AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°2; pp 429-441
Submitted Mar. 19, 2017; Revised May 12, 2017; Accepted May 25, 2017

Design of FPGA-based Handwriting Image Recognition System

*Lei Wang, **Ziheng Yang*, ***Guangqiang Xu, ****Meili Fu, *****Yu Wang

*Information and Network Center, Heilongjiang University, Harbin 150080, China
**School of Electronic Engineering, Heilongjiang University, Harbin 150080, China
(yzh@hlju.edu.cn)
***School of Electronic Engineering, Heilongjiang University, Harbin 150080, China
****School of Electronic Engineering, Heilongjiang University, Harbin 150080, China
*****School of Electronic Engineering, Heilongjiang University, Harbin 150080, China

Abstract

This paper designs a program to realize the handwriting image recognition algorithm using FPGA. The digital character recognition system designed in this paper is composed of software design and hardware design. In terms of software, the character processing is divided into image denoising, image binaryzation, grey processing and refinement treatment. Each process algorithm is developed with Matlab. By virtue of its powerful image processing function, the algorithm is simulated. By transforming the DBN into large data volume matrix operations and combining the implementability of FPGA, each link of the identification algorithm is determined. The program mainly uses Altera Cyclone IV chip and several calculation cores connected by DBN are realized through hardware programming. The hardware implementation of the module function and algorithm is verified through Modelsim simulation of each module.

Key words

Handwriting image recognition, FPGA, DBN, Matrix multiplication.

1. Introduction

Subordinated to character recognition, digital recognition is a traditional research field of pattern recognition discipline. According to different classification criteria, digital recognition can

be divided into online and offline recognition, print form and handwriting digital recognition, etc. [1]. The recognition of handwritten digits is an area which people are studying more deeply, because the handwritten digit recognition can be applied to many aspects, such as post code identification, automated processing of bank bills, and so on. In recent years, the offline handwritten digit recognition has shown a great development. But because the writing style of handwritten digit varies from person to person, it is hard for the offline handwritten digit recognition to get close to the digit recognition by humans in terms of accuracy, flexibility and fault tolerance [2]. Existing algorithms have their own advantages, but most of the algorithms focus on the recognition of a single number, with weak control of the overall situation. From the point of view of artificial intelligence, it is a good idea to study the thought of people when they identify handwritten numbers, and then improve existing algorithms to increase the recognition rate of handwritten numbers [3]. Therefore, it is very meaningful to carry out research on handwriting recognition in this paper.

Aiming at the handwriting image recognition, many experts and scholars put forward different solutions, and have made a lot of achievements. Tausheck presented the theoretical concept of OCR in the 1920s, and Handel implemented the OCR technology and applied for intellectual property rights in the following years [4]; Casey and Nagy identified and classified over 1,000 characters with the template matching algorithm. Hinton put forward Deep Belief Network (DBN) in literature [5], whose error rate is 2.25% after it is trained. However, DBN network structure is rather complicated, the time complexity is great, and the real time performance is poor. Hinton proposed the Dropout technology in literature to reduce the complexity of the algorithm and solve the problem of over-fitting through increasing the participation of the hidden unit of the neural network in the algorithm [6].

In China, digital recognition started in the 1980s, and after years of development, there have been many methods and means of digital identification, which can be broadly divided into two categories, namely, the statistical analysis based on the overall situation and the structure-based feature analysis [7]. The statistical analysis based on the overall situation is to recognize different digits through using template matching, pixel density, feature points and mathematical manipulation; the structure-based feature analysis is to identify different digits mainly from the outline of the digits and the character shape, including circle, endpoint, arc, horizontal length,etc. In recent years, a large number of digital identification theories and systems have been seen in the world: Wang Yongqian, Lv Rong et al. mainly studied a kind of digit recognition method based on BP neural network. Firstly, the digital characters are segmented and refined, and the digital training

samples are generated. Then, a lot of training to the neural network is carried out by using the digital training samples. Finally, the trained BP neural network is used to identify the untrained digital samples. After testing, the accuracy rate of the digital recognition can reach over 95%. Huang Xinye, Wang Maoxiang and FU Yuqing put forward a new digit recognition approach. Based on the analysis of structural features, this algorithm realized fairly high recognition rate. On the basis of this algorithm and combing with the image pretreatment module, the feature extraction module and the recognition module, a digital identification system is established finally. The recognition ability of the system is tested and the result is good [8].

