FPGA Implementation of CNN Algorithm for Detecting Malaria Diseased Blood Cells

Serkan Sağlam
Electrical and Electronic Engineering
Marmara University
Istanbul, Turkey
serkansaglam@marun.edu.tr

Fatih Tat
Electrical and Electronic Engineering
Marmara University
Istanbul, Turkey
fatihtat@marun.edu.tr

Salih Bayar
Electrical and Electronic Engineering
Faculty of Engineering, Marmara University
Istanbul, Turkey
salih.bayar@marmara.edu.tr

***Abstract*—In this study, Field Programmable Gate Array (FPGA) implementation of Convolutional Neural Network (CNN) for classification of malaria diseased cell is done. The hardware is designed and implemented on Xilinx Zynq-7000 FPGA using Very High–Speed Integrated Circuit Hardware Description Language (VHDL). In accordance with this purpose, Convolutional Neural Network (CNN) classification method used by image processing to make it easier for experts to comment on diseased cells. The classification method allows us to make a simpler interpretation by classifying complex images. Thanks to this research, it facilitates early diagnosis using image processing in the medical field as soon as reduces death and treatment costs. According to the experimental results, the accuracy rate for finding malaria diseased cell using CNN method is 94.76% for 200 8x8 binary images. The average execution time of CNN algorithm using Matlab on desktop PC is 174 microseconds. On the other hand, the maximum achievable frequency on Zynq FPGA is 168MHz (i.e. the longest critical path is 5.93 nanoseconds). The occupied area of CNN on Xilinx Zynq FPGA is only 783 six-input Look Up Tables (LUTs) of 17600, which is about 4.34% of Xilinx Zynq-7000 (XC7Z010-1CLG400C) FPGA.**

***Index Terms*—Matlab, CNN, FPGA, Image Processing, Classification, Malaria Detection, VHDL**

I. INTRODUCTION

At the present time, the medical disease is spreading rapidly, therefore this problem has the top-priority which leads to new technological advances and forms some rapid-grow research areas. For this reason, diagnosis of the medical disease has needed the computer-aided for a faster treatment process. Therefore, research on new diagnostic processes and systems for the medical disease is actively being conducted.

Current techniques of biomedical to detect medical disease are based on image processing and applications of this method's show a significant increase in recent years [1]. These methods are especially used in biomedical and medical fields to generate solutions for treatment. One of the significant parts of medical image processing methods is more functional for easy detection of the diseases. Furthermore, it is a practical method that allows the human eye to see the images that are transferred to the computer environment and to perform the desired results. In this way, using image processing methods let us comment on the disease by classification. Mostly used classification algorithms can be considered as Linear Classifiers: Nearest Neighbor, Neural Networks, Random Forest, Support Vector Machines, Decision Trees, Boosted Trees, Naive Bayes Classifier, Logistic Regression. Some of these algorithms can be used for image classification. It has been observed that the results of the CNN algorithm give better efficiency compared to other algorithms [2].

This study purposes to implement the CNN algorithm on FPGA by using VHDL. In addition, CNN classification method is performed on the Matlab as the control mechanism. The main purpose of this control mechanism is to compare the accuracy of the results of the system and improve the efficiency of the system. It was observed that the CNN classification method run on Matlab was slow compared to FPGA and the importance of FPGA usage was analyzed.

Hardware implementation of image processing algorithms on FPGA devices become a promising alternative for the implementation of software algorithms. FPGAs are preferred due to providing high efficiency, low energy usage and low use of embedded circuit platform hardware utilization of integrated circuit platform on research and development [3]. Especially, Xilinx FPGAs are very commonly used in both academia and industry. Xilinx FPGAs have a dominant share in the market. As Xilinx has 50% of the market, its biggest rival, Intel, has 37% and the others (e.g. Lattice Semiconductor) has the rest of the market. As a result of this, SRAM based Xilinx FPGAs will be used in the scope of this study.

This study is presented as follows: Section II CNN method which is applied for classification. Section III explains related work. Section IV provides insight into FPGA simulation and implementation. Last but not least, section V includes the conclusion and future work.

