Mesopotamian journal of Computer Science
Vol. **(2021)**, 2021, **pp**. 18–32
DOI: https://doi.org/10.58496/MJCSC/2021/004 ; ISSN: 2958-6631
https://mesopotamian.press/journals/index.php/cs

Research Article

Character Recognition Techniques and approaches: a literature review

Mostafa Abdulghfoor Mohammed[1,*], Shuyuan Yang [2]

[1] *Great Imam University College, Baghdad, 10053, Iraq*

[2] *Deptartment of Electrical and Computer Engineering ,Xidian Univ, Xi'an,Shannxi , China, 710071*

ARTICLE INFO

Article History

Received 01 Nov 2020
Accepted 28 Dec 2020
Published 15 Jan 2021

Keywords

Neural Network
FPGA
OCR
ANN
VHDL

ABSTRACT

Researchers have carried out many approaches to recognizing the OCR through software-based and FPGA-based and its relationship with the ANNs and training these NNs by using the BP algorithm. The FPGA supports high speed to recognize the character because it works in parallel. It involves many logic circuits and can process at the same time (clock cycle). There have been problems with the established system because the size, shape, and style of supposedly identical characters might differ from person to person and even within the same person on rare occasions. The photograph is vulnerable to noise and can lose some or all of its visual clarity as a result. There are strict guidelines that govern the way a character looks that the character must adhere to. The NN and its learning process could only be defined by heuristically deducing its rules from samples and the enormous dataset.

1. INTRODUCTION

Many people are interested in concepts with biological roots, like NNs, evolution, and learning. The creation of complex, intelligent systems that can learn and adapt has received a lot of attention lately. These biological concepts have been studied by researchers and scientists in an effort to incorporate their desirable qualities into computing systems [1]. Field-programmable gate arrays (FPGAs) are a highly promising technology due to the demands for higher degrees of integration, more intensive on-board computing, and reduced power consumption[2].

ANNs are a creation whose ideas grew from brain research and were later used in digital computers. It was in 2007 that [3] introduced the first models of ANNs, which he called "neural networks" based on the brain's arithmetic rule. These studies demonstrate that neurons in ANNs get data as input from various sources, including other neurons and external data. The dissemination of this information takes the shape of a non-linear performance whose outcome is a weighted sum of inputs. The architecture of an ANN, including the amount of inputs each neuron gets and the frequency of conduction in every neuron, is highly task-dependent. As a result, for successful implementation, it is essential to strike a balance between many parameters in special-purpose network topologies[4].

2. NEURAL COMPUTATION

The human brain which consists of concerning a hundred billion neurons that connected by concerning a under trillion connections as shown in Figure 1, is that a lot of difficult object in familiar universe. There has been a rise in interest in the NN industry in recent years. Warren McCulloch and Walter Pitts presented the first model of artificial neurons in 1943, and every decade since then, researchers have been developing fresh and increasingly refined approaches to improve upon that foundation[5].

*Corresponding author. Email: alqaisy86@gmail.com

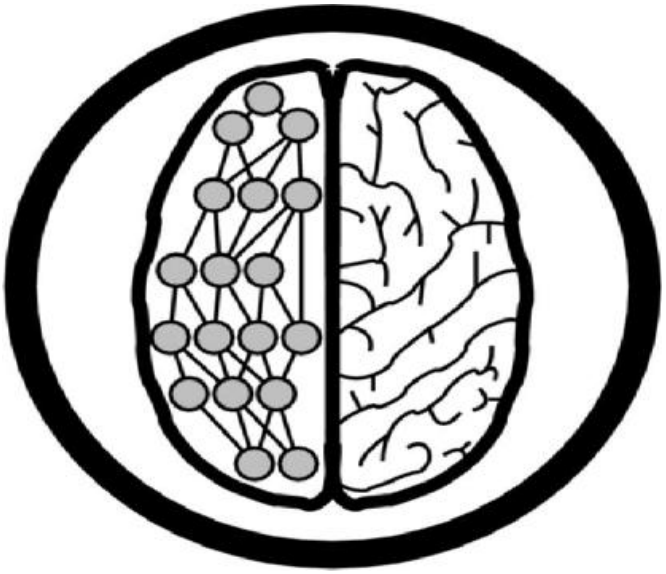


Fig.1. Mapping of Human Brain

Researchers examining the scientific and engineering challenges of understanding the brain and building computers can benefit from the insights provided by neural computation, as depicted in Figure 2. It provides context for common issues and techniques in brain modelling and the design and construction of neurally-inspired scientific discipline systems. Comprehensive coverage of neural computation is provided through an array of article types, including full-length research pieces, timely brief communications, and reviews highlighting advances in the field [6].

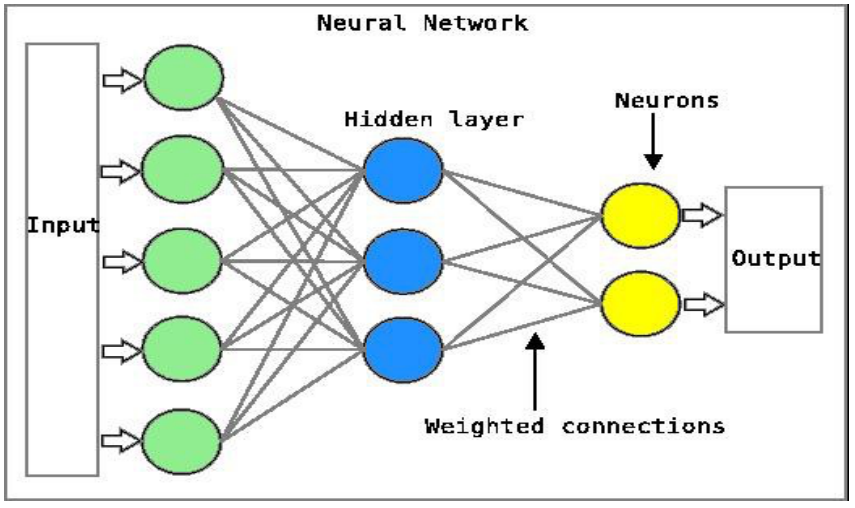


Fig.2. Structure of Neural Network

Authors propose a new method based on NN by generating a 20x20 character matrix (Figure 3). By primarily employing feed forward (FF) and backpropagation (BP) with adjustable learning rate and momentum, it produces a network structure that is both effective and efficient. It is clear from Figure 4 that the NN architecture requires 400 inputs and 26 neurons in its output layer in order to correctly recognize the letters. The network has two layers: log-sigmoid/log-sigmoid. A character matrix is an array of white and black pixels, represented as a vector of 1 diagrammatic by black and 0 by white. They are generated automatically by the user in any size or font conceivable; also, various typefaces of the same alphabet may be utilized beneath different training sessions. The network was then trained many times on different input vectors. In this scenario, training a network on multiple sets of noisy vectors challenged the network to know how to cope with noise. The limitation of this strategy is that the big matrix requires time-consuming processing [7].

