# Object Detection using Convolutional Neural Network

This project aims to design a system to find the number of times the test pattern appears in the input image. Given a test image and a test pattern, the designed system can successfully detect the pattern and the number of times it occurs in the test image. Convolutional Neural Network (CNN) is used for implementing the framework of the system. The layers of CNN used are Convolutional Layer, Max Pooling Layer.

A) Implementation
i) Block Diagram
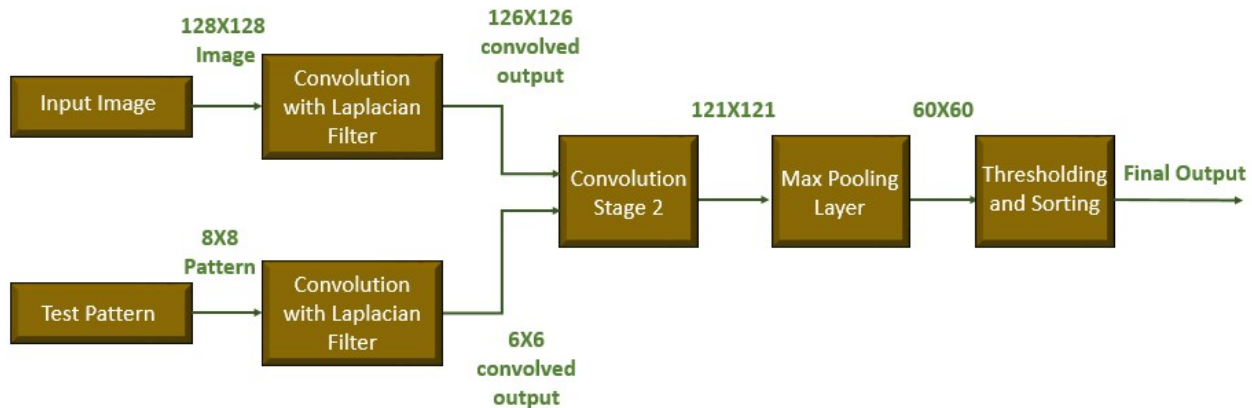
1) Input Image convolved with the Laplacian filter - The size of the image is (128X128). It is convolved with the Laplacian Filter to get a matrix of 126X126 dimensions.

2) Test Pattern convolved with the Laplacian Filter - The size of the Pattern to be detected is (8X8). It is convolved with the Laplacian Filter to get a matrix of 6X6.

3) Convolution of the Matrices obtained from step (1) and step (2) - The output matrices from step (1) and step (2) are convolved with each other. The output matrix obtained is of the dimension 121X121.

4) Max Pooling on the Output of step (3) - The factor used for Max Pooling is 2. The output matrix obtained from this layer is of the size 60X60.

5) Thresholding and Counting number of times the pattern is detected - In the final step, using the thresholding, each element in the matrix obtained at step (4) is compared with the value of convolution of the pattern with itself. Values greater than 25% are considered as matched pattern.