This paper designs a program to realize the FPGA-based handwriting image recognition algorithm. The program mainly uses Alter a FPGA chip and hardware programming to recognize several calculation cores connected by DBN and the digital information in the image and to display the identification result. This program also makes full use of the design concept of FPGA parallelism and modularization to improve the calculation efficiency of this algorithm.

2. DBN Recognition Algorithm

2.1 Image Pretreatment

Image preprocessing first transforms the outside non-digital signals into picture-form digital signals with a video camera. Then, these digital signals will be processed with a computer before they are recognized. The purpose is to have the obtained pictures preliminarily arranged, which will facilitate the subsequent identification. Several handling ways for image preprocessing are as follows:

1) Filter denoising and signal enhancement

When an image is captured by an image pickup apparatus, the external factors such as the light may cause the acquired image to have some additional information, and it is necessary to filter out the disturbance noise and highlight the characteristics of the image, but this operation may cause the image information to be too faint, which is hard to be processed by the Computer. Therefore, the enhancement of the signal strength is the key to ensure the smooth operation of the following steps [9].

2) Grey Processing

The images are mostly colorful. Other colors are expressed generally through the mutual overlap of red, green and blue [10]. If the color is ignored in the recognition process, grey processing shall be carried out to change the color value into black and white, which is generally expressed with the gray value.

3) Binaryzation processing

It is usually used after the image gray processing.In order to highlight the difference between the main body and the background, this kind of expression is used to represent the theme and the background in the picture, which can greatly reduce the amount of information in the picture and the complexityof calculations.

4) Refinement treatment

The shape of the image is very important information, which is more emphasized in many image recognitions. The refinement processing is to remove the redundant information and retain the shape information of the image, and then refine the entire image into a frame composed of lines, thus extracting its shape and contour.

2.2 Image Feature Extraction

Before theimage information is input to the classifier for identification, thestepof feature extraction shall be performed, because many pixel points in the image is irrelevant to the information required by the classifier. If the information of the entire picture is input to the classifier, the amount of data will be quite large, which will increase a lot of useless computing, and result in inefficient identification process. However, the feature extraction may lead to the loss of information which is good for classification during the process of information integration [11]. Therefore, it is a key step before the classification to simplify the data size of the image under the premise of not damaging the image information for the feature extraction.

2.3 Deep Belief Network

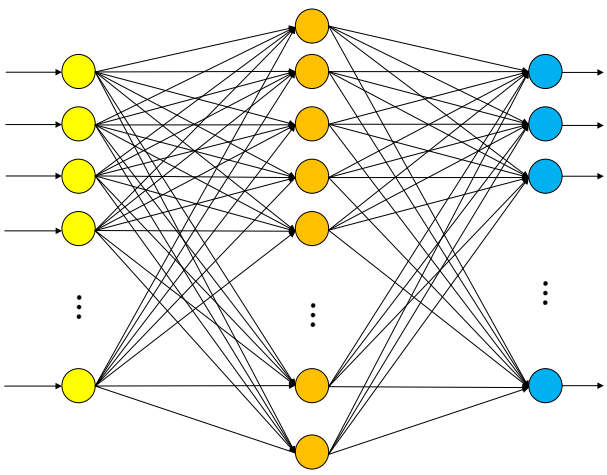

Fig.1. Three-layer BP Neural Network

Deep Belief Network (DBN) is a generative deep learning structure, which can obtain the joint probability distribution of the observed data and the corresponding category, and can easily predict the prior probability and posterior probability [12]. DBN can be seen as the superposition of a number of RBMs. The first RBM input serves as the global input, the output of the hidden layer is the input of the next layer of RBM, and the RBM output of the last layer is theglobal output. When stacking RBM, RBM is used at the input end, and BP network is used for connection at the output end.

The BP network is usually composed of input layer, hidden layerand output layer. The training consists of forward propagation ofinput and feedback propagationof error. Figure 2-4 is a simple BP neural network model with anhidden layer.

Now the learning and training process of the three-layer BP network is elaborated herein with Figure 2-4 as examples. Assume the input vector $\vec{x}=(x_1, x_2, \cdots x_n)$, the output vector $\vec{y}=(y_1, y_2, \cdots y_m)$, the expected output vector $\vec{o}=(o_1, o_2, \cdots o_m)$ and the hidden layervector $\vec{p}=(p_1, p_2, \cdots p_l)$. The weight from input layer to the hidden layeris $\vec{w}_i=(x_{i1}, x_{i2}, \cdots x_{in})$, i=1, 2, …, l. The weight from hidden layer to output layer $\vec{v}_j=(x_{j1}, x_{j2}, \cdots x_{jl})$, j=1, 2, .., m. Where n is the number of input layer units, l is the number of hidden units and m is the number of the output units.

