IRIS MATCHING STEP IMPLEMENTATION IN FPGA

Aumama M. Farhan[1], **M. F. Al-Gailani** [2]
[1,2] College of Information Engineering, Al-Nahrain University,Baghdad, Iraq
aaumama@gmail.com[1], m.falih@coie-nahrain.edu.iq[2]
Received:8/5/2019, Accepted:24/5/2019

***Abstract*- Iris recognition system has been recently widely used as it is in the forefront of other biometric systems since it contains distinctive patterns that give it a powerful strategy to distinguish between persons for identification purposes. However, implementation of this system requires large memory capacity and high computational power due to the size of the data and the processes on which it is implemented. These factors make the challenge to find a way for running this algorithm in a hardware platform. Efficient design in hardware reduces the execution time by exploiting the parallelism and pipeline architecture. The present work addressed this issue and the execution time was actually reduced when implementing iris matching step using hamming distance algorithm on the target device FPGA KINTEX 7 utilizing Xilinx system generator. The obtained result demonstrates that the execution time has been accelerated to 1.32 ns, which is almost at least four times faster than the existing works.**

keywords: **Hamming distance, System generator, matching, FPGA, iris.**

I. INTRODUCTION

To access certain secure application or to securely save or transmit data, traditional methods require entering a password or secret keys. Unfortunately, this confidential information is vulnerable to attack or forgetfulness. This is the way the modern trend is toward the use of biometrics[1] Biometrics identify people based on their physiological features like face, iris, and fingerprint, or on their behavioral characteristics like signatures and handwriting[2], [3]. Between all biometric systems, iris is considered as one of the best reliable biometrics since it has unique textures that remain unchanged over life[4]. Iris recognition system consists of the following steps, image acquisition, iris segmentation, normalization, feature extraction and matching[5]. The execution speed of the system is low due to the large computation which is one of the most important problems that limit its use. However, to overcome this problem and verify real-time processing FPGA has been used which offers high flexibility, scalability, parallel processing, and pipeline structure. Thus, data can be processed on time with minimal power consumption[6].

II. RELATED WORK

In this section, a number of recent published researches within the subject of the research topic are reviewed, as follows. In 2009, [7]authors presented a novel architecture design to parallelize the segmentation, features extraction, and matching steps of the iris recognition system using FPGA. In addition, the authors have made a comparison in the performance of these parts when implemented them in CPU and FPGA, the time to implement the matching step in Cyclone II EEP2C35 was 20ns.

In 2011, [8] authors implemented iris recognition system in software using C++ language, then they improved the system and programmed it in VHDL language and synthesized using CycloneII EP2C35FPGA. The improved system consists of two modules, first module implemented using Hough Transform for segmentation and normalization steps, while the second used Gabor filter on the normalized iris and Hamming distance for calculating the similarity between images. The

required time to implement the matching step was $546\mu s$.

In 2015[9]authors provided an iris recognition system with two enhancements.First,enhancement by parallelizing the structure of the iris code using FPGA to speed up the performance of the system. The second by extracting the quarter part of the iris region. Only feature extraction and matching steps were implemented in FPGA using VHDL language. The synthesized time of Hamming distance algorithm was20ns.

In 2018, [10]author's suggested work, the preprocessing, segmentation,and normalization were implemented in MATLAB 2016a. Canny edge detection and CHT algorithms were used for segmentation ,Daugman's rubber sheet model for the normalization process.RED algorithm and HD were simulated by means of Xilinx ISE(14.7)and implemented on Virtex-5 FPGA(XC5VLX110T) to gain parallel processing and to reduce the time delay. The RED algorithm simulation delay time was $4.796\mu s$ and the HD synthesized time delay was 5.22ns.