ii) Convolution with Laplacian (First Stage)
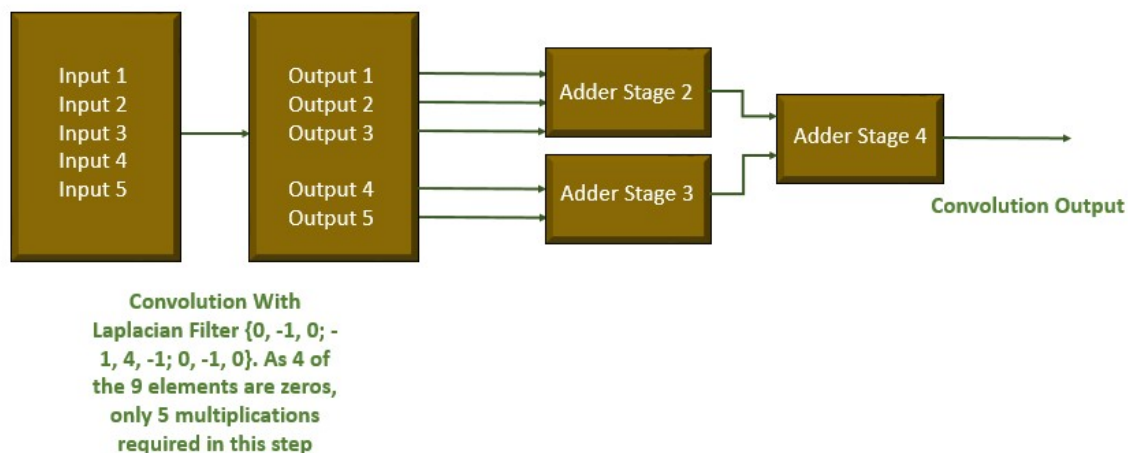
Above block diagram represents the pipelined implementation of first stage of convolution of input image and test pattern with Laplacian filter. Each convolution of 3X3 image/pattern with 3X3 Laplacian filter is completed in 4 clock cycles. The operation of convolution has been divided in to 4 stages, which helps in pipelined implementation of the convolution operation. Because of pipelining, an output is obtained at each clock cycle.

iii) Convolution of image with the pattern (Stage 2)