The activation value s_i of each nerve cell in the hidden layer is:

$$s_i=\sum_{q=1}^{n} w_{iq}x_q-\theta_i,\ i=1,2,\cdots,l \tag{1}$$

θ_i is the threshold value of the hidden unit and the activation function is:

$$f(x)=\frac{1}{1+e^{-ax}},\ 0<f(x)<1 \tag{2}$$

The output value of the i_{th} unit of the hidden layer:

$$p_i = f(s_i) = \frac{1}{1+\exp\left(-\sum_{q=1}^{n} w_{iq} x_q + \theta_i\right)},\ i = 1, 2, \cdots, l \tag{3}$$

θ_i and the weight are constantly revised during the learning process.

The activation value s_j of the nerve cell on the output layer:

$$s_j = \sum_{i=1}^{l} v_{ji} p_i - \theta_j,\ j = 1, 2, \cdots, m \tag{4}$$

θ_j is the threshold value of the output layer unit.

The output value of the ith unit of the output layer:

$$y_j = f(s_j) = \frac{1}{1+\exp\left(-\sum_{i=1}^{l} v_{ji} p_i + \theta_j\right)},\ j = 1, 2, \cdots, m \tag{5}$$

When the error is propagandized in a contrary direction, the output layer will rectify the mistake.

$$d_j = (o_j - y_j) y_j (1 - y_j),\ j = 1, 2, \cdots, m \tag{6}$$

$$e_j = \left[\sum_{j=1}^{m} v_{ji} d_j\right] p_i (1 - p_q),\ i = 1, 2, \cdots, l \tag{7}$$

The connection weight from the output layer to the hidden layer and the correction of the threshold value of the output layer;

$$\Delta v_{ji} = \alpha d_j p_i \tag{8}$$

$$\Delta \theta_j = \alpha d_j \tag{9}$$

α is the learning coefficient which is larger than 0 and smaller than 1.

The correction from the hidden layer to the input layer is:

$$\Delta w_{iq} = \beta e_j x_q \tag{10}$$

$$\Delta \theta_i = \beta e_j \tag{11}$$

β is the learning coefficient which is larger than 0 and smaller than 1.

It can be learned from the above summary that BP algorithm is mainly composed of two processes: the forward propagation of the input and the reverse propagation of the error. The two processes alternate constantly until the training reaches the preset number of times or the error satisfies the requirement.

RBM (Restricted Boltzmann Machine), is a generative probability model, including a hidden layer and a visual layer. The hidden layer unit and the visual layer unit are all connected through the weight matrix and the bias vector, but the layers within the unit are not connected to each other. RBM can be used to model the probability data, and the energy function and the joint probability distribution formula are as follows:

$$E(v,h) = -b^T v - c^T h - h^T w v \tag{12}$$

RBM contingent probability:

$$p(h \mid v) = \prod_i p(h_i \mid v) \tag{13}$$

$$p(v \mid h) = \prod_j p(v_j \mid h) \tag{14}$$

Formula for weight update and bias adjustment:

$$w = w + \alpha(p(h_i = 1 \mid v_j)v_j - p(h_{i+1} = 1 \mid v_{j+1})v_{j+1}) \tag{15}$$

$$b = b + \alpha(v_j - v_{j+1}) \tag{16}$$

$$c = c + \alpha(p(h_i = 1 | v_j) - p(h_{i+1} = 1 | v_{j+1})) \quad (17)$$

RBM adopts the greedy layer-wise training. The parameters of each layer will be fixed after being trained. As the visual layer of next RBM, these parameters will train the parameters of the next layer. The cycle training is completed until the training of all RBM contained in DBN is finished.

In this paper, we first use DBN pre-training weights and thresholds to initialize the BP network to obtain faster speed and better results, and then use Dropout technology to improve the BP algorithm to improve the algorithm speed and robustness [13]. In this paper, DBN recognition algorithm is used and MNIST database handwriting image data are used for training and testing.