First Font	Second Font
00000000000000000000	00000000000000000000
00000000011000000000	00111111111111111100
00000000111100000000	00111111111111111100
00000001111110000000	00111111111111111100
00000011111111000000	00111000000000011100
00000111000011100000	00111000000000011100
00001110000001110000	00111000000000011100
00011100000000111000	00111000000000011100
00111000000000011100	00111111111111111100
00111000000000011100	00111111111111111100
00111000000000011100	00111000000000011100
00111111111111111100	00111000000000011100
00111111111111111100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00111000000000011100
00111000000000011100	00000000000000000000

Fig.3. Character Matrix A of Different Fonts [7]

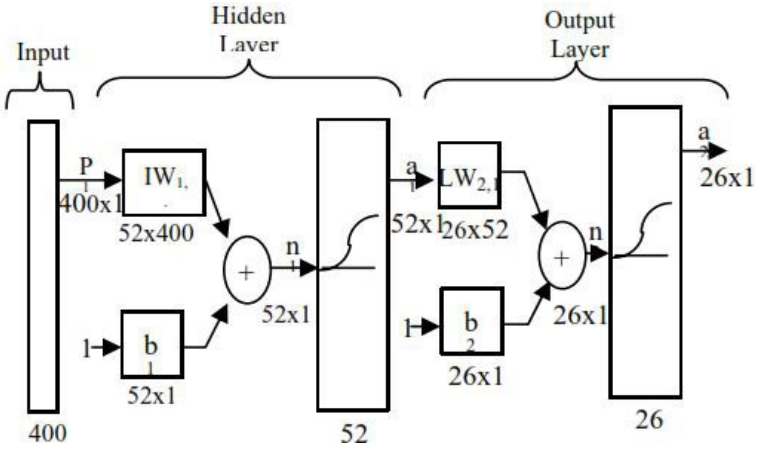


Fig.4. Neural Network Architecture [7]

An author offered a hand-printed character recognition method employing extracted characteristics and a sector approach. This method separates the normalized and thinned character picture into four sectors, each consisting of three segments covering a specified angle (Figure 5). Backpropagation neural networks (BPNN) and fuzzy logic techniques use this technology's sector-based feature extraction method. The characteristics consist of vector distances and angles that have been standardized. This proposed recognition algorithm can accommodate 15 to 20 percent character form diversity. Fuzzy logic is the superior of the two methods when considering computing time and relative risk or risk ratio (RR) as the desired factors. An extensive database is necessary for BPNN to learn and for the fuzzy logic approach to have a representative knowledge base. In recent years, character identification in noisy conditions has garnered the attention of numerous researchers, who have employed various strategies to address the issue. The ANN-based system has a relatively high identification accuracy at low noise levels; however, as the noise level increases due to training with large data sets, the recognition of characters are negatively impacted[8].

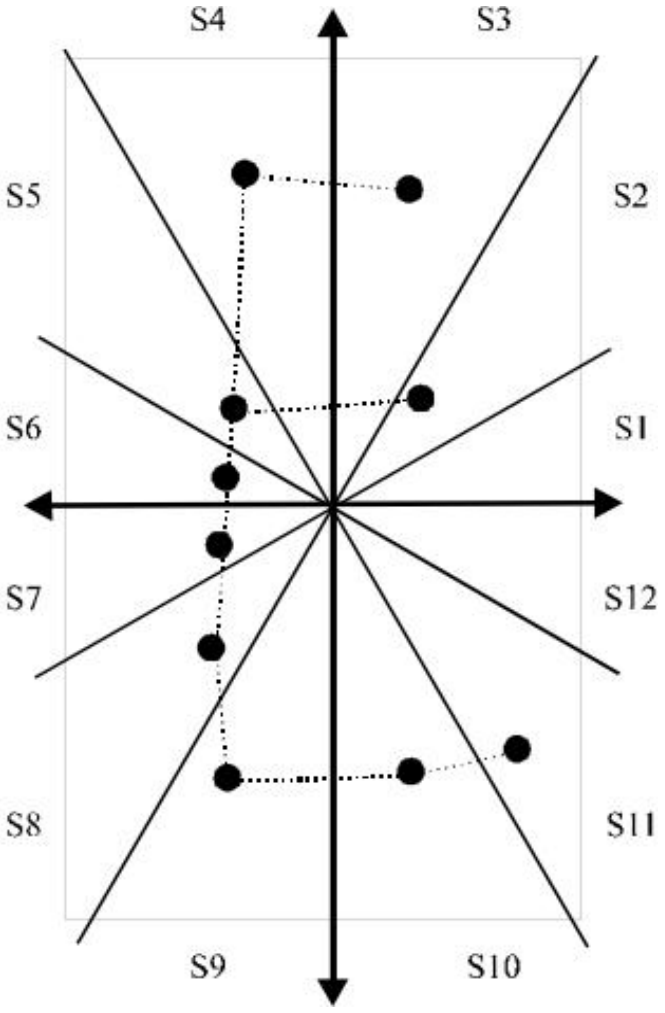


Fig.5. Reconstructed the letter 'E' based on features [8]

Other studies discussed the utility of neural networks in constructing intelligent systems. In addition, the special incentives for developing NNs include structures and learning processes to evaluate the NN application via the character recognition technique. The author addressed how to examine character recognition utilizing a variety of classes and classification procedures, including the principles of neural networks, strategies for supervised learning, and the three major classes of the learning procedure [9].