III. IRIS RECOGNATION SYSTEM

The purpose of automatic speaker recognition is to The proposed system has two phases, first represents the enrollment operation to store the features of iris in the database. The second represents the identification operation that matches between the extracted features and those stored in the database. The steps of the proposed iris recognition system are shown in Fig. 1[11]

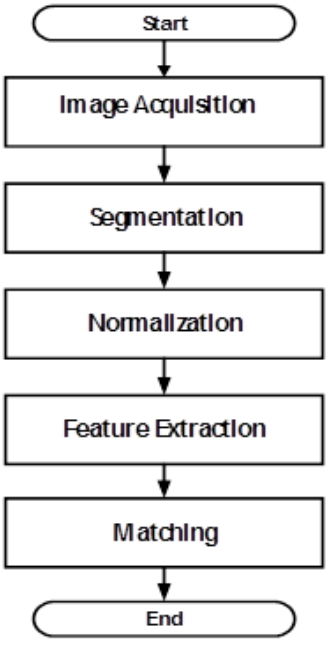


Figure 1: Iris recognition system Steps [11]

A. *Image acquisition*

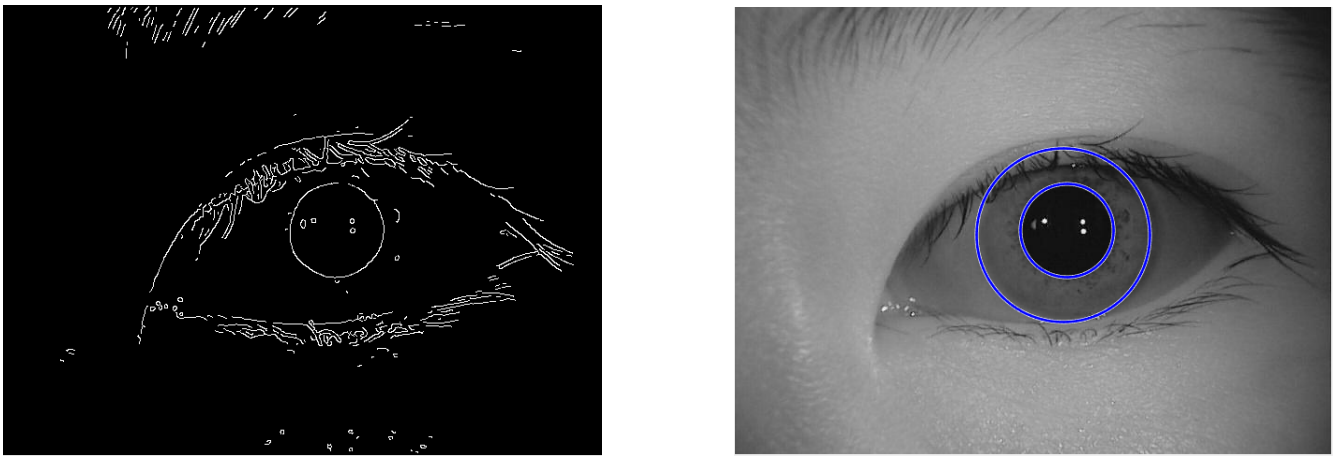
This step is used to insert the image of the eye either directly through the camera or through a database designed for this purpose. CASIA iris-lamp V4[12] was used, with image size of 640×480 pixels type.jpg.

B. Segmentation

This step is used to segment the iris region from the whole image of the eye. It is processed by using canny edge detection to detect the edge of the image, and circular Hough transform to detect the parameter of iris and pupil. These parameters represent the coordinates and radius of each circle, which is computed using equation 1

$$(x-a)^2 + (y-b)^2 = r^2 \tag{1}$$

Where r is the radius, (a,b) is the center of the circle, and (x,y) represents the circle coordinates [13]. The output of canny edge detection is shown in Fig. 2a and the output of the circular Hough transform is shown in Fig. 2b.



(a) (b)

Figure 2: The output of the segmentation step

C. Normalization

The output of this step produces the normalized iris with constant dimensions. Thus, two different images of the same iris under different conditions will have the same characteristic features[14]. In this step, our algorithm[15]which is based on Daugman's algorithm, is used where, the process is accelerated by dividing the iris region into two half-circles to be converted into rectangular shape, simultaneously. The upper rectangle depends on theta from 1 to 180 degree and the lower rectangle depends on theta from 181 to 360 degree, this is shown in Fig. 3.

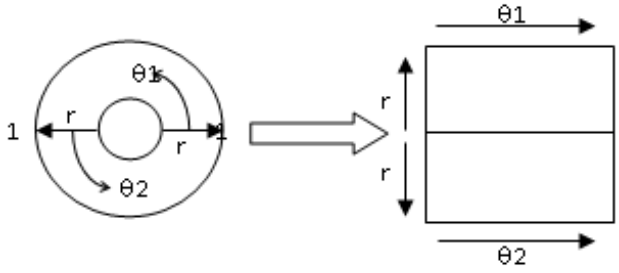


Figure 3: Converting iris ring into a rectangular shape [15]

The output of the normalization step is an image of size 130×40 pixels as shown in Fig. 4.