Below block diagrams represent the pipelined implementation of the stage 2 of convolution of outputs obtained at stage 1.
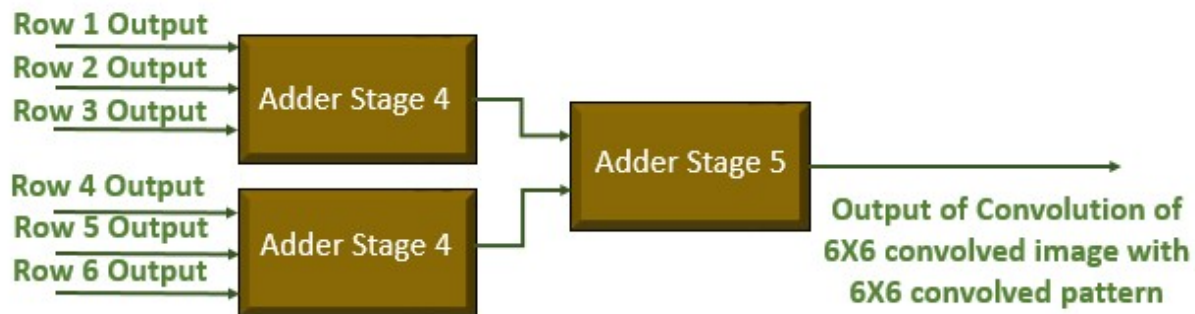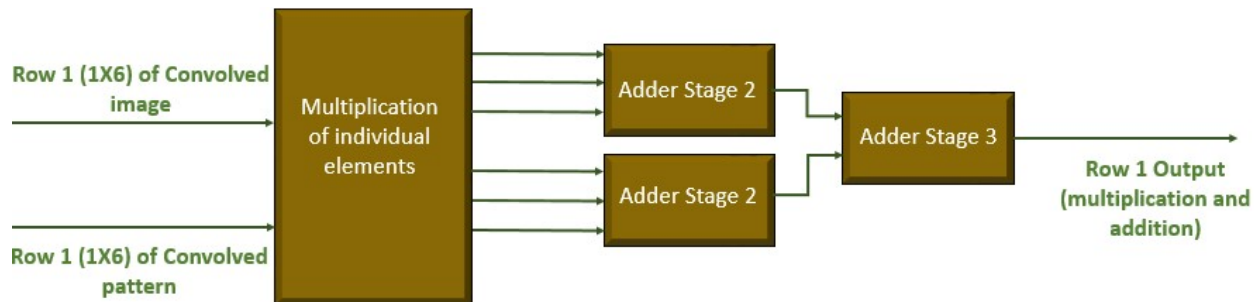
In the pipelined implementation, same numbered rows are given as input to the multiplication block, which gives 6 output and addition is also performed in a pipelined manner. Overall, a 6X6 convolution operation takes 5 clock cycles, because of use of pipelining and hardware parallelism. Once an output is obtained, each clock cycle produces one new output.
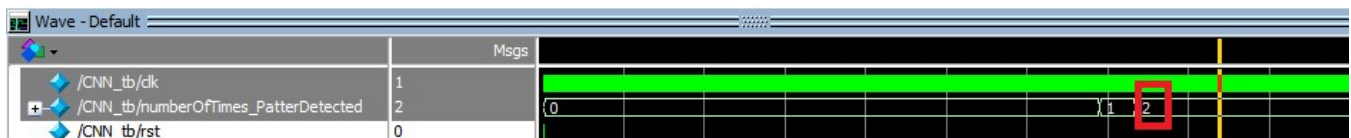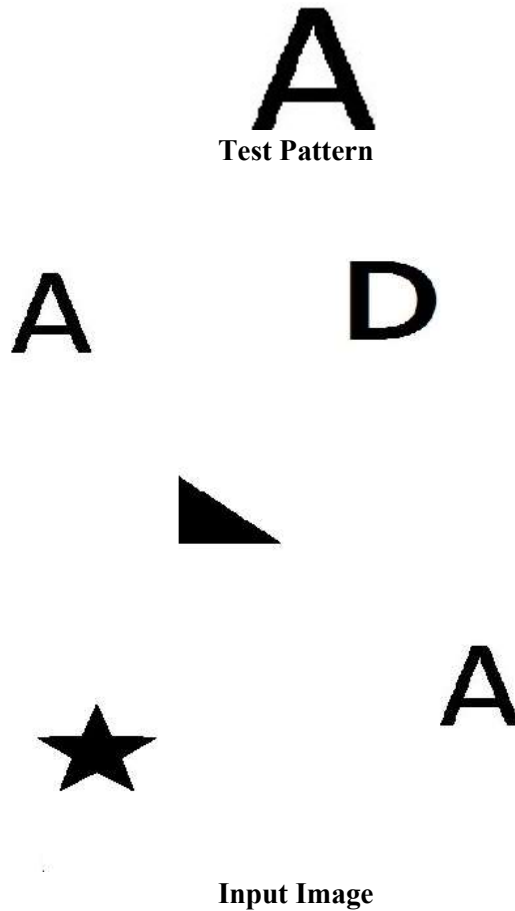
iv) Maxpooling Layer (Stage 3)
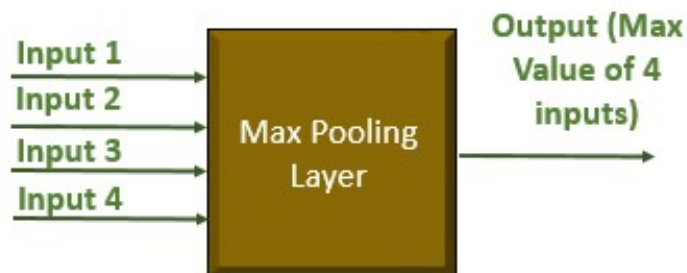
v) **Thresholding**

In the direct layer, instead of sorting and then comparing the values for finding the number of times a pattern appears in the image, direct comparison is done to avoid the time-consuming activity of sorting.