II. METHOD

A. Convolutional Neural Network

A CNN is a deep learning algorithm which can take an input image, then separate the various views and objects in the image as shown in Fig. 1. Furthermore, CNN is deep artificial neural networks that are mainly used to classify, name, clustering

978-1-7281-3729-2/19/$31.00 ©2019 IEEE

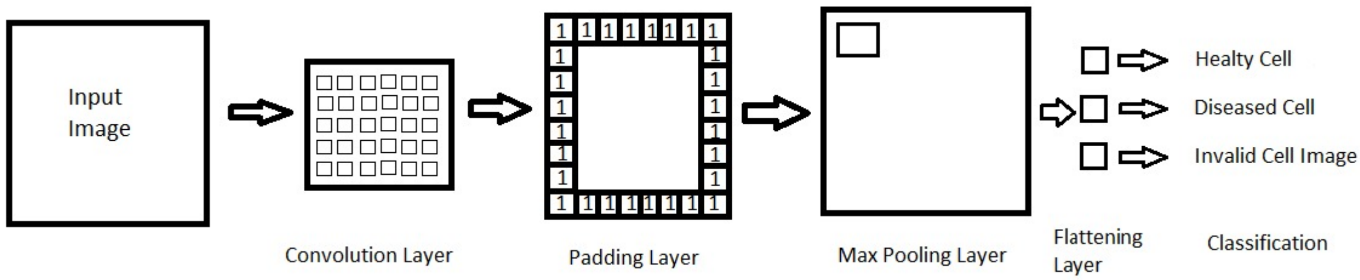


Fig. 1: Steps of CNN

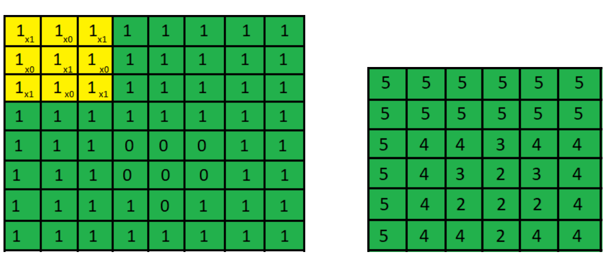
images, and object recognition. It is a frequently preferred algorithm for classification in image processing [4].

Because of its CNN structure, it takes photo or video as input and in this study, the input will be an image which can be converted to the appropriate format. In this research, the algorithm is set to perceive the given image as a matrix. Therefore, the image is first converted to a matrix format using the binary type of image. After this stage, the output matrix for the classification is created by applying the Layer, Padding, Pooling Layer and Flattening Layer methods, respectively.

B. Convolution Layer

CNN's primary construction block is Convolution Layer and applies selected filters to the image to remove the low- and high-level features in the image. This filter which usually multi-dimensional and contain respectively pixel values representing the height-width of matrices and the depth of matrices can be filter detecting edges or detecting any object.

First, the filter which is the matrix is placed in the upper left corner of the matrix format of the image to apply the Convolution Layer method. Here, the indices between the two matrices (image and filter) are multiplied by each other then all the results are summed and stored in the output matrix. Thereafter, this filter is shifted one pixel to the right and the same operations are applied. After the first line is finished, it is passed to two lines and all operations are repeated until the output matrix is created [5].



(a) (b)

Fig. 2: (a) Convolution input matrix of CNN (b) Convolution output matrix of CNN

For instance, assume that the filter matrix 3x3 and the image matrix 5x5 are taken. Thence, the 3x3 filter will move three times horizontally and vertically, and the resulting matrix will be 3x3 as shown in Fig. 2. In addition, if the image is 8x6 and the filter is 3x3, the output matrix will be 6x4. The output matrix is often called the feature map.

In short, the properties are determined by moving the filter over the image and using simple matrix multiplication. Generally, more than one filter is used to detect multiple properties, i.e., more than one convolution layer in a CNN network.

C. Padding

Padding controls how the filter evolves around the input image. Padding size affects size of the Feature Map. In the first stage of the CNN, when filtering, padding is used as it is necessary to protect as much information as possible for other Convolution Layers. The Feature Map appears to be smaller than the original input image due to convolution. Therefore, Padding will add "1" to this map to preserve the size of the image as shown in Fig. 3.