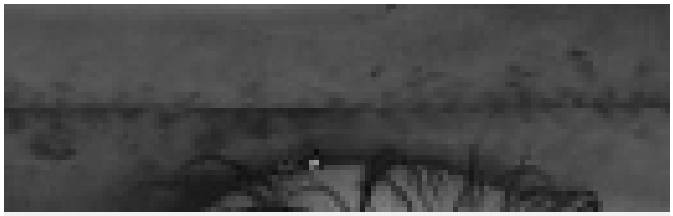
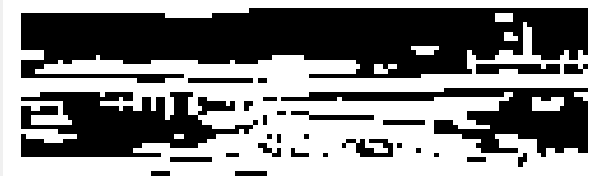


Figure 4: The output of the normalization step

D. Feature extraction

In this step, Ridge energy direction(RED)is implemented to find the existence ridges of the iris and their orientation. This algorithm depends on two(9×9) filters(vertical and horizontal)that are used to generate a template which represents the existence ridge from the iris and also used to generate a mask which represents the location of noise in the image[16]. The output of feature extraction is shown in Fig. 5, where the template of size (130×40) pixels is shown in Fig. 5a and the mask with the same template size is shown in Fig. 5b.

(a) Template



(b) Mask

Figure 5: The output of the feature extraction step

E. Matching

Hamming distance(HD) algorithm is used as a metric for matching between two images. Where the template and mask for each image are considered in the matching process to decide if the similarity between the images are exists. HD

equation[17]:

$$HD = \frac{\sum((TemplateA \oplus TemplateB) \bigcap maskA \bigcap maskB)}{\sum(maskA \bigcap maskB)} \quad (2)$$

Where$\oplus$: XOR operation, and $\bigcap$: binary and operation and $\sum$: indicates the summation of ones

IV. XILINX SYSTEM GENERATOR

Xilinx system generator (XSG) is a very useful tool for hardware design and implement. It is used to optimize the design and thus to minimize the delay and resources. XSG is important todays to facilitate hardware implementation [18]. To use XSG two software tools are required, MATLAB, and Xilinx Integrated Synthesis Environment (ISE) for synthesis and analysis the HDL designs. The Xilinx DSP blocks are used to build the system; they are existing in the library of MATLAB. Over 90 blocks are available in system generator which are used to design the systems [19].Fig. 6 shows the library of Xilinx block sets.