2.4 Designer of Classifier

Adopting the method of pattern recognitionand large picture sample training, the classifier useserrors to adjust the various parameters in the classifier model. Take the eigenvalues extracted from the image as input. After the calculation by the classifier model, the category of the picture will be learned to achieve the purpose of identification. The classifier is the most important step in image recognition.

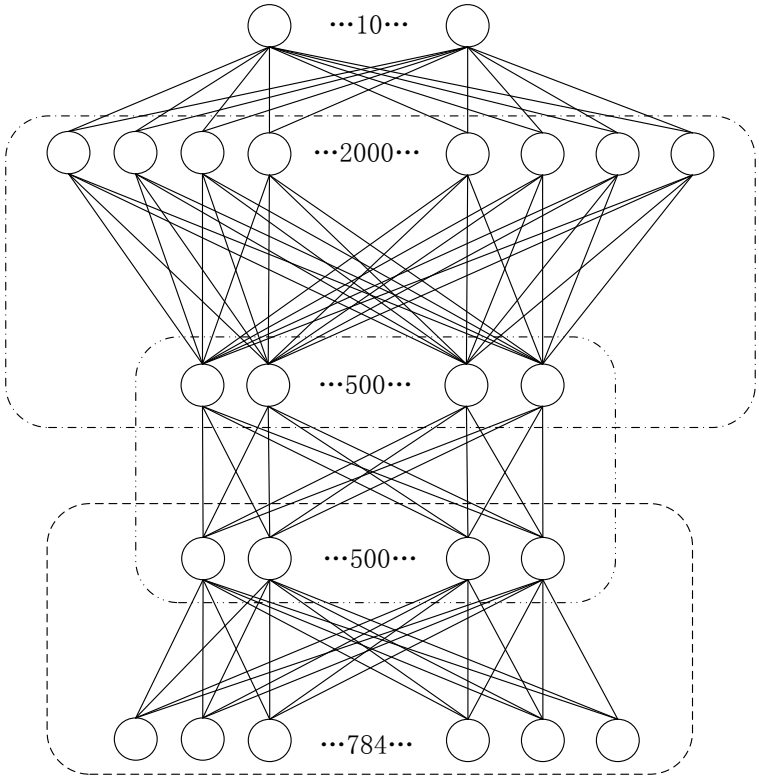

Fig.2. DBN Network Structure

Handwriting recognition uses the DBN structure of three-layer hidden layer. The three hidden layers are set to include 500,500,2,000 units respectively. Since the database contains 10 digits from 0-9, the output layer contains 10 units. The final system model structure is 28×28(784)-500-500-2000-10. 28×28(784)-500-500-2000 is the input layer and three hidden layers of DBN. 10 is the number of species of samples. The data are transmitted to soft max neural network structure for recognition and classification.

3. Realization of FPGA

The Verilog code is written and compiled through Quartus II and the handwriting image recognition algorithm is implemented on FPGA hardware. DBN network training and image preprocessing are implemented in Matlab. After training, the network weights are obtained and stored temporarily in SD card.

The main work of FPGA is divided into four steps:

1) Read the network weight data of each layer stored in PC from SD card;

2) Cache the data of each layer into the externally connected mass-memory unit DDR2;

3) Read from DDR2 following certain rules and carry out calculations of related DBN classifier;

4) Results of DBN classification and identification are shown through LCD after VGA controller matches appropriate image results

The main work flow is data reading, and the data will be written by FPGA from SD card into DDR2. Due to the huge amount of image data and network weights, in order to improve the calculating speed, the speed of data access must be improved. DDR2 has a bulk cache. The DBN classifier network weights is all from DDR2, and the overall structure is shown in Figure 3.

DBN classifier mainly uses the floating-point matrix multiplication, matrix addition, matrix division, matrix EXP and other operations. The matrix multiplication and addition operations mainly realize the transformation of DBN network abstraction operation to the actual operation [14]. Other operations are mainly used to implement the sigmoid excitation function, the error adjustment formula and the Gibbs sampling theory as well as the quantification and prediction of the output data. The image data input by DBN recognition network is stored in the ROM, and the weight matrix is stored in DDR2. Before the operation, the weight data are read out line by line to be written into RAM. Then the floating-point multiplication operation shall be carried out between the metadata and the weight data at corresponding read address. Add the product to the last sum and storage the sum into the register inside FPGA for next summation operation. After the summation of the data in a row is completed, the output value needs to be further quantified through the sigmoid excitation function. Repeatedly read the network weights and the quantified coding operation that is related to recognition algorithm shall be done between the network weights and corresponding unit data of the visual layer.

