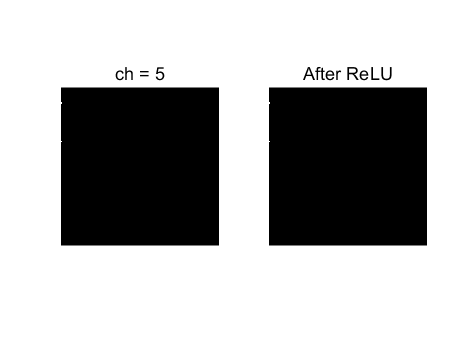
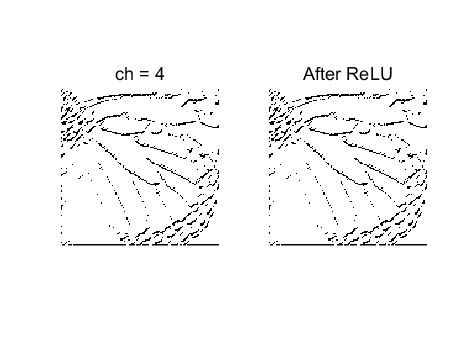
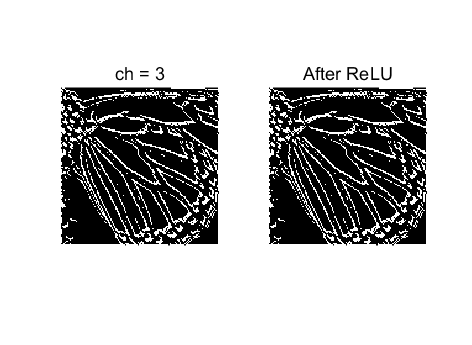
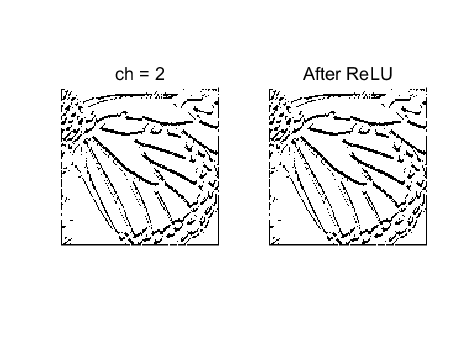
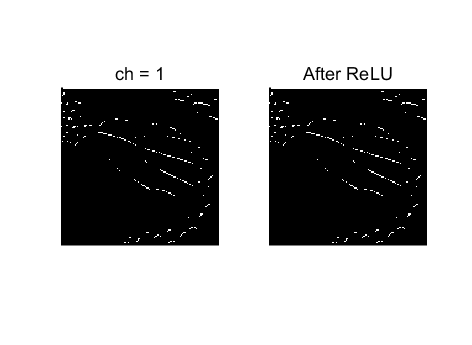
**Solution 2 : Quantization**

