

# CMPE 650

## Pattern Detection Using Convolutional Neural Network (CNN) Kernels

Convolution is a key kernel in Convolutional Neural Networks (CNN). It is also a powerful tool for object and pattern detection. In this project, we going to detect a specific pattern/object from a pool of objects and patterns.

You will be provided with a pattern and an image where you need to detect that pattern and determine how many times the pattern is being matched. Therefore, you will supply your fpga with two input images and fpga will render an output of how many times a match is found.

Step 1: we would be using convolution to extract features (filter) from the sample-image and pattern-image. There are several filters you may choose to apply here:

1. Sobel filter for vertical edge detection  $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
2. Sobel filter for horizontal edge detection  $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
3. Laplasian filter  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

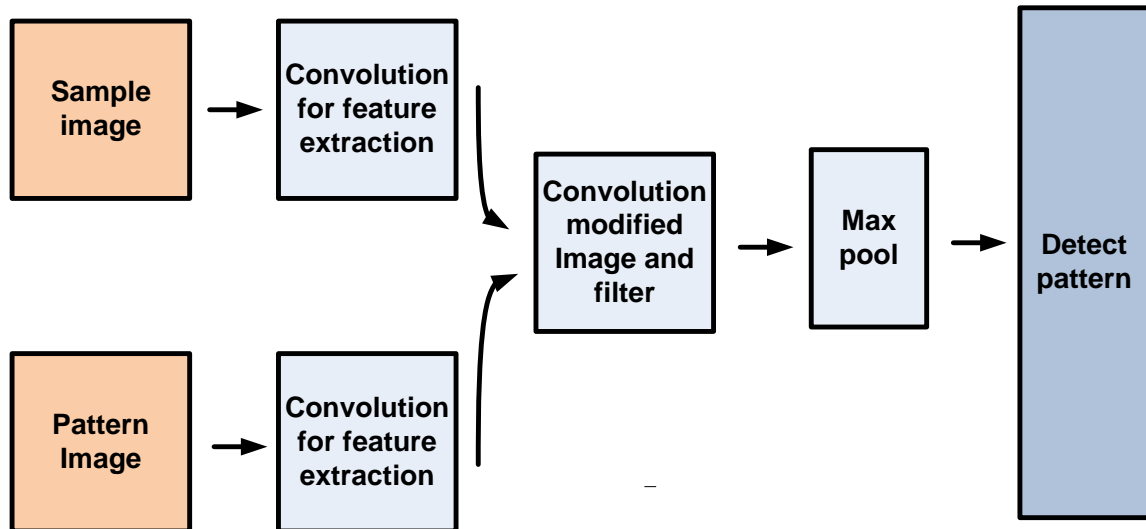
These are 3x3 filters with depth 1 (1D). Since the input is 3D you need to construct 3D filter by replication.

Step 2: after this feature extraction layer; we will convolve the feature extracted version of sample-image and pattern-image.

Step 3: a maxpool layer may be cascaded with the convolutional layer for dimensionality reduction.

Step 4: At positions of match, high value of convolution result will appear. You need to devise a scheme to detect how many times match is found. One way of doing it would be to analyze positions of maximum and near maximum spikes. You will then black out matched pattern by covering it with the black rectangle, for instance if you find 2 patterns then you should replace those patterns with three black rectangles and the final image should have only three black rectangles making rest of the image white.

Step 5: Finally, you will display the image on VGA port.



**Figure 1. Block diagram of Patter recognition by filter convolution and max pooling**

The objective is to design a fully working hardware that can perform pattern recognition using the steps explained earlier. The possible modules that need to be designed are shown in Figure 1 and as follows:

- module for convolution
- module for max pooling
- module for sorting (to detect how many time peaks are happening)
- module that controls this sequence

## **Delivery and Timeline**

### **Phase 1: Due April 25th**

- You are provided Convolution and maxpool module. You need to use these modules to construct a software prototype (preferably in Matlab) of the whole system.
- you discuss on choice of filter to use and why (discuss filter effect on Detection pattern)
- How much memory you need to handle in each layer
- How many bits of data in each layer

## Phase 2 due May 2nd

- Detailed design presentation and characterization such as number of bits for input/output and middle, memory size, control sequence, number of cycles to finish, parallel vs serial architecture, pipeline
- Verilog design and verification for convolution and maxpooling kernels,
- Convolution block should be able to read the save image and do the processing.
- Resource, timing and power analysis for kernels

## Phase 3 due May 9<sup>th</sup>

- Finish all kernel design, state machine and memories and verification in simulation
- Full analysis on timing, power and resource utilization of the kernels and projection for the design

## Phase 4 due May 16th

- Full working system, demo in simulation and on VGA.
- Your hardware should work with any new image and pattern. The demo will be evaluated upon functionality and accuracy of your design with the given image during demo time where a new image will be given

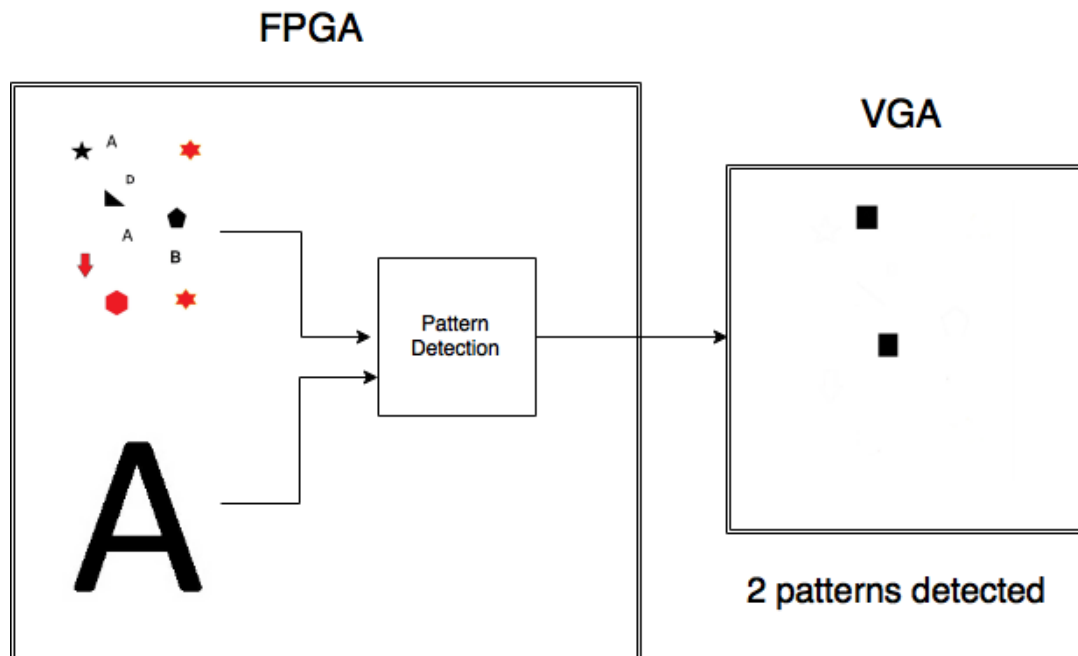


Figure 2. Desired output