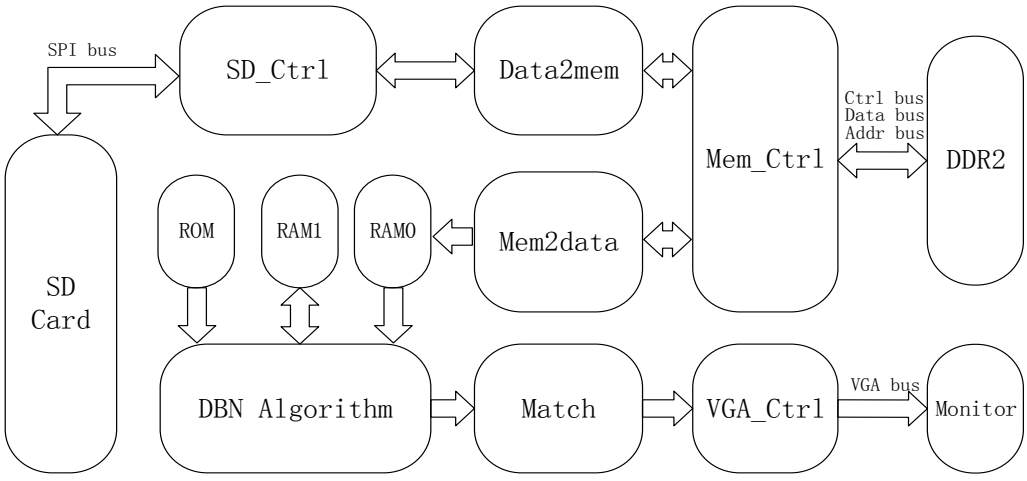

Fig.3. Overall Structure Chart

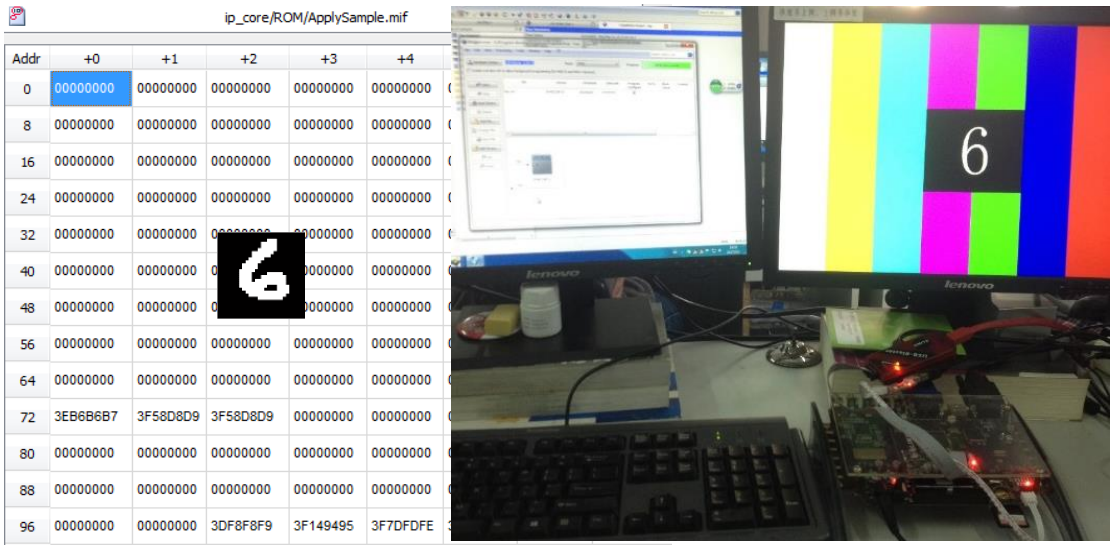

Fig.4. FPGA Handwriting Image System Recognition Results

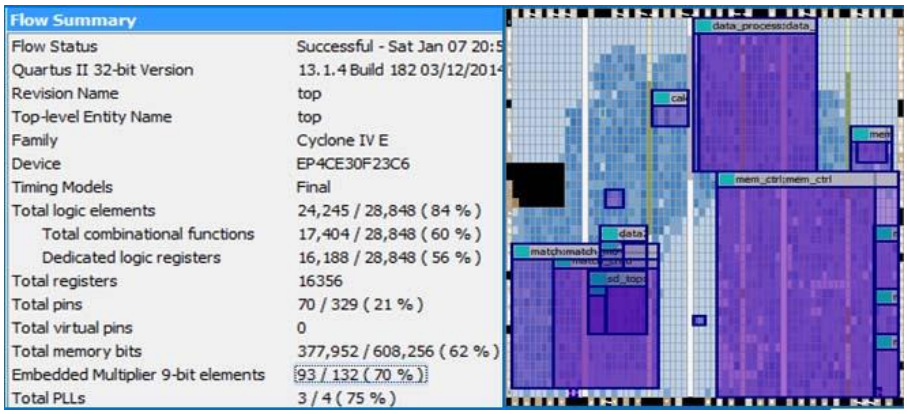

Fig.5. FPGA Hardware Resources Report and Physical Resources Mapping

Conclusions

This paper presents a FPGA-based program to realize handwriting image recognition. After the system is designed, the system was jointly debugged [15]. The image of the handwritten digit of "6" was input and stored as a ROM readable mif file. This image data serve as the input of the recognition system, and the result is recognized by the test system. After the recognition of the recognition system, the output result is the digit "6", as shown in Figure 4. Due to the limited hardware resources, the algorithm cannot be fully parallelized, and the recognition speed needs to be improved. FPGA hardware resource report and physical resource mapping situation are shown in Figure 5.

References

1. M.S.B. Ameur, A. Sakly, A. Mtibaa, Implementation of real coded genetic algorithms using FPGA technology, 2013, 10th International Multi-Conference on Systems, Signals and Devices (SSD).
2. M.S.B. Ameur, A. Sakly, A. Mtibaa, Implementation of real coded PSO algorithms using FPGA technology, 2014, 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA).
3. M.S.B. Ameur, A. Sakly, FPGA based hardware implementation of bat algorithm, 2017, Applied Soft Computing, vol. 58, pp. 378-387.
4. D. Bratton, T. Blackwell, Understanding particle swarms through simplification: a study of recombinant PSO, 2014, Proceedings of the 9th 1013 Annual Conference on Genetic and, Evolutionary Computation (GECCO'07). pp. 2621–2627.
5. M.C. Chi, Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem, 2015, Applied Soft Computing, vol. 26, pp. 378–389.