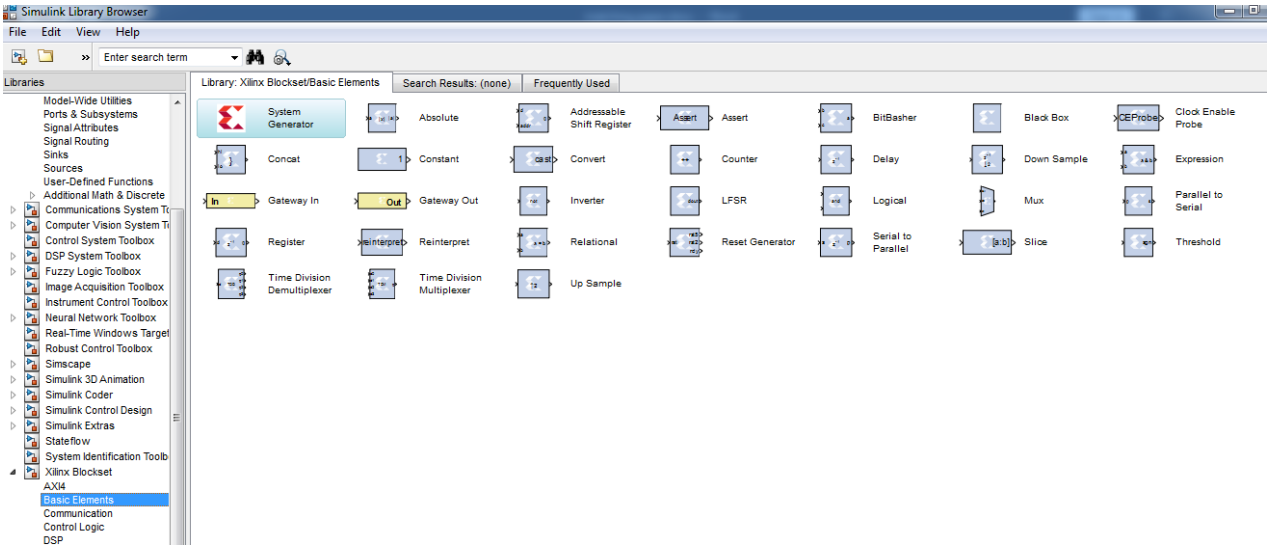


Figure 6: Library of simulink xilinx system generator

V. MATCHING STEPS IN XSG

The matching part of the iris recognition system is designed and implemented using XSG as follows:

A. *Image preprocessing blocks*

As an image is a two-dimensional matrix it should be converted to a one dimensional to meet the requirement of hardware. The flowchart of reading and preprocessing the image is shown in Fig. 7

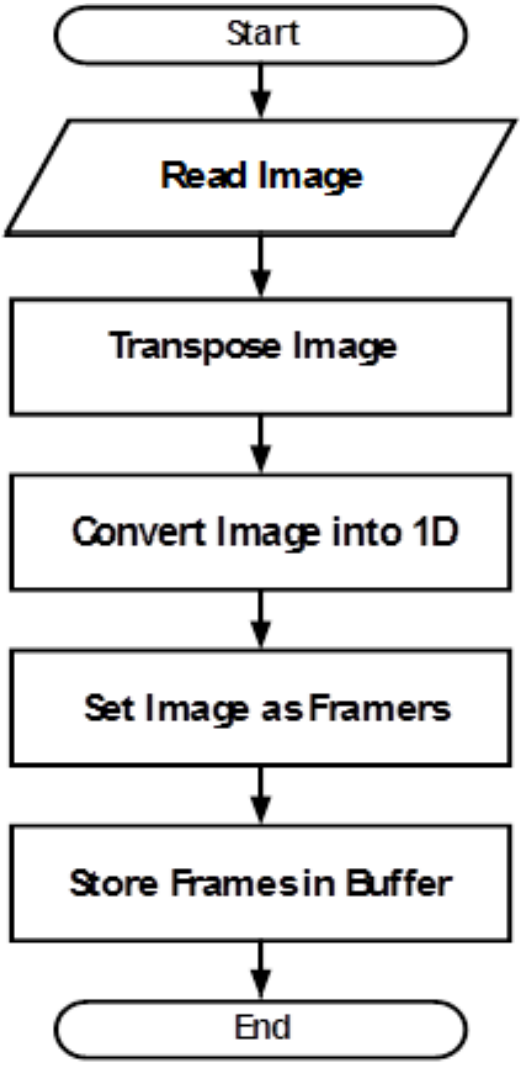


Figure 7: Image preprocessing

B. Hamming distance blocks

After reading the four images: Template A from the captured image and it's corresponding Mask A,and Template B from the database and it's corresponding Mask B, the following procedure which is shown in Fig. 8 is applied. First, the logical block is used to perform an XOR operation between template A and template B. Then, another logical block is used to do an AND operation between mask1 and mask2. The output of the AND operation is entered to the accumulator block to calculate the number of pixels that holds the value of one. After that, the output of the AND logical block and the XOR logical block are entered to the AND logical block, then to the accumulator block to calculate the numbers of pixels that their values are one. The output of the two accumulators is divided and the result is displayed in the display block. If the output of hamming distance is equal or less than 0.32, the two images are similar, otherwise, they are different[9].

