

---

# Development of C programs for Convolutional Neural Network Accelerators

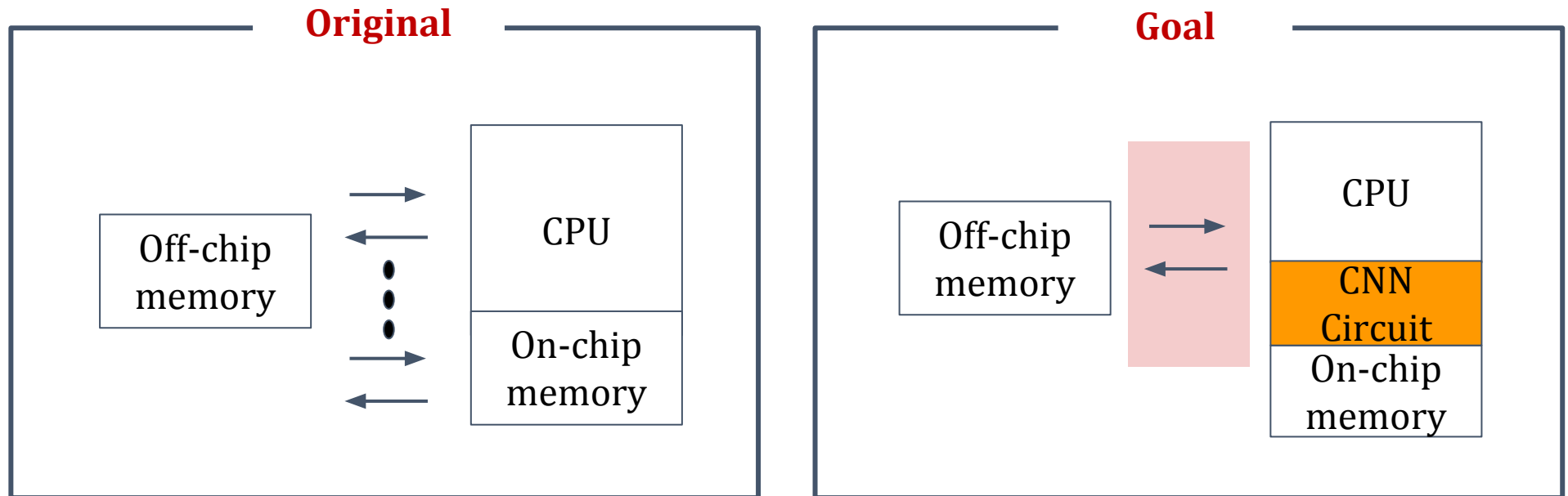
JiYoung An, Sujin Kang  
Prof. Nikil Dutt , Kenshu Seto, Hamid Nejatollahi

Dept. of Computer Engineering in Kyung Hee ,University  
Dept. of Computer Engineering in Hanyang University  
University of California Irvine  
University in Tokyo