3. BACK PROPAGATION (BP) ALGORITHM WITH ARTIFICAL NEURAL NETWORKS (ANNs)

As depicted in Figure 6, ANNs are machine models inspired by the principles of computations done by biological NNs in the brain. NNs contain a number of enticing features that will ultimately allow them to exceed a number of the limitations of conventional computation methods[10].

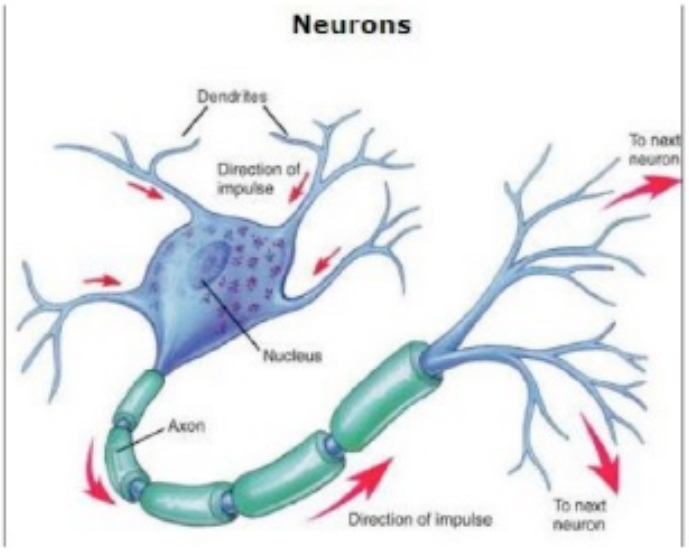


Fig.6. Neuron Components

The roles of NNs within the applications fall generally into two classes, namely pattern recognition and practical approximation. The basic objective of pattern recognition is to produce a substantive categorization of input patterns. In practical approximation, the network detection algorithm approximates the specific mapping between input and output given a collection of patterns. Then ANNs (Figure 7) are not just the networks but they considered of primitive functions, completely different models of ANNs differ principally within the assumptions concerning the primitive functions used, the correlation pattern, and therefore temporal arrangement of the transportation of data. Therefore, three elements are essentially substantial in any model of ANNs:

- The composition of the nodes.
- The architecture of the network.
- The learning algorithm used to determine the network's weights.

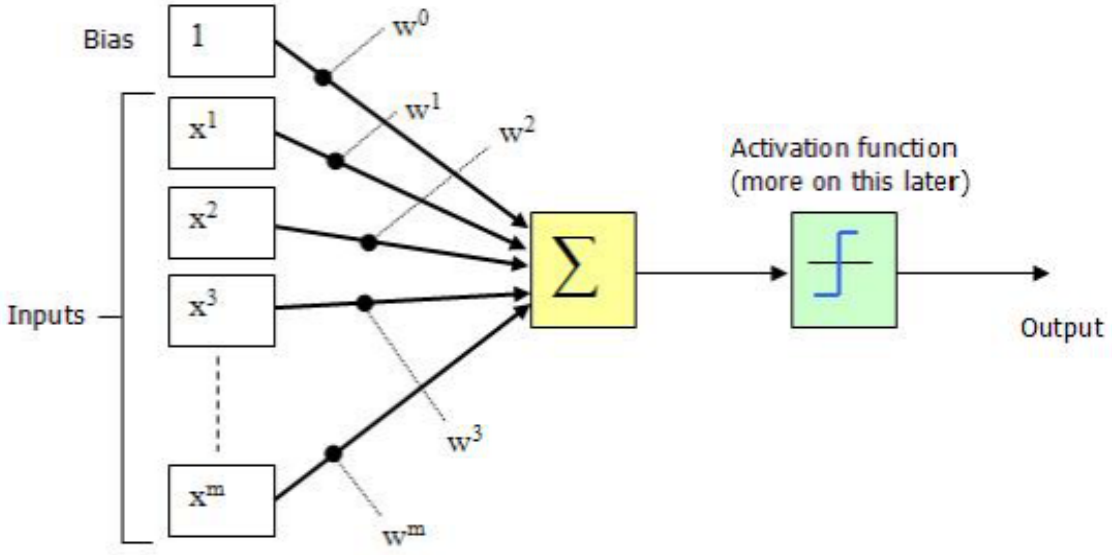


Fig.7. An Artificial Neuron

A study developed a system able to recognize handwritten characters or symbols; the objective is to create a handwritten character recognition system. the author uses several different digital image processing techniques to clear the noise, and then the image is processed to 32x32 pixels for normalization. Skeletonization is removing extra pixels from a character to reveal only its skeleton system must accommodate transformations of scaling, translation, or both. Barve (2012) devised the OCR based on ANNs. A neural network (NN) is a massively parallel-distributed processor composed of simple processing units, which has a natural propensity for storing experimental information and making it accessible for users. Figure 8 depicts the fundamental procedures involved in optical character recognition (OCR): document page analysis, preprocessing, image acquisition, feature extraction, training, and recognition[11].

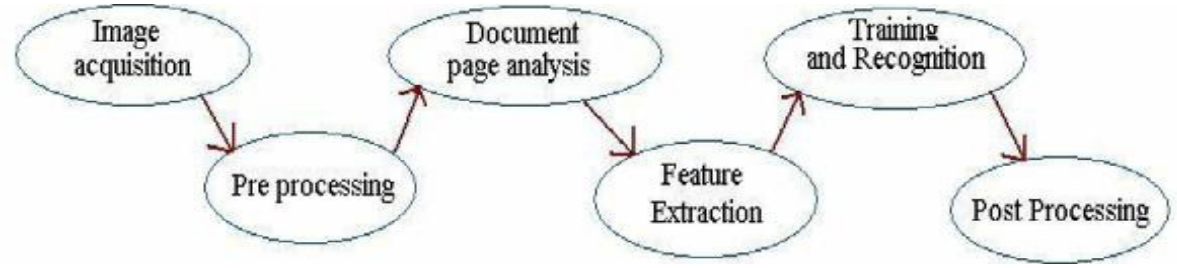


Fig.8. Step of OCR [12]

4. FIELD PROGRAMMABLE GATE ARRAYS (FPGA)

FPGAs are devices containing several building components, such as ports and logics, and a complete interconnected architecture (Figure 9). By coding these connections, several logical blocks or slices can be coupled to create the desired circuit [13].