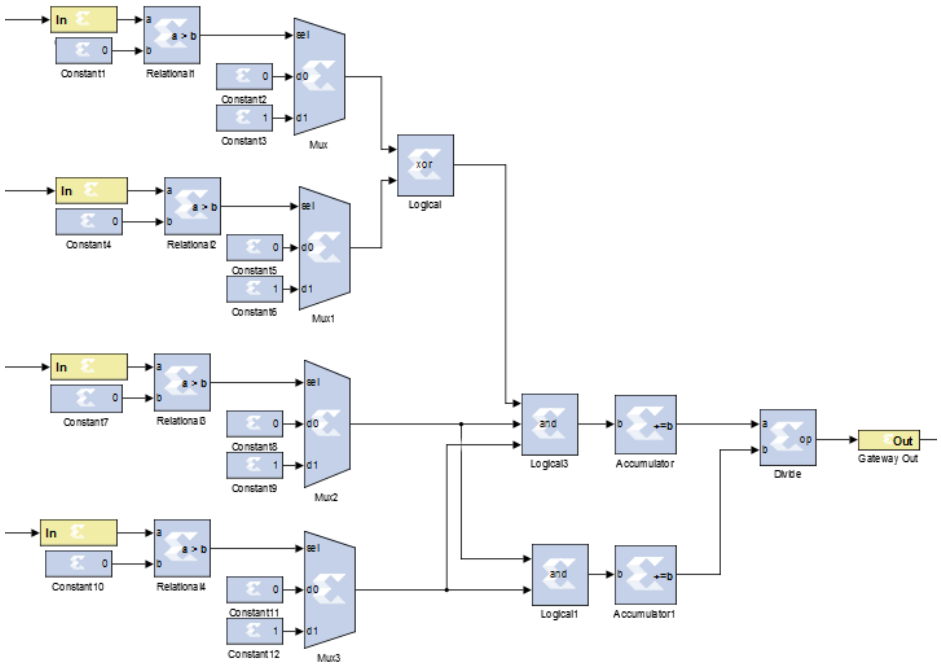


Figure 8: XSG element for hamming distance algorithm

The inputs to the system in Fig. 8 are the four images after preprocessing as shown in Fig. 9.

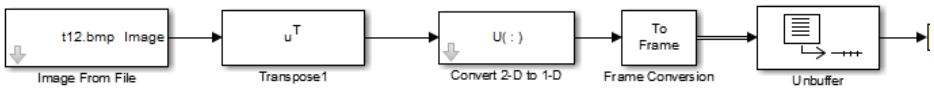


Figure 9: XSG image preprocessing

In Fig. 10 the output is displayed as 0 which means the images are similar.

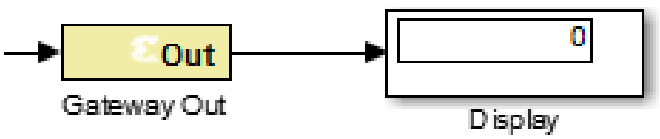


Figure 10: The output of the system module

After verifying the results, the token of the system generator is used to implement the design on the FPGA board by converting it into a netlist file after specifying the target device (Kintex7 xc7k325t-2ffg900) and the language (VHDL) as

shown in Fig. 11.

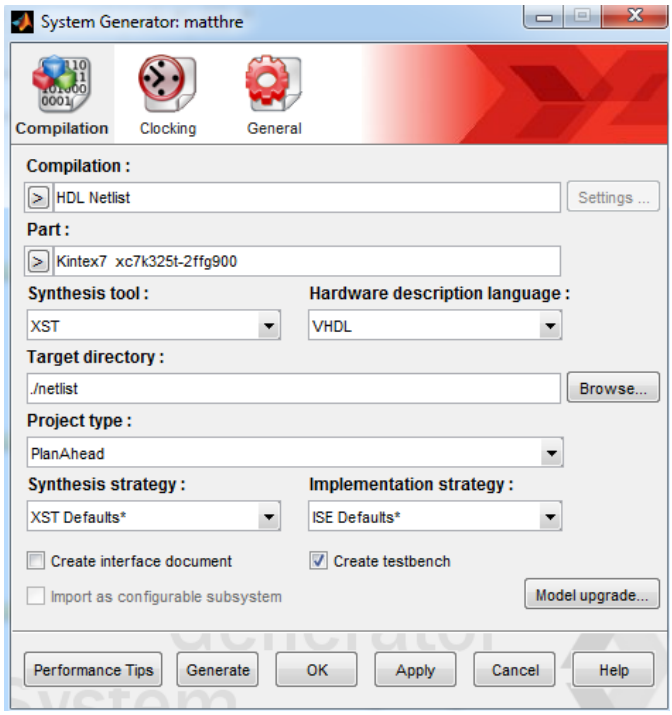


Figure 11: Token of system generator

VI. Results and Discussion

Hamming distance algorithm is implemented on the target device kintex7 xct325k using Xilinx system generator with MATLAB 2013b and ISE 14.7. The result of implementation shows the efficiency of the design where the implementation time is 1.32 ns while it was 0.171 sec in MATLAB. Table I depicts the utilization summary of the device, which shows the number of available and used registers, LUTs, IOBs, and buffers.