B) Results

i) **Input Image, Test Pattern and output from the RTL:**

Test Pattern

Input Image

Output from simulation (number of times the pattern appears in the image is highlighted in red)

ii)   Clock cycles required for each stage

For all stages	37722
For Convolution of Image (128X128) with Laplacian Filter (3X3)	15880
For Convolution of pattern (8X8) with Laplacian Filter (3X3)	400
For Convolution of convolved image (126X126) with convolved pattern (6X6)	14647
For Max Pooling Layer (121X121)	3602
For Thresholding (direct layer)	3600

iii)   Maximum clock speed achievable 155.67 MHz

iv)   Comparison of outputs with results MATLAB implementation and RTL implementation (with example of convolution of Pattern with Laplacian Filter)

Result from MATLAB implementation

Simulation Result from RTL implementation
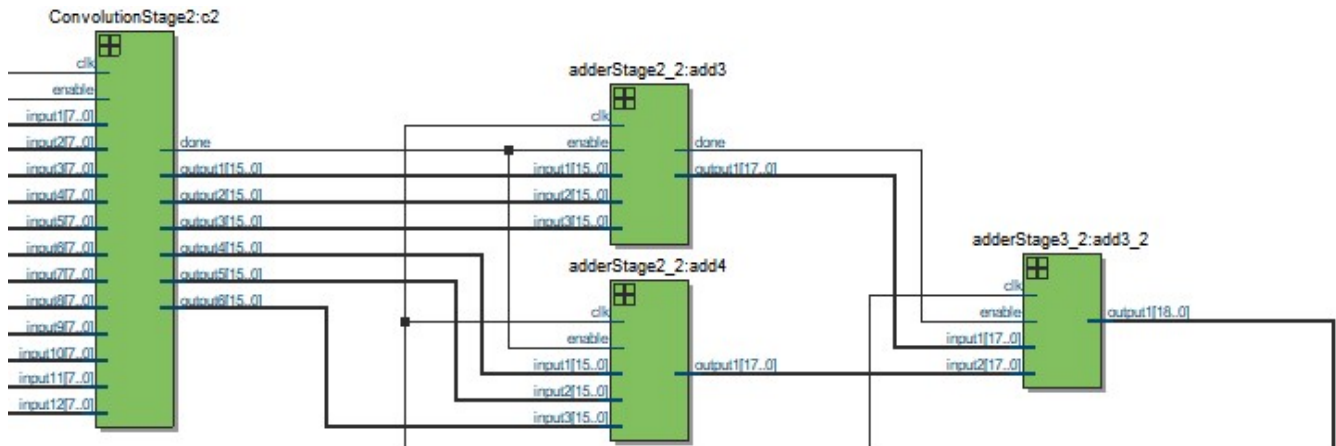
As can be observed the results match, with no error

v) **RTL netlist for individual blocks**

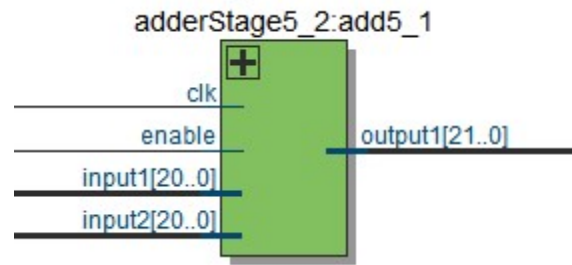### a) **Convolution (First Stage)**
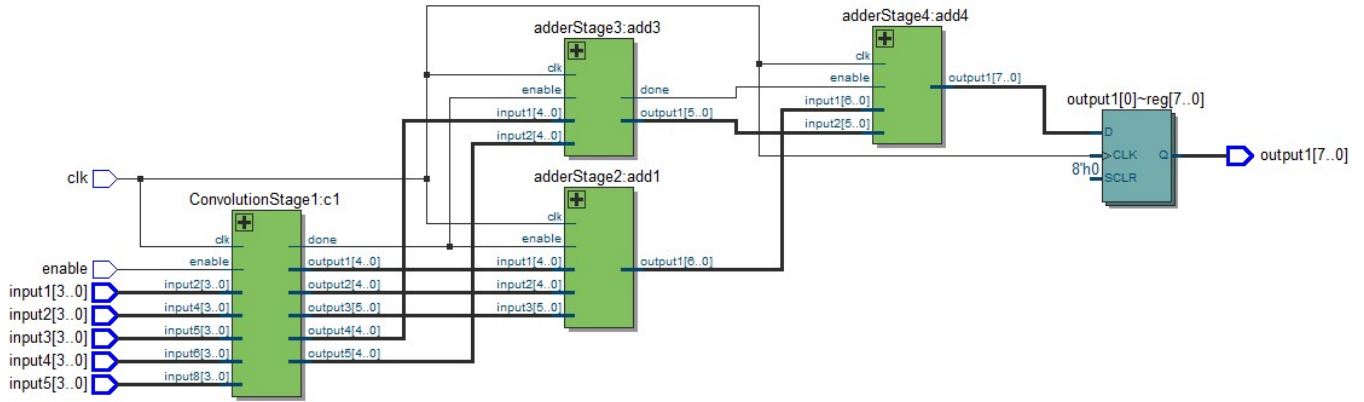
b) **Convolution (Second Stage)**