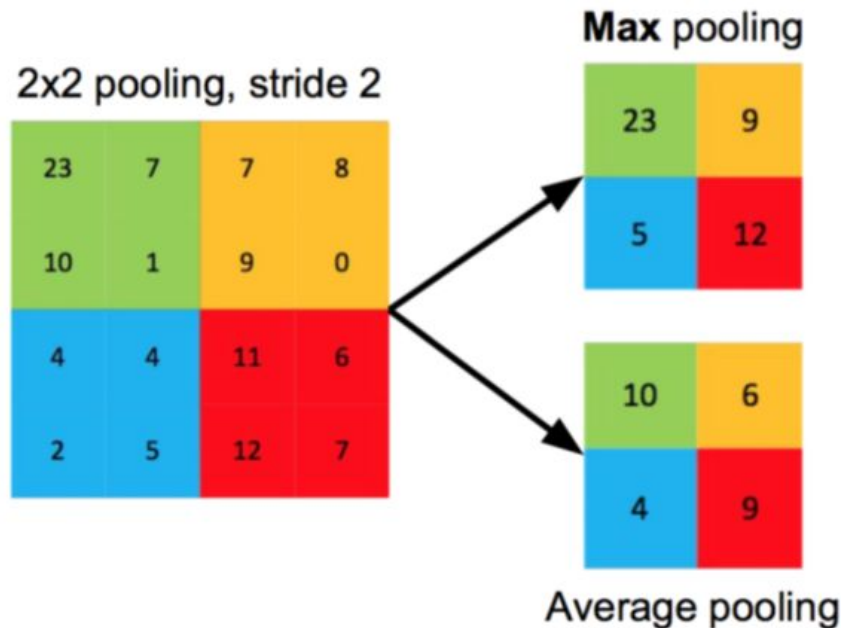
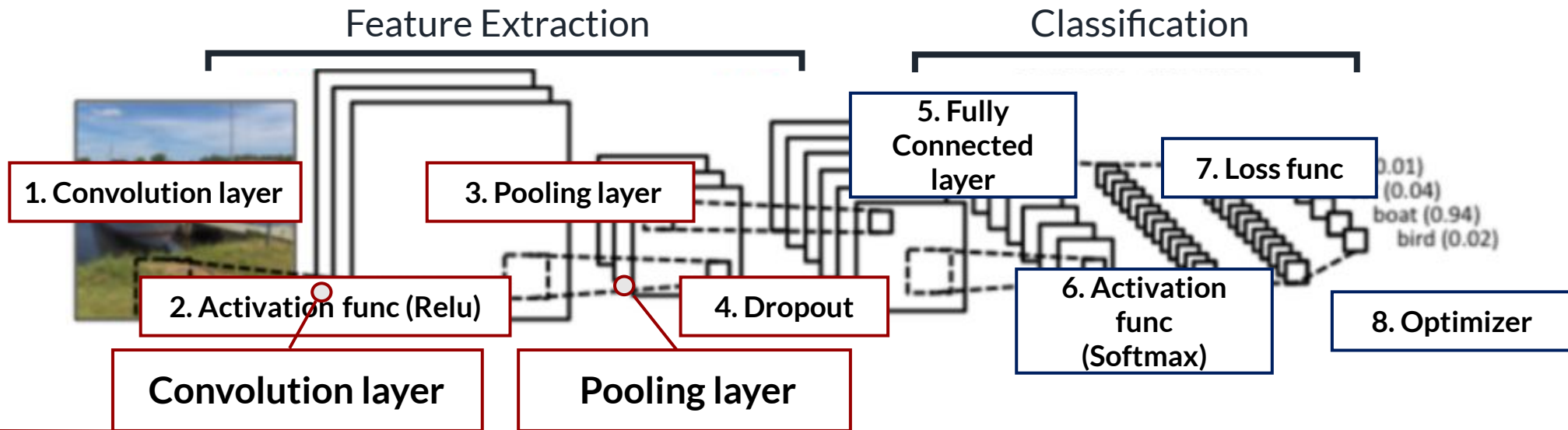
# 1. Project

## Acceleration of Convolution Neural Networks to embed in light machine

Focus on Reducing Memory access time

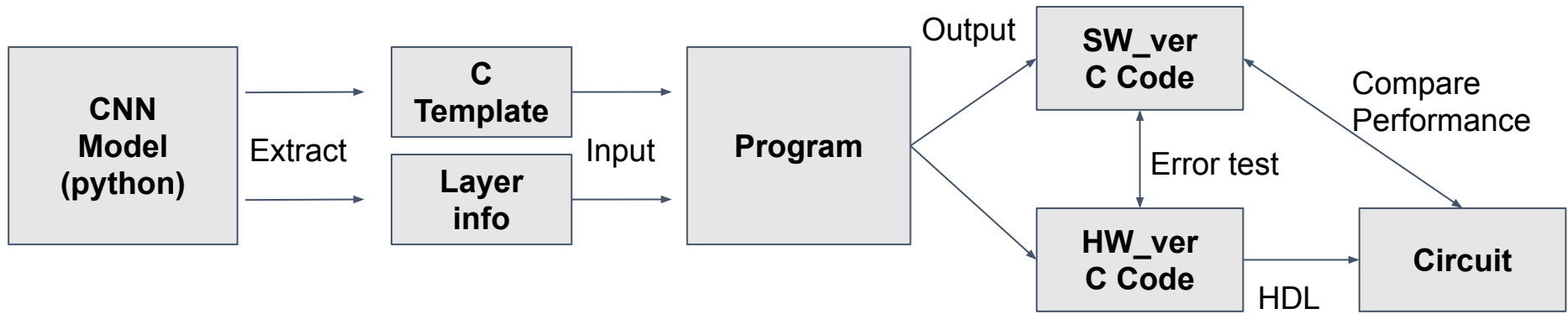


## 2. Background

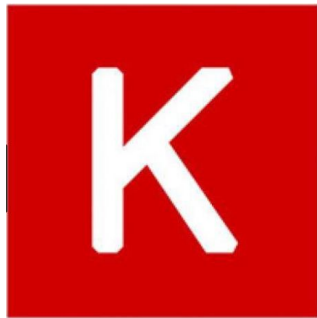


### 3. Work Process and Tool

#### 1) Work process



#### 2) Environment and Tool



# 4. Work Progress

## 1) Previous Work

- Understanding the concept of CNN , HDL and FPGA
- Install Vivado HLS
- Template for VGG 16
- Converter Program