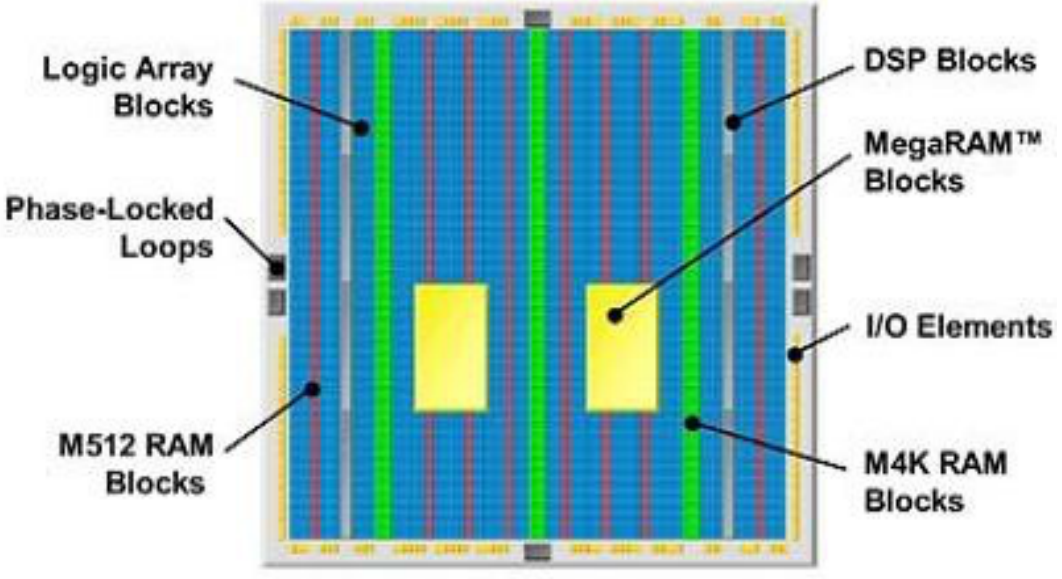


Fig. 9. Altera FPGA Structure

As depicted in Figure 9, the FPGA is a specialized IC that is frequently employed for prototyping.
Each FPGA consists of three primary elements [14].

1- The most important component is the Configurable Logic Block (CLB). CLB supplies the FPGA-downloaded software with physical support.
2- The Input Output Block (IOB) offers input and output for the FPGA and enables communication with the outside world.
3- The Programmable Interconnect (PI) that connects and enables communication between the various FPGA components.

Due to the high amount of computations required by NN applications, especially during the training phase, numerous rapid implementations of NN are being developed [15]. Several similar devices, including massively parallel computers, neuro-computers, and analogue or digital ASICs, are utilized for these rapid implementations. Programmable digital circuits like FPGAs combine hardware devices' performance and parallelism granularity with the adaptability of code implementations. They appear to be particularly well-suited to the typical requirements of neural implementation [16].

5. NEURAL NETWORKS ON FPGAs: SPECIFIC ASSETS

ANN research is becoming increasingly interested in FPGAs that are primarily ANN-based. In accordance with the parallelism and unique characteristics of NN, the developed system is greater than the software approach since it will take advantage of these characteristics. Moreover, since FPGA is a digital device with programmable properties and high flexibility, several academics have made substantial efforts to implement NN using FPGA technology [17]. As represented on FPGAs, hardware implementations can be made cost-effectively, simply, and adaptable. FPGAs provide a low-cost, simple, and adaptable option for neural applications and their many specialized benefits. In addition, they have several particular features for neural implementations [10], like the following:

- Reprogrammable FPGAs allow prototyping: in most applications, multiple neural structures must be evaluated to choose the most effective. This can be achieved directly with the hardware effectiveness of an FPGA-based approach at no extra cost. In addition, a chip with a solid design that has been implemented can be replaced with a better architecture without the need to build a new chip.
- On-chip learning is frequently viewed as difficult and ineffective. Indeed, it is rarely employed. However, on-chip learning typically reduces the efficiency of a hardware implementation since it requires particular operators, such as a high level of accuracy. Consequently, off-chip learning is selected when dynamic learning is not required. In a reconfigurable FPGA, it is possible to do on-chip learning prior to a specific, optimized implementation of the learnt NN using the same chip.
- FPGAs could be employed for embedded applications when the robustness and easiness of neural computations are required most, even for small-scale productions.
- FPGA-based solutions can be transferred onto new and upgraded FPGAs, which is a significant advantage given that FPGA speeds and areas roughly double yearly. Large NNs may soon be constructed on a single FPGA if the implementation method is sufficiently scalable. The FPGA approach is a significant improvement in ensuring the robustness of direct hardware mappings of NNs.

A recent study suggested using the implementation of NNs from hardware implementation instead of software package implementation on serial machines; as a result, they are swift due to the significant correspondence inherent within the hardware devices. Additionally, programmable gate arrays offer extra flexibility over the sector since these devices are unit reconfigurable, and their style will be altered whenever required. This methodology used one layer perceptron network with offline training to acknowledge three sets of four characters. In order to do this, it used Verilog HDL implemented on the Flex EPF10K70RC240 FPGA chip. By taking advantage of the bipolar input illustration, the researcher was ready to avoid the employment of multipliers within the style by merely passing on positive and negative weights. An OCR application of FPGA-based NN was a completed method to avoid multipliers using the methods described. The characters here are unit nation language alphabets that are input units to the network and when training process. These alphabet units are represented diagrammatically as a 5x5 matrix, as shown in (Figure 10) which indicates that there are twenty-five input neurons in the NN's input layer. In FPGA, higher-speed devices with a more significant number of logic gates are utilized. By utilizing the run-time reconfiguration capabilities, they can give increased flexibility and reliability[18].