TABLE I
Device Utilization Summery

Device Utilization Summary(estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	32	407600	0%
Number of Slice LUTs	42	203800	0%
Number of fully used LUT-FF pairs	32	42	76%
Number of bonded IOBs	102	500	20%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig. 12 shows the Top level of RTL, where the four input images (Template A, Template B, Mask A, and Mask B) are entered the system through the gateway in, there is also a clock in the input to control the entered pixels from each image.

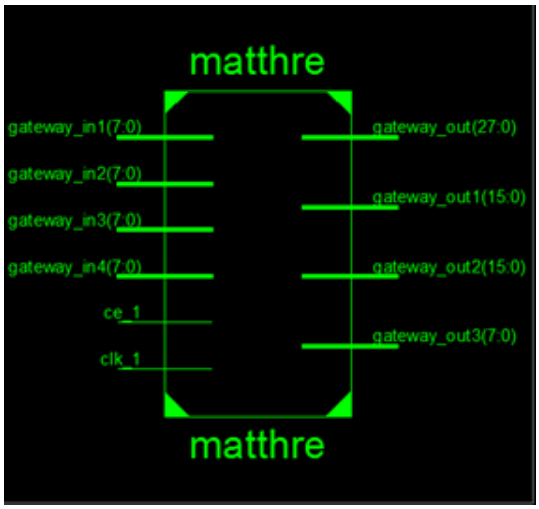


Figure 12: Top level RTL view of hamming distance algorithm

The internal level of hamming distance algorithm is shown in Fig. 13

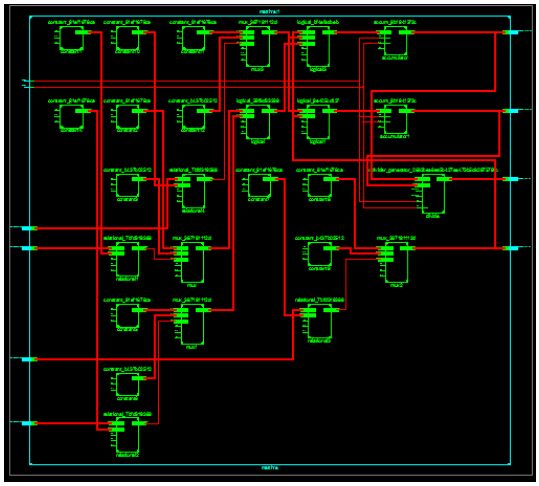


Figure 13: Internal level RTL view of hamming distance

In addition, the result of the implementation is compared to other works references[7], [9], [10]. Fig. 14 summarizes the results of comparisons, which is obviously clear that the proposed algorithm is faster than the others. Since the matching process include a comparison of each input image with images stored in a huge database that required large amount of memory. This process could be successfilly implemented in an FPGA hardware and the results of resources utilization in the target FPGA hardware show which the LUT-FF is 76%, while the bounded IOBs is 20% and BUFG/BUFGCTRL is 3%.

VII. CONCLUSIONS

Hamming distance algorithm is efficiently designed and implemented in hardware using XSG. The implementation result of the iris recognition system has been significantly accelerated compared to both software and related work.

