Ho Chi Minh City University of Technology


Faculty of Computer Science and Engineering
Course: Computer Architecture Lab (CO2008)

# Assignment

## CONVOLUTION OPERATION

Ho Chi Minh City, September 2024

# Contents

# 1 Outcomes

After finishing this assignment, students can proficiently use:

- MARS MIPS simulator.
- Arithmetic & data transfer instructions.
- Conditional branch and unconditional jump instructions.
- Procedures.

# 2 Introduction

In this modern day and age, artificial intelligence has been developing exponentially fast. Rapid improvements in technique and data lead to a remarkable increase in calculations. Its sub-categories, machine learning and deep learning, have reached a point where billions of calculations must be done to compute the millions of parameters in a model.

Convolutional neural network (CNN) is one of the most widely used type of network used in deep learning. Its usage can be seen in a wide range of applications, particularly in fields that involve image and video analysis. However advanced a CNN model maybe, it always stems from matrix computations and most notably, convolution operations.

The convolution operation involves sliding a filter, also known as a kernel, across the input image. This filter is a small matrix of weights that is applied to a local region of the input matrix, producing a single value in the output feature map. The process is repeated across the entire matrix, allowing the network to learn various features of the matrix. In the case of image analysis, these features can include edges, textures, and patterns. The mathematical operation performed is essentially a dot product between the filter and the receptive field of the input image.

The process involves taking a small matrix, called a kernel or filter, and sliding it over an input matrix, such as an image. At each position, the dot product of the kernel and the overlapping region of the input matrix is computed, and the result is stored in the output matrix. This process is repeated for all positions of the kernel over the input matrix, effectively blending the kernel with the input to highlight specific features like edges or textures. In advance,


Figure 1: An example of convolution operation

In Figure 1, a 5x5 input matrix and a 3x3 kernel filter have been provided. We first match the kernel onto a corner of the input, in this case the small 3x3 matrix covered in a large red square. We then calculate the sum of element-wise products of both matrices to get a single value. This value will be the first value of the output matrix. Proceed to slide the kernel from left to right, then up to down to cover every square of the input to get the final result. Notice that the stride here is 1 as the kernel move 1 column and 1 row at a time.

One tricky issue when applying convolutional layers is that we tend to lose pixels on the perimeter of our image. One straightforward solution to this problem is to add extra pixels of filler around the boundary of our input image, thus increasing the effective size of the image. Typically, we set the values of the extra pixels to zero.

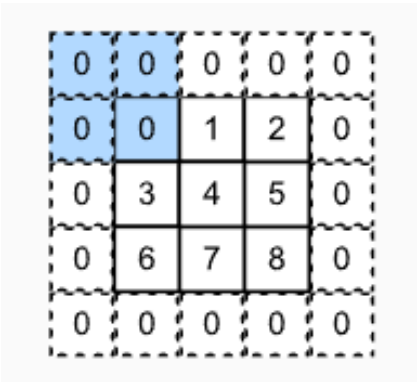

Figure 2: An example of matrix with zero-padding

In Figure 2, applying the padding with value equal to 1 into 3x3 input matrix will increase its size to 5x5.

# 3 Requirements

In this assignment, your task is to perform convolution operations using MIPS assembly. The requirements are listed in the following sections.

## 3.1 Input

The program should be able to receive input from an external input file. It should be named **input_matrix.txt**. The content is numbers separated by spaces. Both image matrix and kernel matrix are square matrix mxm and nxn, respectively. Additionally, **input_matrix.txt** contain 3 rows, the first row will include 4 values which are:

- **N**: The size of the image matrix. $(3 \leq N \leq 7)$
- **M**: The size of the kernel matrix. $(2 \leq M \leq 4)$
- **p**: The value of padding. $(0 \leq p \leq 4)$
- **s**: The value of stride. $(1 \leq s \leq 3)$

Thus, the contents of image and kernel matrix will be in the second and third row, in turn.

For simplicity, the stride will be the same for both dimensions while symmetrical padding is applied.

All of the matrices content are **floating-point numbers** while the stride and padding are **integer numbers**.

## 3.2 Output

A text file named **output_matrix.txt** contains the convolution matrix result, also numbers separated by spaces.

## 3.3 Pre-determined variables

Some variables must be defined as follows for grading:

- **image** (word): Where the image matrix is saved.
- **kernel** (word): Where the kernel matrix is saved.
- **out** (word): Where the output matrix is saved.

### 3.4 Test cases

You are provided a set of example test cases along with this file. It is also crucial to remember that the result of such test cases will not be included. You are advised to develop a separated program in high-level programming languages you have learned so far to test the MIPS program yourself.

### 3.5 Report

A report about your work should be made for this assignment. Images of test runs as well as description of the functions and logic are anticipated. Flow charts and other visualization of your ideas are also encouraged. The report must contain basic information such as name, ID, class, and subject.

## 4 Submission

Students are requested to submit the MIPS program(s)/source code (.asm files) and the Assignment report to BK E-learning system (LMS) right after the last lab session of your class. Assignment must be done individually.

Students who do not submit on time will get 0 for the assignment.

The report should not contain code. Instead, students should present the algorithms as well as the idea in your implementation.

There will be NO deadline extensions.

## 5 Plagiarism

Similarity less than 30% in MIPS code is allowed. In other words, you will get 0 for assignment if your answers are similar to another student's more than 30%. Note that, we will use the MOSS tool developed by Stanford for checking similarity (https://theory.stanford.edu/~aiken/moss/).

## 6 Rubric for evaluation

Your grade will base on the number of test cases you got right.

You will receive a 0 if you fail to submit on time the source code and report.