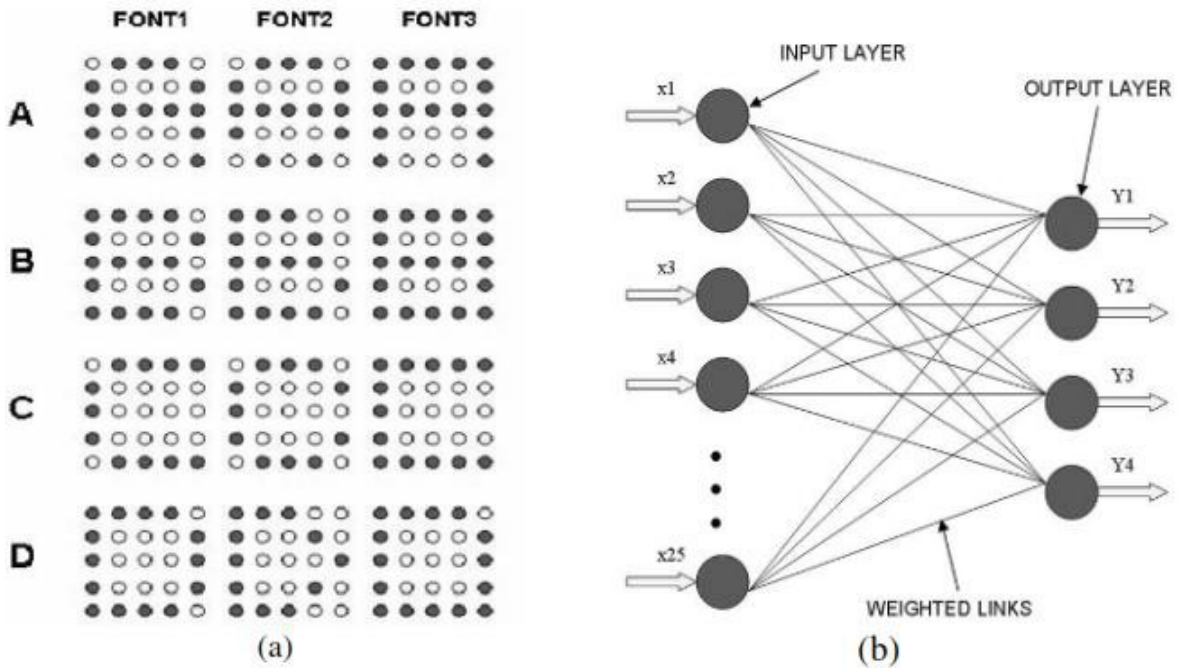


Fig.10. (a) Patterns of input with three distinct fonts for every character. (b) Used Perceptron model for character recognition issue[18].

The method proposed by [19] is utilized to extract attributes from 40x40 pixel normalized handwritten digit images. Using a two-layer MLP ANN with a small number of neurons in the hidden layer, this approach is applied to a Farsi digit recognition system. Various implementation strategies of the MLP on FPGAs have been utilized, and scholars have examined the implementation phase of the MLP decision. The system is simple and easy, more accurate, and less complex than other comparable systems. This approach of feature extraction is suitable for FPGA implementation because it can be implemented using simply addition and subtraction. Utilizing this method, a more precise and rapid recognition system can be achieved than with software-based implementation. The method of feature extraction combined of three approaches namely,

1) Utilizing a statistical method, the spatial distribution of pixel values in a binary image is represented. (Figure 11).

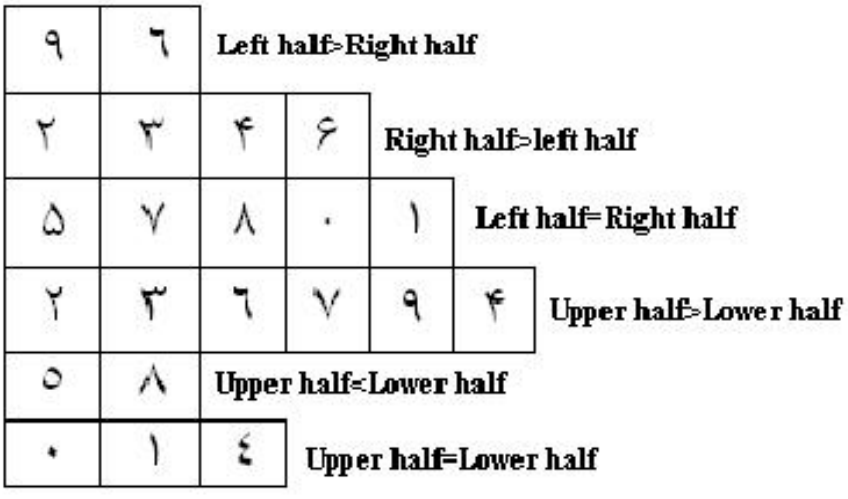
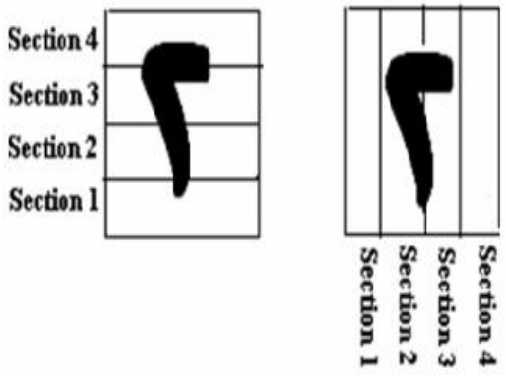
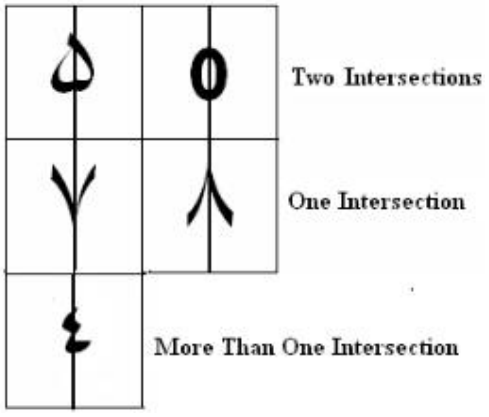


Fig.11. Compare left halves and right halves and upper halves and lowerhalves of digits [19]

2) Count the number of intersections along middle vertical ray and divide the pictures to eight sections (4 vertical,4 horizontal) as shown Figure 12.