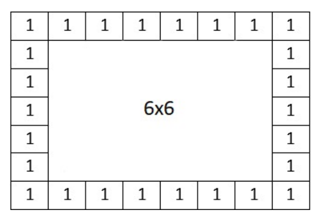
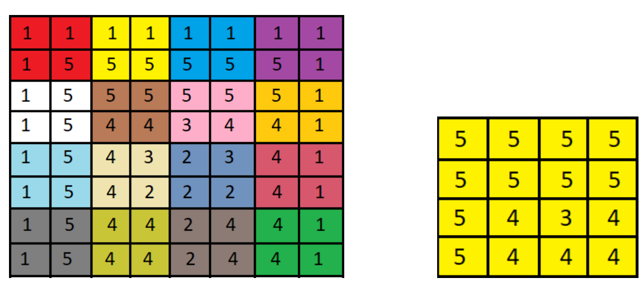


Fig. 3: Padding Layer of CNN

D. Pooling Layer



(a) (b)

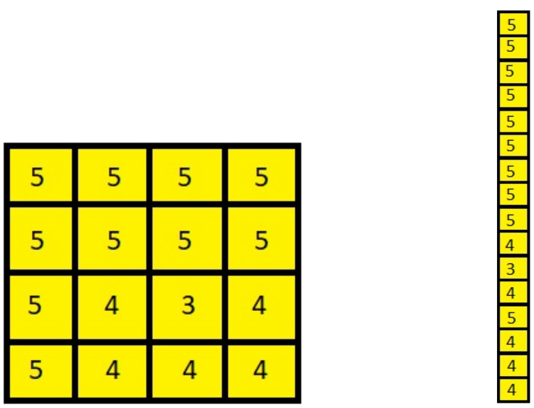
Fig. 4: (a) Max pooling input matrix of CNN (b) Max pooling output matrix of CNN

Pooling Layer which provides frequently added between consecutive convolutional layers is to reduce the shift size of the representation and the number of parameters and calculations within the network. In this way, the mismatch in the network is checked and although there are many pooling processes, the most popular is max pooling. There are also average pooling and L2-norm pooling algorithms that work on the same principle, yet the Max-Pooling method is used in the study because of that it gives better results [6].

A 2x2 filter was selected to apply the Max Pooling method. A 4x4 matrix was selected to be described in the sample. We apply the filter on the 4x4 matrix, and as you can see in Fig. 4, the filter gets the largest number in the area it covers. In this way, it uses smaller outputs that contain enough information for the neural network to make the right decision.

E. Flattening Layer

The task of this layer is simply to prepare the data in the input of the last and most important layer, Fully Connected Layer. In general, neural networks receive input data from a one-dimensional array as shown in Fig. 5. The data in this neural network is the one-dimensional array of matrices from the Convolutional and Pooling layers.



(a) (b)

Fig. 5: (a) Flattening Layer input matrix of CNN (b) Flattening Layer output matrix of CNN

III. RELATED WORK

A. Hardware Implementation of Diseased Cell Classification

In study [7], A Verilog-based image processing system is described using the sequential processing of conventional algorithms to discover the ideal border detection solution for skin lesion images. In accordance with this purpose, Contrast manipulation, Brightness manipulation, Gray level transform, Invert transform and Thresholding methods are utilized in the system. These techniques are implemented using a Verilog HDL on the FPGA.

B. k-Nearest Neighbors for Skin Disorders

This study [8] examines the geometric characteristic of skin disorders to detect and classify skin disorders. The k-Nearest Neighbors (k-NN) algorithm is used to classify the distortion images of the skin and to determine whether the deterioration of the skin is a malignant or benign melanoma and show a 90% of accuracy in the functionality test.

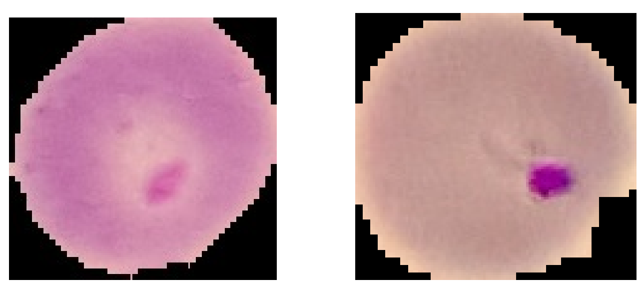
C. Support Vector Machine for Melanoma Cancer Classification

In [9], the authors developed a Computer-Aided Design(CAD) that determine to improve a real-time embedded classifier in a low-cost hand-held device to provide early detection of melanoma cancer using classification method which is the Support Vector Machines (SVMs). Melanoma is thought to be the form of skin cancer responsible for a significant proportion of skin cancer mortality. Because of this reason, early diagnosis for melanoma cancer provides a reduction in mortality and treatment costs. Therefore, CAD systems are needed to improve early detection of cancer.