****

****

****

****

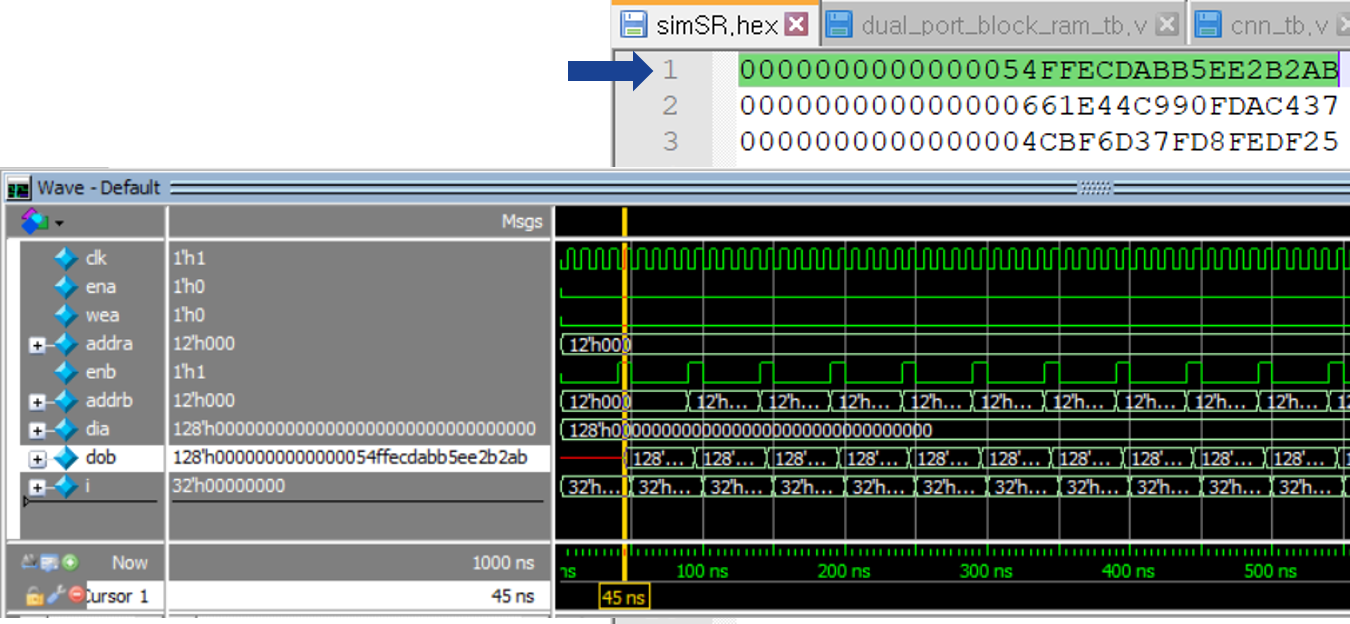
****

**Problem 3 (10p): Dual port block ram for CNN**

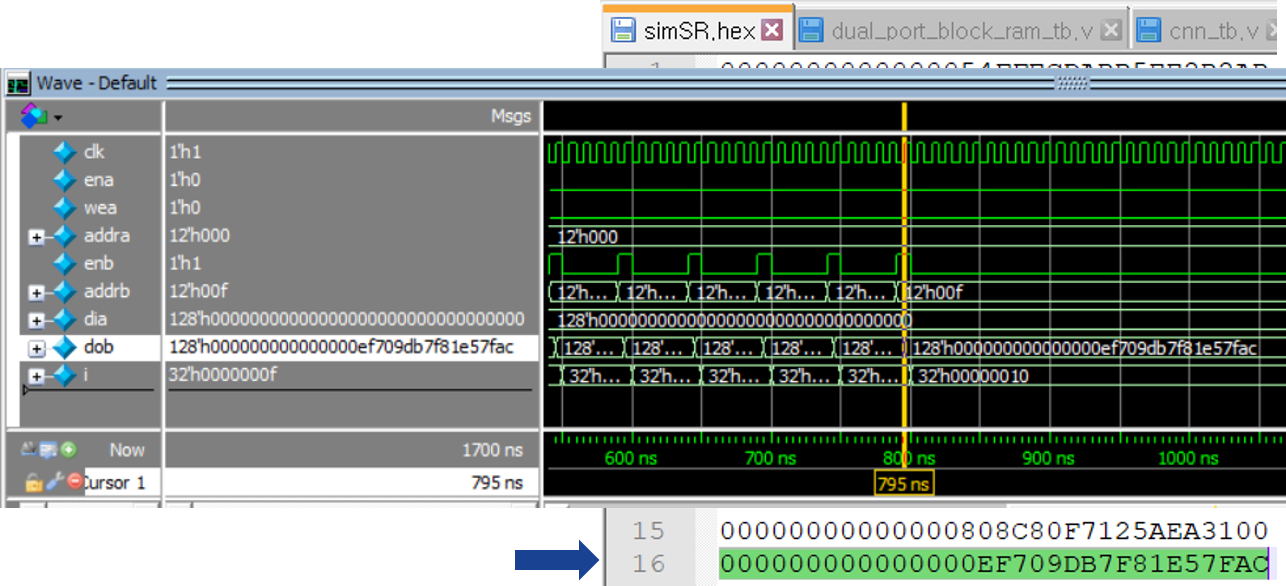
Implement a dual-port ram and its test bench. Please see the description in the lecture note for details.