(a)



(b)

Fig.12. a) Divide picture to four horizontal and four vertical section, b) Count the number of intersections along middle vertical ray [19]

3) Elastic Meshing Directional Feature Extraction/Multi-Layer Perception (MLP). Recognition system rate for testing data was 97.62% by using 16 hidden layer of classifier (Figure 13).

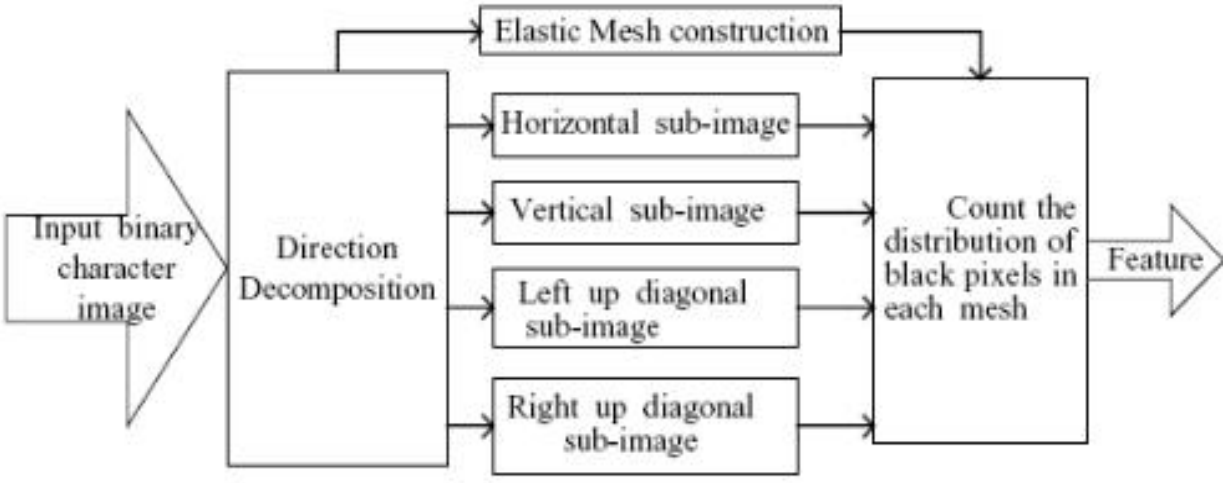


Fig.13. The block diagram of elastic meshing directional feature extraction (Moradi et al., 2009)

6. OPTICAL CHARACTER RECOGNITION (OCR)

An OCR is the process of converting visual handwritten or printed text, into a more usable electronic representation that can be indexed and searched. Optical Character Recognition is the method used to convert pictures of printed, written, or typed text into a machine-readable format. OCR is a matured field under which people have developed many complex and intelligent algorithms to extract characters from images. Therefore, hardly any contribution is made to the development of the field by the current researchers. Although the existing algorithms are much intelligent and robust in nature, they have the disadvantage of being not fast enough to be applied in real time applications. The basic reason behind this is the processing nature being sequential in microprocessor-based systems.

Many applications can be implemented for OCR. Nowadays, it can be found everywhere, from banks and post offices to airports and airline offices. Form readers can read and process up to 5,800 forms per hour; address readers sort incoming and outgoing mail; check readers in banking institutions capture images of checks for further processing; airline passport and ticket readers are used for a variety of purposes, including accounting for travelers profits and verifying database records [7]. Automating the data-entry process has been greatly aided by character recognition technologies. There are a plethora of character recognition systems available in the software. Up until recently, improved microprocessors were used for digital signal processing. Generic products have a large market because of their low price, but their performance could be better for many uses.

The researcher proposed a plan to construct a comprehensive OCR system for five fonts and sizes of Devanagari script (Indian letters) to utilize this system in the banking industry and corporate sectors. The OCR process includes a few stages: classification, feature extraction, segmentation, and preprocessing (Figure 14). In general, the researchers concluded that traditional OCR systems performed poorly with documents like old books (poor print and paper quality caused by ageing), copied documents (photocopies or faxes produced poor print quality), and periodicals (printed using low-quality paper in general).

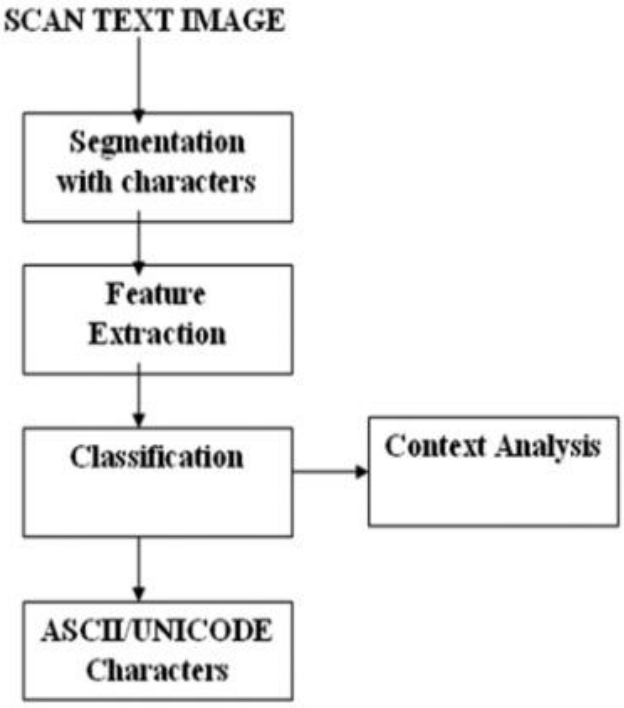


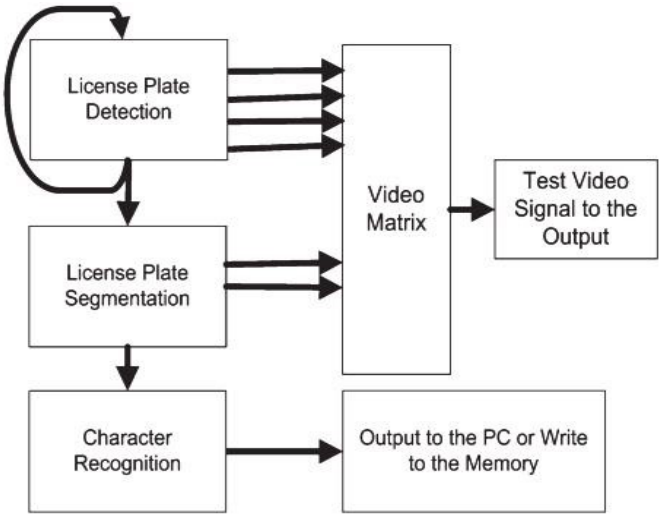
Fig.14. Stages in OCR Design [20]