name	layer_type	batch	inputbatch_outfilters	kernel_sizeactivation	padding	strides	post_size	units	dtype	trainable	use_bias	bias_regularizer	name
input_1	InputLayer	None	224/None, 224, 224, 3										
block1_conv2D	Conv2D	None	224/None, 224, 64(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_1
block1_pooling2D	MaxPooling2D	None	224/None, 224, 64(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_1
block2_conv2D	Conv2D	None	112/None, 112, 128(3, 3)	relu	valid	(2, 2)	(32, 32)			TRUE	TRUE		class_2
block2_pooling2D	MaxPooling2D	None	112/None, 112, 128(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_2
block3_conv2D	Conv2D	None	56/None, 56, 256(3, 3)	relu	valid	(2, 2)	(32, 32)			TRUE	TRUE		class_3
block3_pooling2D	MaxPooling2D	None	56/None, 56, 256(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_3
block4_conv2D	Conv2D	None	28/None, 28, 512(3, 3)	relu	valid	(2, 2)	(32, 32)			TRUE	TRUE		class_4
block4_pooling2D	MaxPooling2D	None	28/None, 28, 512(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_4
block5_conv2D	Conv2D	None	14/None, 14, 512(3, 3)	relu	valid	(2, 2)	(32, 32)			TRUE	TRUE		class_5
block5_pooling2D	MaxPooling2D	None	14/None, 14, 512(3, 3)	relu	same	(1, 1)				TRUE	TRUE		class_5
flatten	Flatten	None	7/None, 25088							TRUE	TRUE		class_6
dense	Dense	None	250/None, 4096	relu				4096		TRUE	TRUE		class_7
predictionsDense	Dense	None	4096/None, 1000	softmax				1000		TRUE	TRUE		class_8

Template.txt

vgg16.csv

```
import csv
import argparse
from string import Template

parser = argparse.ArgumentParser()
parser.add_argument('--input', default='', type=str,
                    help='The filename of image to be completed,')
parser.add_argument('--weight', default='', type=str,
                    help='The filename of weight')
parser.add_argument('--output', default='output.txt', type=str,
                    help='Where to write output,')

if __name__ == '__main__':
    #open the Template
    maxpooling2D_sw = open('MaxPooling2D_SW.txt')
    maxpooling2D_hw = open('MaxPooling2D_HW.txt')
    conv2D_sw = open('Conv2D_SW.txt')
    conv2D_hw = open('Conv2D_HW.txt')
    flatten_sw = open('Flatten_SW.txt')
    dense_sw = open('Dense_SW.txt')
    main_f = open('main.txt')
    #read it
    maxpooling2D_SW = Template(maxpooling2D_sw.read())
    maxpooling2D_HW = Template(maxpooling2D_hw.read())
    conv2D_SW = Template(conv2D_sw.read())
    conv2D_HW = Template(conv2D_hw.read())
    flatten_SW = Template(flatten_sw.read())
    dense_SW = Template(dense_sw.read())
    main_t = Template(main_f.read())

    #open csv
    csv_file = open('vgg16.csv')
    csv_reader = csv.DictReader(csv_file)
    # Make variables of functions
    SW_def_func = ""
    HW_def_func = ""
    SW_functions = ""
    HW_functions = ""
```

code.py

## 2) Future Work

- Extend Template to various models - VGG19, RasNet 50, MobileNet
- Test Code
- Build Circuit
- Compare Performance between HW and SW version

```
1 #include <stdio.h>
2 typedef int DATA_T;
3
4 void SW_Conv2D_padding_act_relu_block1_conv1(DATA_T I[3][224][224], DATA_T W[64][3][3], DATA_T B[64], DATA_T O[64][224][224], int M, int C, int R, int S, int E, int F, int U)
5 {
6     int m, x, y, i, j, k;
7     DATA_T itm, ofm;
8     for (m=0; m<M; m++) {
9         for (x=0; x<E; x++) {
10             for (y=0; y<F; y++) {
11                 ofm = B[m];
12                 for (k=0; k<C; k++) {
13                     for (i=0; i<R; i++) {
14                         for (j=0; j<S; j++) {
15                             if (xi < E && yj < F) {
16                                 itm = I[x][y][i][j];
17                                 if (itm < 0) itm = 0;
18                             }
19                             ofm = ofm + itm * W[m][k][i][j];
20                         }
21                     }
22                 }
23                 if (ofm < 0) { // relu activation
24                     ofm = 0;
25                 }
26                 O[m][x][y] = ofm;
27             }
28         }
29     }
30 }
31
32 void SW_Conv2D_padding_act_relu_block1_conv2(DATA_T I[64][224][224], DATA_T W[64][3][3], DATA_T B[64], DATA_T O[64][224][224], int M, int C, int R, int S, int E, int F, int U)
33 {
34     int m, x, y, i, j, k;
35     DATA_T itm, ofm;
36     for (m=0; m<M; m++) {
37         for (x=0; x<E; x++) {
38             for (y=0; y<F; y++) {
39                 ofm = B[m];
40                 for (k=0; k<C; k++) {
```

vgg16.c

---

# Question

---