What you have to do:

1. Complete the missing code in dual\_port\_block\_ram\_tb.v to load 16 conv. filters (simSR.hex).
2. Capture the simulation result.



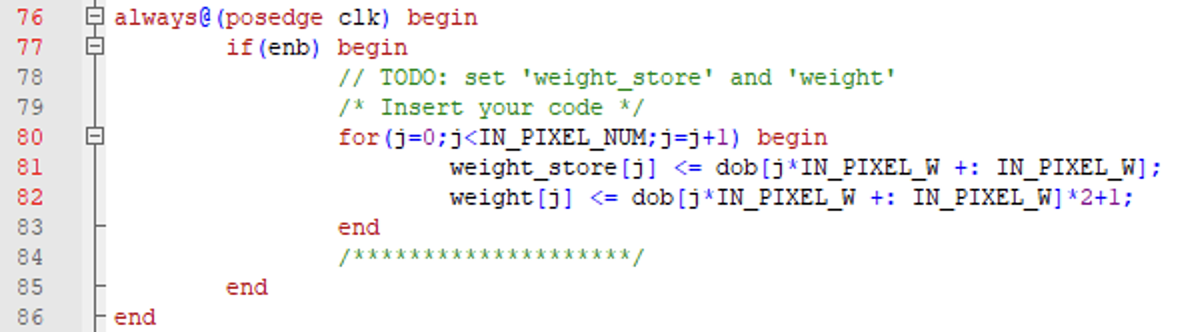
1. The first set of convolution filters

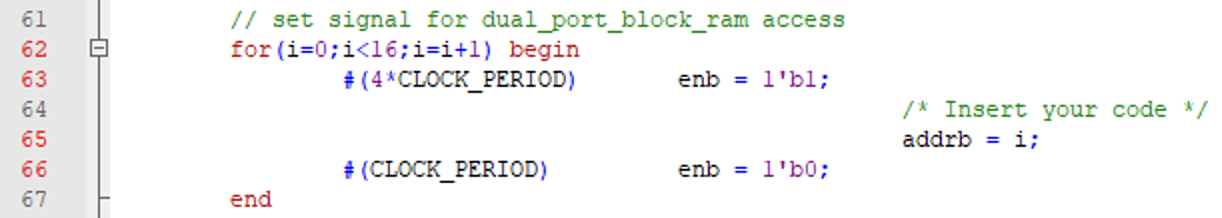


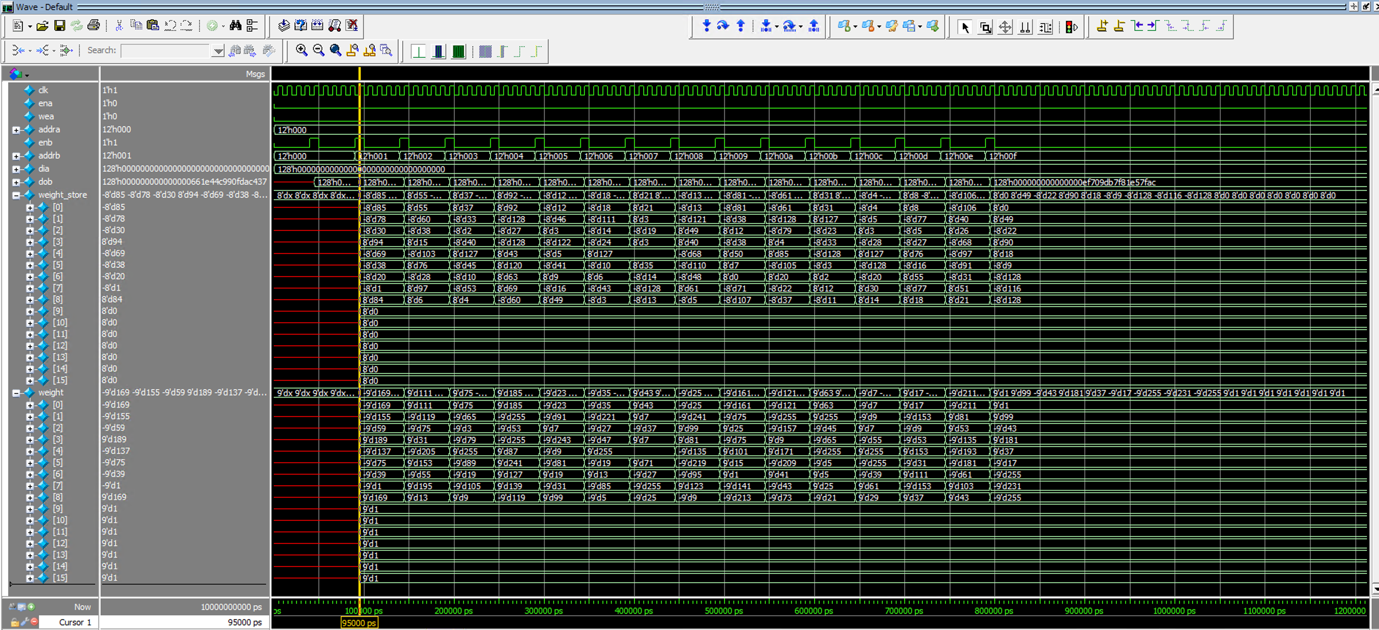
1. The 16th set of convolution filters

Fig. 3-1: Simulation results.

**Solution 3 : Dual port block ram for CNN**

****

****

****

**Problem 4: (Optional) SRNPU (2p)**

Reuse the codes in Problems 1 and 2 for the SRNPU. Fig. 4-1 shows the working procedure in a convolutional layer.

1. The weight is taken from the weight buffer (Line 230).
2. Do convolution (line 232).
3. Adding biases (line 236).