D. CNN Classification Method for detection of Malaria diseased cells on Matlab

In [10], the authors implemented CNN algorithm in order to classify malaria diseased cell on Matlab. A very extensive experimental set is used to achieve a high accuracy rate. Although the proposed accuracy very high, the implementation on Matlab could slow down system performance. However, we propose a massively parallelized hardware implementation of CNN algorithm which results in not only high accuracy but also high performance in terms of both time and circuit area.

IV. FPGA SIMULATION AND IMPLEMENTATION



(a) (b)

Fig. 6: (a) Example of a healthy blood cell (b) Example of a malaria diseased blood cell

The Vivado is the software development tool used to improve the proposed system and the CNN algorithm. This study is written in VHDL which is one of the hardware description languages. Healthy and diseased data were taken from the U.S National Library of Medicine site [11]. A total of 200 images were tested with 90 malaria diseased, 90 healthy and 20 invalid samples on Vivado Simulation. According to the CNN algorithm implementation, 189 out of 200 hundred images are classified correctly; this shows the accuracy rate as 94.76%. Firstly, all tests are done using Matlab in a Windows desktop PC; Intel Core i5-6500CPU @3.20GHZ with 8GB DDR3 RAM. Implementation of CNN algorithm using Matlab for processing a single 8x8 binary image on desktop PC takes about 174 microseconds. The same test takes about 5.93 nanoseconds on the Xilinx Zynq-7000 (XC7Z010-1CLG400C) FPGA. When the results obtained on FPGA are compared with the results of Matlab, it is obviously seen that the system speeds up highly.

Examples of healthy and malaria diseased blood cell images are given in Fig. 6(a) and Fig. 6(b), respectively. In addition, the classification results of the experiments performed on 200 samples are shown in Table 1.

TABLE I: RESULT OF CNN CLASSIFICATION

#	Training Set	Classification Output
1	Image 1 to 90	Diseased Cell
2	Image 91 to 180	Healthy Cell
3	Image 181 to 200	Invalid Cell Image

Simulations were performed on the diseased sample given in Fig. 6(b) and the size of the image was selected as 8x8 in order to show the samples of the matrices in the article.

Input_Array[0:7,0:7]	
[0,0:7]	1,1,1,1,1,1,1,1
[1,0:7]	1,1,1,1,1,1,1,1
[2,0:7]	1,1,1,1,1,1,1,1
[3,0:7]	1,1,1,1,1,1,1,1
[4,0:7]	1,1,1,0,0,0,1,1
[5,0:7]	1,1,1,0,0,0,1,1
[6,0:7]	1,1,1,1,0,1,1,1
[7,0:7]	1,1,1,1,1,1,1,1

(a)

Filter_matrix[0:2,0:2]	
[0,0:2]	1,0,1
[1,0:2]	0,1,0
[2,0:2]	1,0,1

(b)

Fig. 7: (a) Experiment image matrix (b) Filter matrix

Fig. 7(a) represents the binary version of the diseased image used in the simulation. In addition, Fig. 7(b) is the filter selected to calculate the convolution matrix for CNN algorithm.

Fig.8(a), Fig. 8(b), Fig. 9(a), and Fig. 9(b) gives a break downs of the Convolution Layer simulation output, Padding Layer simulation output, Max Pooling Layer simulation output, Flattening Layer Layer simulation output on Vivado of the simulated diseased sample, respectively.

V. CONCLUSIONS AND FUTURE WORK

The main contribution of this study is to classify malaria diseased cell with the help of the image processing thanks to CNN algorithm by using the VHDL language on FPGA which provides high efficiency, low energy usage and low use of embedded circuit platform hardware. According to the experimental results, the accuracy rate for finding malaria diseased cell using CNN method is 94.76% for 200 8x8 binary images. The average execution time of CNN algorithm using Matlab on desktop PC is 174 microseconds. On the other hand, the maximum achievable frequency on Zynq FPGA is 168MHz (i.e. the longest critical path is 5.93 nanoseconds). The occupied area of CNN on Xilinx Zynq FPGA is only 783 six-input Look Up Tables (LUTs) of 17600, which is about