adderStage3_2:add3_2

clk
enable
input1f17_0]
input2f17_0]
outout1f18_0]

adderStage3_2:add3_3

clk
enable
input1f17_0]
input2f17_0]
outout1f18_0]

adderStage3_2:add3_1

clk
enable
input1f17_0]
input2f17_0]
done
outout1f18_0]

adderStage4_2:add4_1

clk
enable
input1f18_0]
input2f18_0]
input3f18_0]
done
outout1f20_0]

adderStage3_2:add3_5

clk
enable
input1f17_0]
input2f17_0]
outout1f18_0]

adderStage3_2:add3_4

clk
enable
input1f17_0]
input2f17_0]
done
outout1f18_0]

adderStage4_2:add4_2

clk
enable
input1f18_0]
input2f18_0]
input3f18_0]
outout1f20_0]

adderStage3_2:add3_6

clk
enable
input1f17_0]
input2f17_0]
outout1f18_0]

adderStage5_2:add5_1

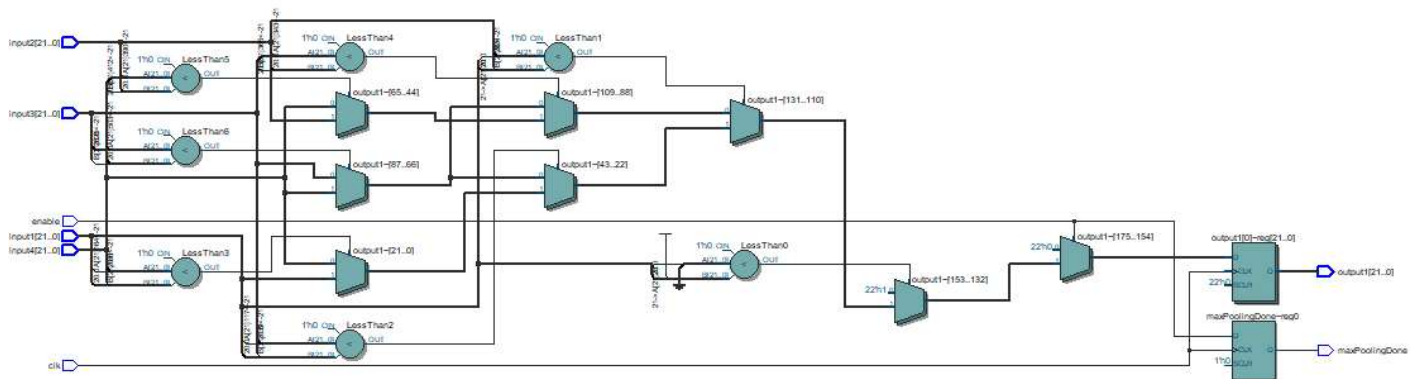### c) Max Pooling Layer

Notes about the repository:

1) In the main repository, folders Convolution Stage 1, Convolution Stage 2 and Max Pooling Layer contain the implementation and testing of individual blocks.
2) In the main repository, the input image and the test pattern are stored in .txt files named as image1.txt and pattern.txt