The researchers demonstrated an automated OCR system that uses ANN, with the ANN accepting a matrix of 11x36 values and generating 36 values, one per character in the database. Using a template-matching strategy and relying on correlation coefficients based on the cross correlation of source characters for the same shape character, this method took into account the human psychology behind how characters are seen. Topological factors including shape, symmetry, open and closed areas, number of pixels, and where confusion lies in character identification are all taken into account by this approach. This technique, also known as Computer Vision, is used to train a system to "read" specific data [21].

[22] Tried to present how to recognize character recognition by a video processing methodology for an FPGA-based license plate recognition (LPR). The LPR system in hardware that is suitable for real-time applications consists of the following three major phases:

1) License plate detection
2) License plate character segmentation
3) Character recognition

During license plate detection, the authors use a variety of algorithms to recognize the license plates' color, form, and texture. In the detection phase, it employs the Gabor filter and the Hough transform. For training a massive number of sample characters, the NN is utilized to recognize the characters. Template matching involves a collection of diverse character fonts and weights. Self-organizing maps (SOM) have found application in various fields, including optical character recognition (OCR). Using two layers dubbed the input layer and the computation layer, the SOM is a mechanism that produces similar outputs for comparable inputs. Processing units are found in the computation layer. In the course of learning, the SOM weight matrix is calculated. The devised apparatus selects the output character according to the hamming distance between both the weight matrices of all neurons and the input picture. (Figure 15). The authors are aware that the memory requirements are mostly meant to be remarkably low, making use of smaller, and hence less expensive, and options possible. In remote places, where bulky data-gathering systems are impractical, this design is preferred due to its quick response time, portability, and low cost. This cost-effective dedicated hardware machine can transmit only the extracted text information over wireless data networks rather than extensive image data, lowering operational expenses.

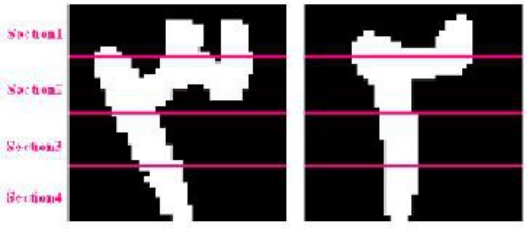


(a)

(b)

Fig.15. a) Embedded LPR system block diagram, b) Hardware block diagram [22]

he authors present a method to extract features from standardized handwritten digit photographs from the Standard Hoda database, with each image having 40x40 pixels of normalized data. A fresh and efficient approach was used to construct recognition systems using an FPGA chip. The simplicity of this method is represented by using only addition and subtraction instead of multiplication. This method was implemented by developing VHDL coding and ten parallel processes. The author used three types of feature extraction, as shown in Figure 16. A statistical approach is used to represent the spatial distribution of the pixel values of binary images and Count the number of Intersections. Experimentally results showed that the proposed system achieved about 96% accuracy [19].



(a)

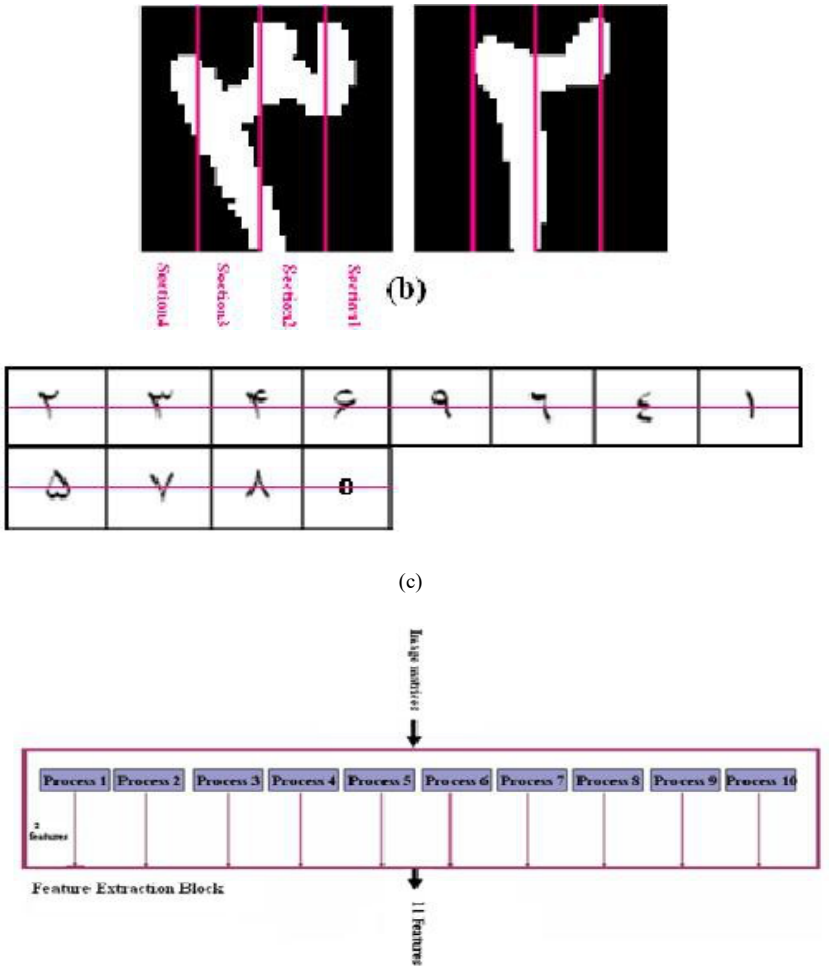


Fig.16. (a) & (b) Pictures are divided to four horizontal and four vertical sections, (c) Count the number of intersections along horizontal ray of and written and typed images, (d) The block diagram of feature extraction block [19]

The authors detailed offline methods for constructing solutions for each handwritten English character (A-Z) and how to incorporate them into a larger whole (0 to 9). The OCR process benefits from this improved method. The figure 17 depicts the application of binarization, thresholding, and segmentation methods in character preprocessing. This approach required four stages of character recognition prior to processing. In the initial preprocessing stage, the colour image is converted to grayscale, which produces a noisy version of the original to remove the displayed noise; filtering is performed next. Afterwards, an edge detection method, such as those for identifying colour edges, grey level edges, and texture edges. Extracting features is the final stage. They were chosen with care since it is anticipated that the feature set will extract the necessary information from the input data to carry out the required task with this condensed representation rather than the full-size input. After that, FF and BP techniques were employed to sort the data and train the network to identify the characters.