4. Do activation (line 244 and line 248).

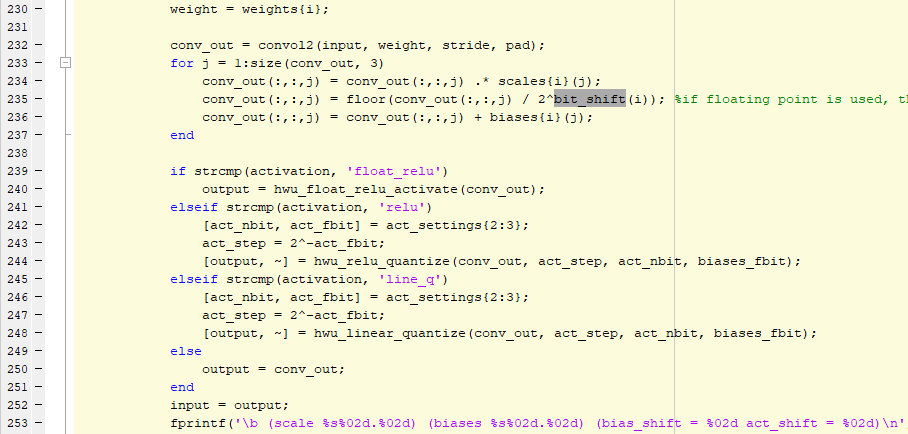


Fig.4-1: Matlab code for a convolution layer.

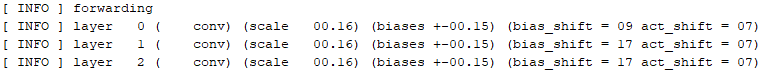


Fig.4-2: The simulation result on Command Window.

Fig. 4-2 shows the simulation result on Command Window. As shown in Line 253, bias\_shift and act\_shift are defined as follows:

* bias\_shift = bit\_shift{i}
* act\_shift = biases\_fshift – act\_fbit.

Explain the purposes of using bias\_shift and act\_shift.

**Solution 4 : SRNPU**

By using bias\_shift and act\_shift, we can determine the step size of quantization. Determining the step size helps the quantization with dividing the section where the numbers hold and mapping.