CPEG 422/622
EMBEDDED SYSTEMS DESIGN

Chengmo Yang

chengmo@udel.edu

Evans 201C

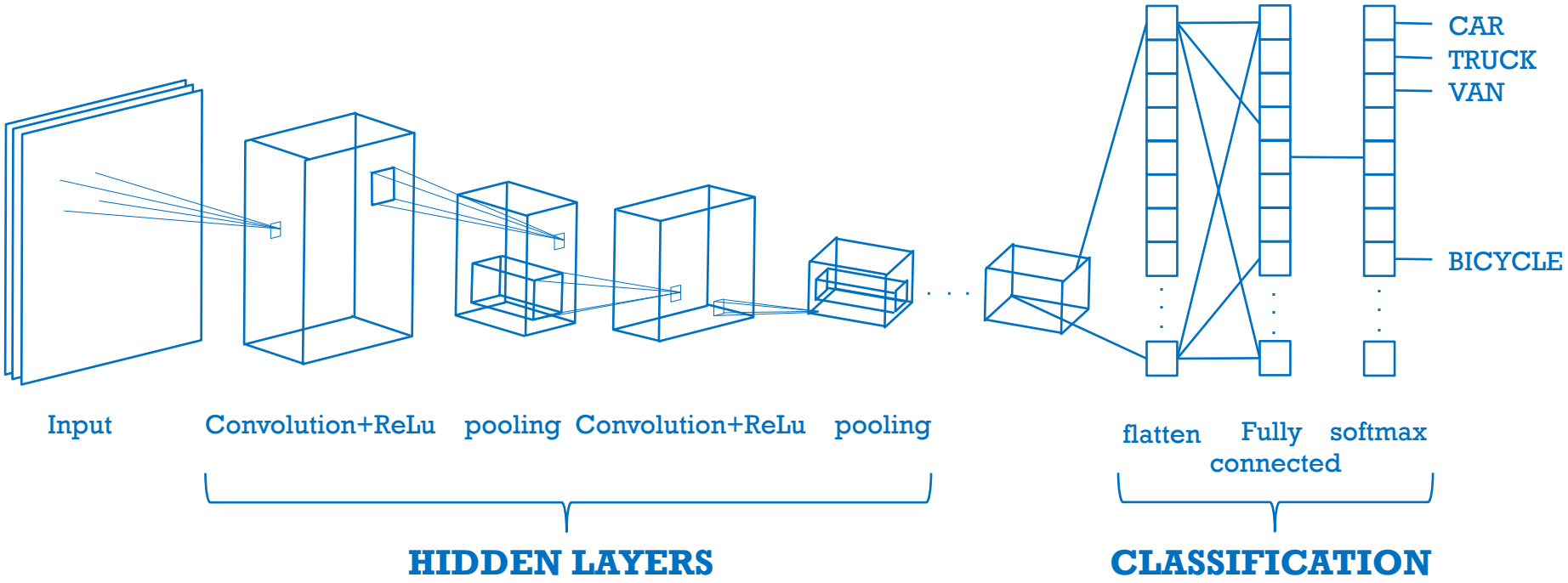
LECTURE 15
PROJECT 4

OUTLINE

- Convolution operation introduction
- Project 4

CONVOLUTION AND POOLING OPERATION

Architecture of CNN:



CONVOLUTIONAL OPERATION

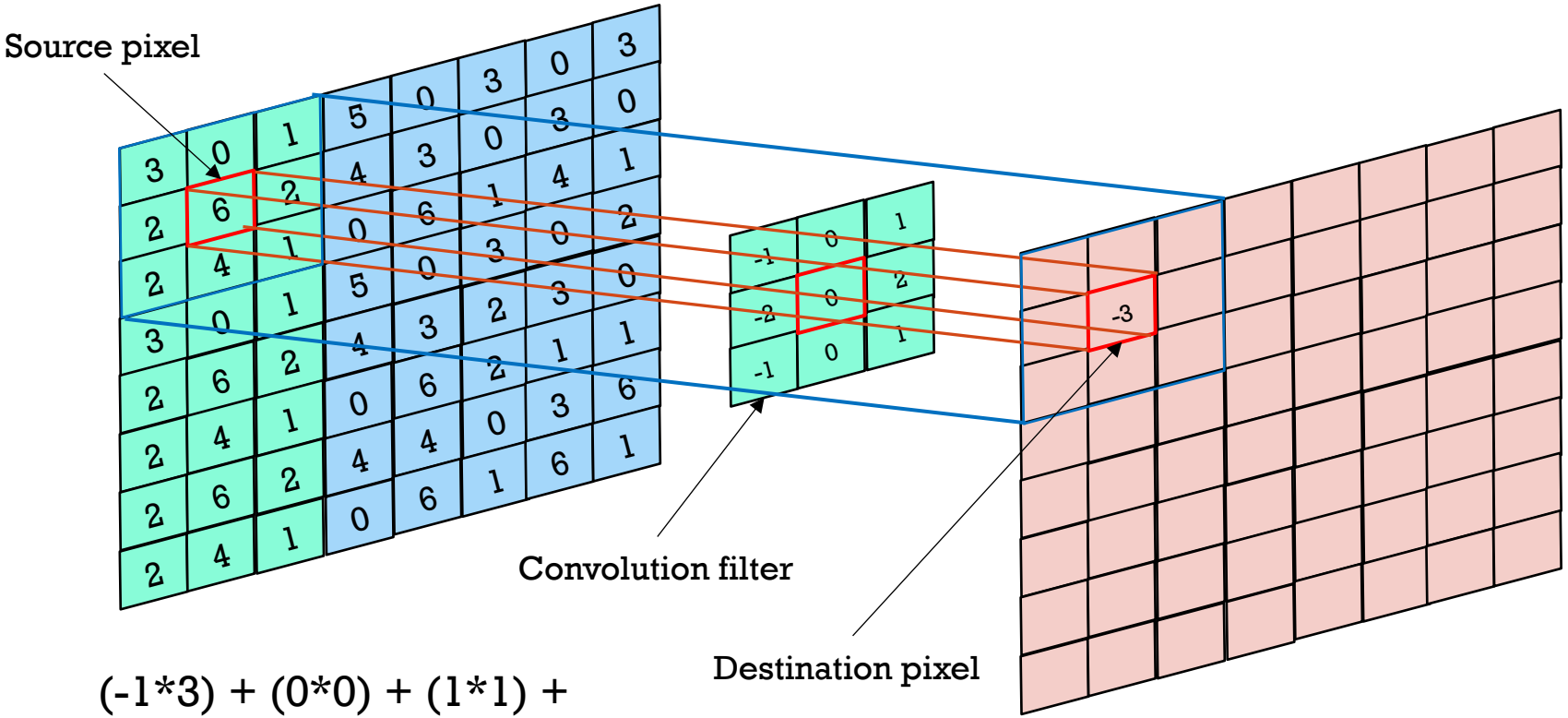
- Assume A is a 3x3 filter and B is an **n × n** matrix. The output of a convolution operation is C=A*B, which C is an **(n-2) × (n-2)** matrix, is defined as:

$$C=\left[\begin{bmatrix} c_{11} & \cdots & c_{1(n-2)} \\ \vdots & \ddots & \vdots \\ c_{(n-2)1} & \cdots & c_{(n-2)(n-2)} \end{bmatrix}\right], c_{ij} = \sum_{x=i}^{i+2} \sum_{y=j}^{j+2} a_{xy} b_{xy}$$

- In project 4, each element of A and B is 16-bit long, while each element of C is 32-bit long. The Multiply-Accumulate (MAC) unit can be used as the basic function unit to implement matrix multiplication. FPGA usually utilizes DSP units to perform MAC operations.

CONVOLUTION OPERATION

Convolution operation:

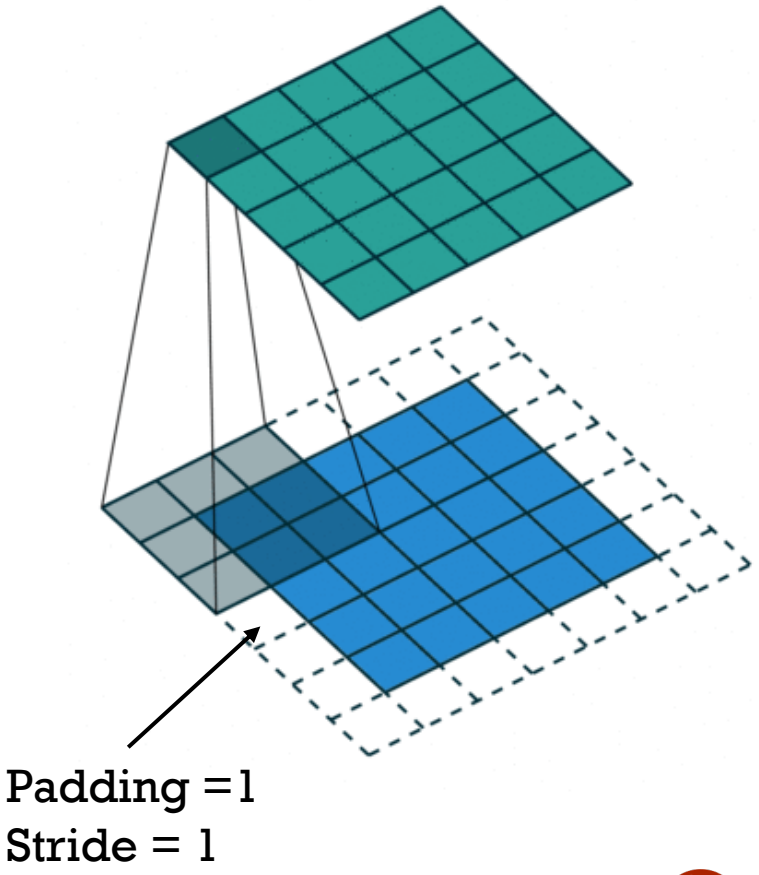


(-1*3) + (0*0) + (1*1) +
(-2*2) + (0*6) + (2*2) +
(-1*2) + (0*4) + (1*1) = -3

The size of destination output should be 6*6 but with padding, the output can achieve its original size 8*8.

CONVOLUTION AND POOLING OPERATION

Convolution operation with padding:

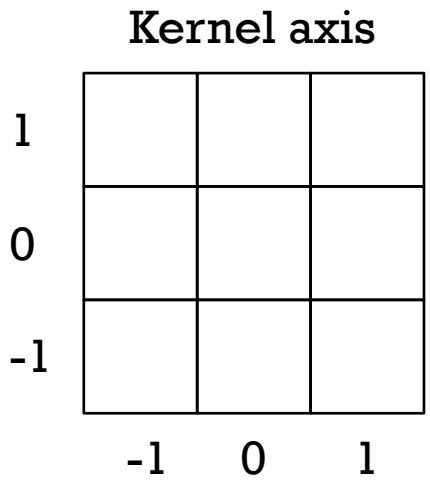


CONVOLUTION OPERATION

Convolution operation with padding:

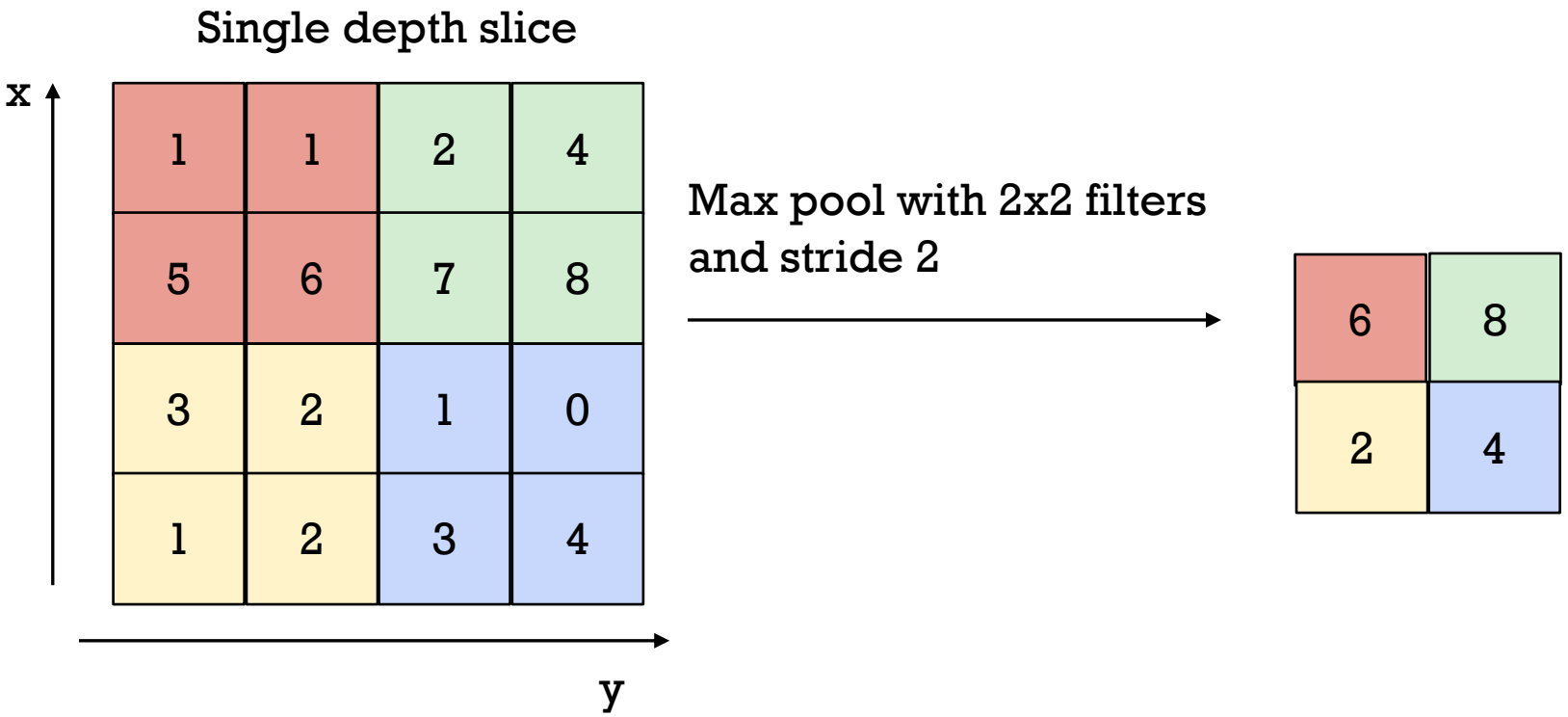
Pseudo code for the convolution of an image f(x,y) with a kernel k(x,y) to produce a new image g(x,y):

```
for y = 0 to ImageHeight do
    for x = 0 to ImageWidth do
        sum = 0
        for i = -h to h do
            for j = -w to w do
                sum = sum + k(j,i) * f(x - j, y - i)
            end for
        end for
        g(x,y) = sum
    end for
end for
```



CONVOLUTION AND POOLING OPERATION

Max pooling operation:



PROJECT 4

- Task 1: implement basic sequential matrix multiplication in software.
- Task 2: implement a fixed 3 by 3 size matrix multiplication in hardware.
- Task 3: implement a scalable matrix multiplication in hardware, and find the maximum matrix size that can be held on FPGA.
- Task4: compare running times of software and hardware implementations for different matrix sizes.
- Task 5: write a comprehensive experiment report.

PROJECT 4

- Task1: Implement the convolution operation for **k** 3x3 filters and an **n** x **n** matrix purely in software (c code). (20%)
- Task2: Implement one 3x3 convolution operation (one 3x3 filter and one 3x3 matrix) in hardware (FPGA). (15%)
- Task3: Increase the matrix size in task2 to **n** x **n**, find the maximum number of convolution operations (9 x (n-2) x (n-2)) that can be done in parallel on the FPGA. (15%)
- Task4: Compare the running time of software and hardware implementations for the same matrices. Find the exact point that the hardware starts to outperform software. (15%)
- Task5: Increase the number of filters to K. Each time you update a filter value, the hardware will update the result matrix. Compare the running time of software and hardware implementations for different K values. Report the amount of speed up achieved by this approach. (15%)
- Task6: Finally, submit your project report. (20%)

PROJECT 4

- Questions:

1.Please describe how to make your VHDL code being scalable for different matrix sizes and different filter numbers.

2. Please numerate possible hardware bottleneck types in term of resource category.

For each possible bottleneck you find, what is the maximum matrix size can be held on FPGA?

In your design, what is the real bottleneck, and the maximum matrix size can be held in your design?

(Hint: resource types such as LUT, RAM…)

PROJECT 4

Submission:

- Demo and report are required.

Due dates:

- Code: May 17th noon
- In-class demo: May 17th
- Report: May 19th midnight