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

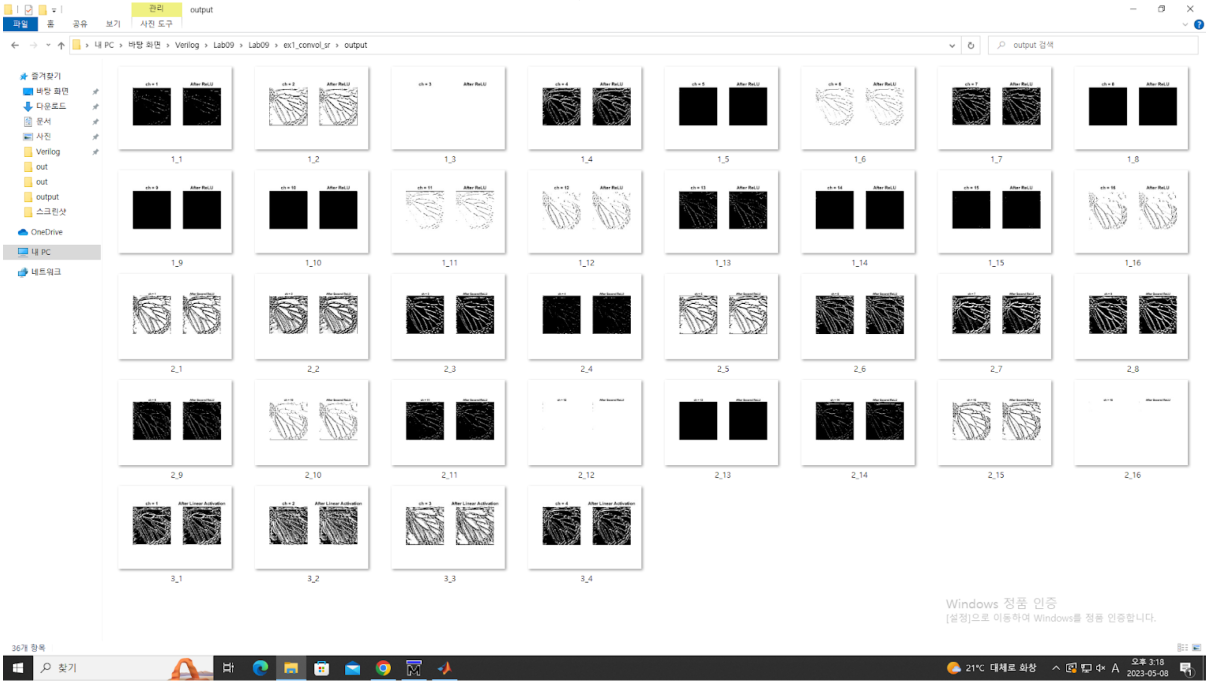
Satyam(2023-81784)

All the codes can be found in Code/ folder.

****

Problem 1 (10p): Convolutional Layer

a. Completed codes in convol2.m and test_SR.m
b. Output of the feature maps of all three layers.



c. Number of multiplication operations for a pixel:

Layer	Filter size	Number of input channels	Number of output channels	Input	Output	num_ops
1	3x3	1	16	128x128x1	128x128x16	3x3=9
2	3x3	16	16	128x128x16	128x128x16	3x3x16 = 144
3	3x3	16	4	128x128x16	128x128x4	3x3x16 = 144

Calculate the total number of multiplication operations.