Convolution_Layer[0:5,0:5]	
[0,0:5]	5,5,5,5,5,5
[1,0:5]	5,5,5,5,5,5
[2,0:5]	5,4,4,3,4,4
[3,0:5]	5,4,3,2,3,4
[4,0:5]	5,4,2,2,2,4
[5,0:5]	5,4,4,2,4,4

(a)

Padding_Layer[0:7,0:7]	
[0,0:7]	1,1,1,1,1,1,1,1
[1,0:7]	1,5,5,5,5,5,5,1
[2,0:7]	1,5,5,5,5,5,5,1
[3,0:7]	1,5,4,4,3,4,4,1
[4,0:7]	1,5,4,3,2,3,4,1
[5,0:7]	1,5,4,2,2,2,4,1
[6,0:7]	1,5,4,4,2,4,4,1
[7,0:7]	1,1,1,1,1,1,1,1

(b)

Fig. 8: (a) Convolution Layer simulation output on Vivado (b) Padding Layer simulation output on Vivado

Max_Pooling_Layer[0:3,0:3]	
[0,0:3]	5,5,5,5
[1,0:3]	5,5,5,5
[2,0:3]	5,4,3,4
[3,0:3]	5,4,4,4

(a)

Flattening_Layer[0:15,0:0]	
[2,0:0]	5
[3,0:0]	5
[4,0:0]	5
[5,0:0]	5
[6,0:0]	5
[7,0:0]	5
[8,0:0]	5
[9,0:0]	4
[10,0:0]	3
[11,0:0]	4
[12,0:0]	5
[13,0:0]	4
[14,0:0]	4
[15,0:0]	4

(b)

Fig. 9: (a) Max Pooling Layer simulation output on Vivado (b) Flattenig Layer simulation output on Vivado

4.34% of Xilinx Zynq-7000 (XC7Z010-1CLG400C) FPGA. This development is a significant part of early detection of malaria diseased cell because that early diagnosis is one of the key factors for the treatment of the diseased cell. A CAD system inspired by this work can play an important role in medicine for the early detection of disease.

In the future work, the image size will be increased primarily; it is aimed to examine 32x32 and 64x64 images for classification studies. Another priority is to determine the classification algorithms that can work together for malaria diseased cell diagnosis by examining the CNN algorithm as well as other classification algorithms. In addition, it will be displayed by projecting the classified image on an additional screen using Video Graphics Array(VGA) or High Definition Multimedia Interface (HDMI) instead of obtaining matrix results, solely.

REFERENCES

[1] Bl Basavaprasad and M Ravi. A study on the importance of image processing and its applications. IJRET: International Journal of Research in Engineering and Technology, 3, 2014.

[2] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. Medical image classification with convolutional neural network. In 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pages 844–848. IEEE, 2014.

[3] Sparsh Mittal, Saket Gupta, and Sudeb Dasgupta. Fpga: An efficient and promising platform for real-time image processing applications. In National Conference On Research and Development In Hardware Systems (CSI-RDHS), 2008.

[4] Adhish Prasoon, Kersten Petersen, Christian Igel, François Lauze, Erik Dam, and Mads Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In International conference on medical image computing and computer-assisted intervention, pages 246–253. Springer, 2013.

[5] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

[6] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In International conference on artificial neural networks, pages 92–101. Springer, 2010.

[7] Iuliana Chiuchisan and Oana Geman. An approach of fpga technology in skin lesion detection. In 2018 International Conference and Exposition on Electrical And Power Engineering (EPE), pages 0175–0178. IEEE, 2018.

[8] Noel B Linsangan, Jetron J Adtoon, and Jumelyn L Torres. Geometric analysis of skin lesion for skin cancer using image processing. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pages 1–5. IEEE.

[9] Shereen Afifi, Hamid GholamHosseini, and Roopak Sinha. A low-cost fpga-based svm classifier for melanoma detection. In 2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), pages 631–636. IEEE, 2016.

[10] Zhaohui Liang, Andrew Powell, Ilker Ersoy, Mahdieh Poostchi, Kamolrat Silamut, Kannappan Palaniappan, Peng Guo, Md Amir Hossain, Antani Sameer, Richard James Maude, et al. Cnn-based image analysis for malaria diagnosis. In 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 493–496. IEEE, 2016.

[11] U.S National Library of Medicine. (2018). In Retrieved from https://ceb.nlm.nih.gov/repositories/malaria-datasets/.