6. Q. Wanga, Y. Lia, B. Shaob, S. Deya, P. Lia, Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA, 2017, Neurocomputing, vol. 221, pp. 146–158.
7. M.F. Tolba, A.M. Abdelaty, N.S. Soliman, L.A. Said, A.H. Madian, A.T. Azar, A.G. Radwan, Y. Lia, FPGA implementation of two fractional order chaotic systems, 2017, Int. J. Electron. Commun, vol. 78, pp. 162–172.
8. W. Ni, X. Gao, Y. Wang, Single satellite image dehazing via linear intensity transformation and local property analysis, 2017, Neurocomputing, vol. 176, pp. 25–29.
9. J.L. Raheja, S. Subramaniyam, A. Chaudhary, Real-time hand gesture recognition in FPGA, 2016, Optik, vol. 127, pp. 9719–9726.
10. J.L. Raheja, A. Singhal, Sadab, A. Chaudhary, Android based portable hand sign recognition system, recent trends in hand gesture recognition, 2015, Science Gate Publishing, USA, pp. 1–18.
11. D.V. Rao, S. Patil, N.A. Babu, N. Muthukumar, Implementation and evaluation of image processing algorithms on reconfigurable architecture using C-based hardware descriptive languages, 2006, Int. J. Theor. Appl. Comp, vol. 1, no. 1, pp. 9–34.
12. J. Li, H. Zhang, D. Yuan, M. Sun, Single image dehazing using the change of detail prior, 2015, Neurocomputing, vol. 156, pp. 1–11.

13. P. Irgens, C. Bader, T. Lé, D. Saxena, C. Ababei, An efficient and cost effective FPGA based implementation of the Viola-Jones face detection algorithm, 2017, Hardware X, no.1, pp. 68–75.
14. B. Schrauwen, M. D'Haene, D. Verstraeten, J.V. Campenhout, Compact hardware liquid state machines on FPGA for real-time speech recognition, 2008, Neural Networks, vol. 21 pp. 511–523.
15. K.F.C. Yiua, Z.B. Li, S.Y. Low, S. Nordholm, FPGA multi-filter system for speech enhancement via multi-criteria optimization, 2014, Applied Soft Computing, vol. 21, pp. 533–541.