Layer n: One pixel operations x output pixels

Layer 1: (3x3)x(128x128x16) = 2359296

Layer 2: (3x3)x(128x128x16) = 37748736

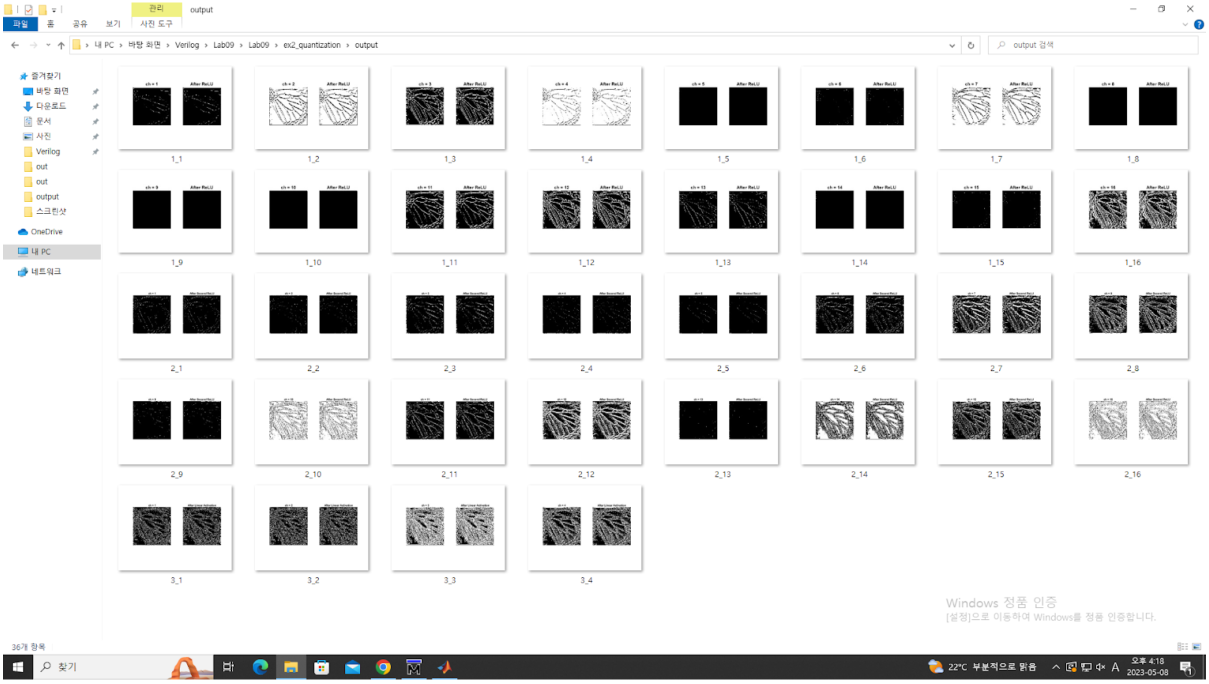
Layer 1: (3x3)x(128x128x16) = 9437184

Therefore, Total number of multiplication operations= 49545216

Problem 2 (10p): Quantization

a. Completed code in uniform_quantize.m, hwu_relu_quantize.v, hwu_linear_quantize and and test_SR_quant.m.

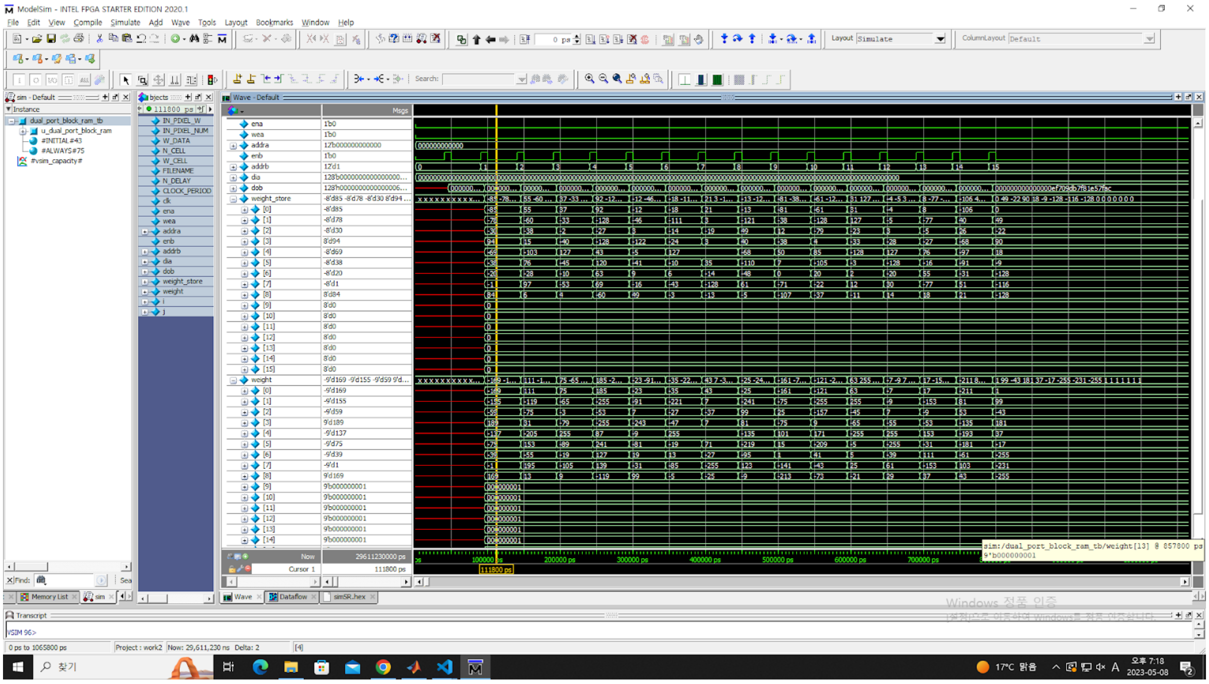
b. Output the feature maps of all three layers.