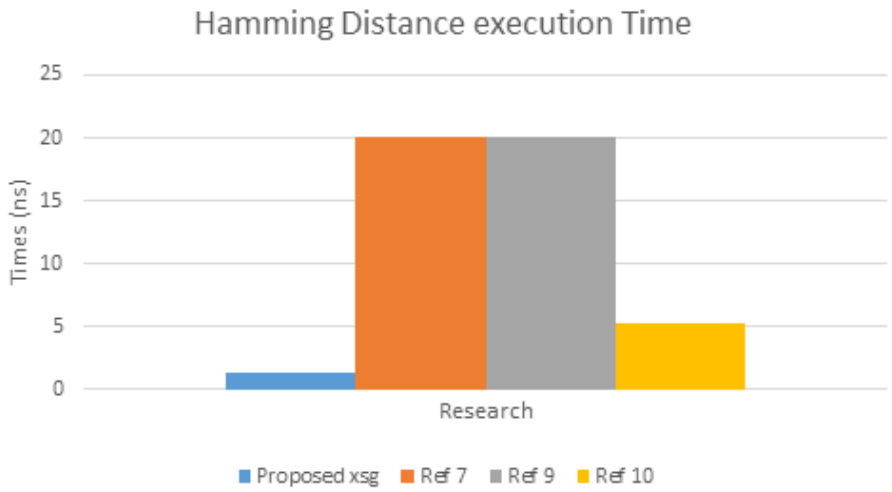


Figure 14: Execution time of hamming distance algorithm

REFERENCES

[1] I. Hamouchene and S. Aouat, "Efficient Approach For Iris Recognition, "Springer, vol. 10 ,no. 7, April 2016.
[2] R. Plaka and G. Kaushal, "Iris Authentication System Using Hybrid Technique-Surf, "International Journal for Technological Research In Engineering, vol.4, no.11, pp.2388-2390, July 2017.
[3] H. K. Rana, Sh. Azam and R.Akhtar, "Iris Recognition System using PCA based on DWT," SM Journal of Biometrics and Biostatistics, vol. 2, no. 3, pp. 1-5, Aug 2017.
[4] M. S. Singh and S. Singh, "Iris Segmentation Along with Noise Detection using Hough Transform,"International Journal of Engineering and Technical Research (IJETR), vol. 3, no. 5, 2015.
[5] R. Vyas, T. Kanumuri, and G. Sheoran, "Iris recognition using 2-D Gabor Filter and XOR-SUM Code, "India International Conference Inf. Process. IICIP 2016 -Proc,2017.
[6] J. Huang and G. Zhou, "On-board Detection and Matching of Feature Points, " Remote Sensing, vol. 9,no. 6, 2017.
[7] R. N. Rakvic, B. J. Ulis, R.P. Broussard, R.W.Ives, and N. Steiner, "Parallelizing Iris Recognition," IEEE Transactions on Information Forensics and Security, vol.4,no. 4, pp. 812 - 823, 2009.
[8] R. Hentati, M. Abid, and B. Dorizzi ,"Software implementation of the OSIRIS iris recognition algorithm in FPGA, "Proc. International Conference Microelectron. ICM, 2011.
[9] S.S. Omran and A. A. Al-Hillali, "Quarter of Iris Region Recognition using the RED Algorithm, "In 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim), IEEE.
[10] B. Uma and P. K. B,"Implementation of Iris Recognition System using FPGA, " vol. 25, no. 1, 2018.
[11] S. Vijayarani and M. Vinupriya, "Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining "International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 8, 2013.
[12] "Institute of Automation, Chinese Academy of Sciences" [Online] Available:"http://www.cbsr.ia.ac.cn/china/Iris%20Databases%20CH.asp".
[13] N. Cherabit, F. Zohra Chelali, and A. Djeradi, "Circular Hough Transform for Iris localization, "Science and Technology, vol. 2, no. 5, 2012.
[14] S. Mary James, "Iris Recognition Process, "Proceedings of the UGC Sponsored National Conference on Advanced Networking and Applications, March, 2015.
[15] A.Mohammed and M. F. Al-Gailani, "Developing Iris Recognition System based on Enhanced Normalization", SCCS, IEEE, Baghdad, 27-28 March, 2019.
[16] S. Omran and A. Al-Hilali, "Using an FPGA to Accelerate Iris Recognition" in International Conference on Advances in Software, Control and Mechanical Engineering, Antalya (Turkey), Sept. 7-8, 2015.
[17] Y. Jain and M. Juneja, "Ridge Energy Based Human Verification using Iris and Palm Print, "International Journal of Trend in Research and Development, vol.5, no. 2, pp. 30-36, Mar-Apr 2018.

[18] M. Dridi, M. Hajjaji and A. Mtibaa, "Hardware Implementation of Encryption Image Using Xilinx System Generator", 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering - STA 2016, Sousse, Tunisia, 2016.

[19] Y. Icer and M. Turk, "Implementation of Mainly used Edge Detection Algorithms on FPGA", International Journal of Applied Mathematics, Electronics, and Computers, no.4, 2016.