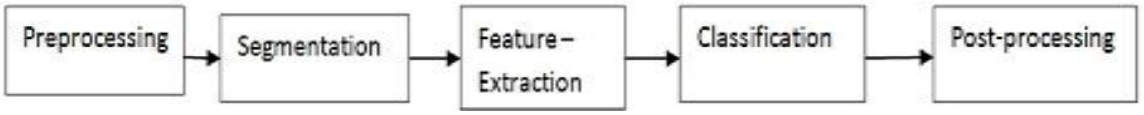


Fig.17. Block diagram of sequence recognition character

7. CONCLUSION

Recent years have seen the creation of the fantastic FPGA family of platforms, which has sparked a revolution in electronic device production. These platforms are the finest option for running cutting-edge digital systems. Given its parallel nature, a neural network (NN) has the potential to perform computations quickly for specific applications. The same quality also makes NN a good candidate for VLSI technology. The successful implementation of even a single neuron is crucial to the overall success of any hardware implementation of a NN. Reconfigurable computing architectures like FPGAs are well-suited for NN hardware implementation. The implementation of high-neuron-count ANNs in FPGAs is still being determined. When it comes to the developed system, many authors have run into issues because how they write the "identical" characters might vary in size, form, and style from one person to the next and even throughout a single writing session. Noise is a problem for visual characters, just like it is for any other kind of image. Strict guidelines govern the character's behavior and the manifestation of a visible identity. Therefore, the NN's definition and learning process required rules to be determined from samples and the massive database.

Conflicts Of Interest

The author declares no conflict of interest in relation to the research presented in the paper.

Funding

The author's paper explicitly states that no funding was received from any institution or sponsor.

Acknowledgment

The author would like to express gratitude to the institution for their invaluable support throughout this research project.

References

[1] H. A. Akkar, and F. R. J. I. J. A. C. T. Mahdi, “Implementation of Digital Circuits Using Neuro-Swarm Based on FPGA,” vol. 2, no. 2, pp. 64-78, 2010.
[2] B. M. J. E. Khammas, and T. Journal, “Design Neural Wireless Sensor Network Using FPGA,” vol. 30, no. 9, pp. 1641-1661, 2012.
[3] S. Haykin, and N. J. N. n. Network, “A comprehensive foundation,” vol. 2, no. 2004, pp. 41, 2004.
[4] S. Sahin, Y. Becerikli, and S. Yazici, "Neural network implementation in hardware using FPGAs." pp. 1105-1112.
[5] C. J. a. p. c. Gershenson, “Artificial neural networks for beginners,” 2003.
[6] G. Hinton, and T. J. Sejnowski, *Unsupervised learning: foundations of neural computation*: MIT press, 1999.
[7] F. Mamedov, and J. F. A. Hasna, "Character Recognition using Neural Networks." pp. 728-733.
[8] V. K. Madasu, B. C. Lovell, and M. J. S. o. I. Hanmandlu, University of Queensland, Australia, “Hand printed character recognition using neural networks,” 2005.
[9] E. Ganea, M. Brezovan, and S. J. S. Ganea, “Character Recognition using Neural Networks,” 2005.
[10] A. R. Omondi, and J. C. Rajapakse, *FPGA implementations of neural networks*: Springer, 2006.
[11] V. Dutt, and S. J. A. i. C. Dutt, “Hand written character recognition using artificial neural network,” vol. 1, no. 1, pp. 18-23, 2011.
[12] S. J. I. J. o. A. R. i. C. E. Barve, and Technology, “Optical character recognition using artificial neural network,” vol. 1, no. 4, pp. 131-133, 2012.
[13] P. Pelgrims, D. Driessens, and T. Tierens, “Embedded Systeemontwerp op basis van Soft-en Hardcore FPGA’s,” 2003.
[14] R. D. J. J. o. e. Abdu-Aljabar, and development, “Design and implementation of neural network in FPGA,” vol. 16, no. 3, pp. 73-90, 2012.

[15] T. Nordström, B. J. J. o. p. Svensson, and d. computing, “Using and designing massively parallel computers for artificial neural networks,” vol. 14, no. 3, pp. 260-285, 1992.
[16] B. Girau, "Neural networks on FPGAs: a survey."
[17] J. Liu, and D. Liang, "A survey of FPGA-based hardware implementation of ANNs." pp. 915-918.
[18] F. A. Khan, M. Uppal, W.-C. Song, M.-J. Kang, and A. M. Mirza, "FPGA Implementation of a Neural Network for Character Recognition," *Advances in Neural Networks - ISNN 2006.* pp. 1357-1365.
[19] M. Moradi, M. A. Poormina, and F. Razzazi, "FPGA implementation of feature extraction and MLP neural network classifier for Farsi handwritten digit recognition." pp. 231-234.
[20] R. Singh, C. Yadav, P. Verma, V. J. I. J. o. C. S. Yadav, and Communication, “Optical character recognition (OCR) for printed devnagari script using artificial neural network,” vol. 1, no. 1, pp. 91-95, 2010.
[21] V. Shrivastava, and N. J. a. p. a. Sharma, “Artificial neural network based optical character recognition,” 2012.
[22] H. Caner, H. S. Gecim, and A. Z. J. I. T. o. V. T. Alkar, “Efficient embedded neural-network-based license plate recognition system,” vol. 57, no. 5, pp. 2675-2683, 2008.