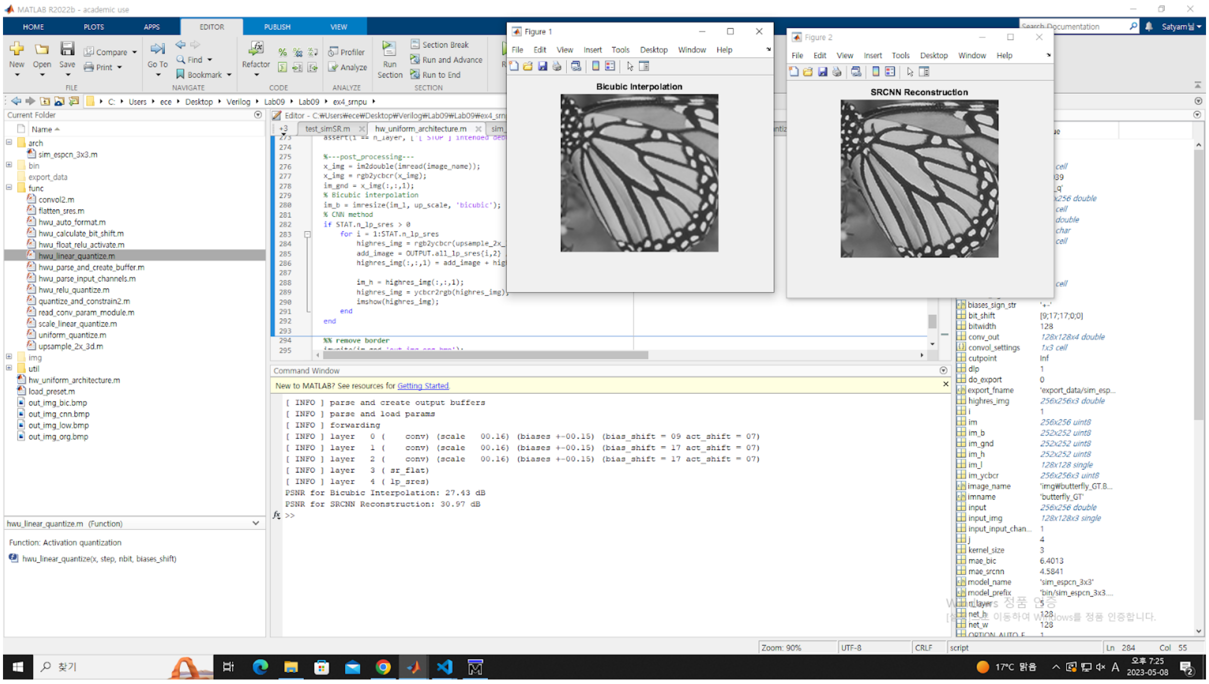
Problem 3 (10p): Dual port block ram for CNN

a) Completed code in dual_port_block_ram_tb.v to load 16 conv. filters (simSR.hex).

b) Capture the simulation result.



Problem 4: (Optional) SRNPU (2p)



```
%---post_processing---
x_img = im2double(imread(image_name));
x_img = rgb2ycbcr(x_img);
im_gnd = x_img(:,:,1);
% Bicubic interpolation
im_b = imresize(im_l, up_scale, 'bicubic');
% CNN method
if STAT.n_lp_sres > 0
    for i = 1:STAT.n_lp_sres
        highres_img = rgb2ycbcr(upsample_2x_
        add_image = OUTPUT.all_lp_sres{i,2}
        highres_img(:,:,1) = add_image + hig

        im_h = highres_img(:,:,1);
        highres_img = ycbcr2rgb(highres_img);
        imshow(highres_img);
    end
end

%% remove border
```

```
[ INFO ] parse and create output buffers
[ INFO ] parse and load params
[ INFO ] forwarding
[ INFO ] layer   0 (    conv) (scale   00.16) (biases +-00.15) (bias_shift = 09 act_shift = 07)
[ INFO ] layer   1 (    conv) (scale   00.16) (biases +-00.15) (bias_shift = 17 act_shift = 07)
[ INFO ] layer   2 (    conv) (scale   00.16) (biases +-00.15) (bias_shift = 17 act_shift = 07)
[ INFO ] layer   3 ( sr_flat)
[ INFO ] layer   4 ( lp_sres)
PSNR for Bicubic Interpolation: 27.43 dB
PSNR for SRCNN Reconstruction: 30.97 dB
>>
```

Explain the purposes of using bias_shift and act_shift.

bias_shift - the amount of decimal moved in the output of the convolutional bias.

act_shift - the difference between the bias's decimal part and activation's decimal part.