

Received April 22, 2018, accepted June 19, 2018, date of publication July 9, 2018, date of current version August 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2854306

Fast Pattern Recognition Through an LBP Driven CAM on FPGA

OMER MUJAHID¹, (Student Member, IEEE), ZAHID ULLAH¹, (Member, IEEE), HASSAN MAHMOOD¹, AND ABDUL HAFEEZ²

¹Department of Electrical Engineering, CECOS University of IT & Emerging Sciences, Peshawar 25100, Pakistan

²Department of Computer Science and IT, University of Engineering and Technology, Jalozai Campus, Peshawar 24240, Pakistan

Corresponding author: Zahid Ullah (zahidullah@cecos.edu.pk)

ABSTRACT This paper proposes a novel method for the design of a pattern recognition system based on an integrated approach of local binary patterns (LBP) and content-addressable memory (CAM), which utilizes the logical resources on a field-programmable gate array (FPGA) device. The proposed system uses LBP frequencies instead of pixel data in order to perform exact pattern matching. A logic-based CAM is used to achieve high searching speed. The proposed system is implemented on Xilinx Virtex-7 FPGA and has the ability to recognize patterns regardless of their size and type. The implementation results show that the worst-case lookup time of the proposed system for one complete recognized pattern is merely 1.05 μ s, which is 37.12% lower compared with the state-of-the-art pattern recognition system.

INDEX TERMS Binary CAM, fast pattern recognition, field-programmable gate array (FPGA), local binary patterns, performance improvement, RAM-based CAM.

I. INTRODUCTION

Local binary patterns (LBPs) are visual descriptors that can be used to perform pattern recognition very efficiently [1]–[3]. Developed pattern recognition systems that make use of LBP histograms are deployed using random-access memories (RAMs) [2], [4]. Other approaches are purely software based [5], [6]. The bitwise comparisons in a RAM-based system and a software-based system are accomplished serially and therefore, matching even a single pattern can take significant amount of time [7]. This results in reducing the speed of pattern recognition system, which is based on LBP using RAMs and software based techniques [2], [5].

However, the matching speed of these pattern recognition systems can be improved by deploying them over content-addressable memory (CAM). CAM is a special type of computer memory that gives us data when we provide it with an input address [8], as shown in Fig. 1. CAM performs all search operation in one clock cycle unlike RAM, which uses one clock cycle for each search operation. This enables CAM to perform matching in very less time. CAM has basically two types [9]—binary CAM (BiCAM) and ternary CAM (TCAM). BiCAM stores only binary bits (0 and 1) [10], whereas TCAM stores three states including binary 1, 0, and a don't care state (x). BiCAM performs exact pattern matching while TCAM performs partial pattern matching.

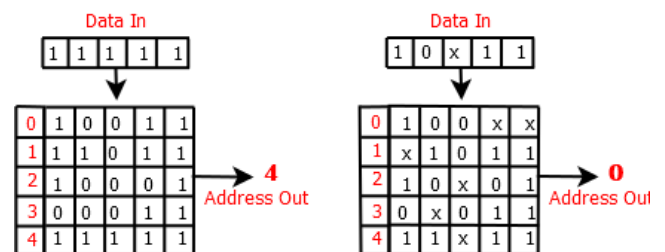


FIGURE 1. Conventional CAM—BiCAM and TCAM.

CAMs are classified as conventional hard CAM and field-programmable gate array (FPGA)-based CAM based on the implementation technology.

A conventional CAM is an expensive computer memory. A cheaper alternative to the conventional CAM is an FPGA-based CAM. An FPGA device does not contain an on-board CAM. We utilize the logical resources of an FPGA device and imitate the functionality of a CAM.

FPGA-based CAM is a dynamic field of research. A number of different techniques are used to implement the functionality of a CAM over an FPGA device. Some of these techniques contain RAM-based CAM [11], [12], lookup tables (LUTs)-based CAM [13], and logical CAMs [14], [15].

A RAM-based CAM uses an FPGA block RAM to imitate a CAM. LUTs-based CAM uses LUTs to imitate a CAM and logic-based CAM uses FPGA logical resources (LUTs and slice registers) to implement the functionality of a CAM.

CAM is capable of high speed search operations but its high speed comes at high cost per bit with low bit density [8]. An FPGA based CAM has several advantages over a conventional CAM .i.e., low cost and high reliability [16]. Since an FPGA device is reconfigurable, this makes an FPGA based CAM more flexible. After careful evaluation of different types of CAM, we chose an FPGA based BiCAM for our system.

The proposed system aims at improving the pattern recognition speed. First, we take advantage of the parallel search capability of CAM. Secondly, we use local binary patterns (LBP) to perform the task of exact pattern matching. By replacing RAM with a CAM, the speed of the recognition process is improved; hence, reducing the processing time. Furthermore, a BiCAM can perform exact pattern matching due to its bit-by-bit matching capability [10]. Such characteristics of BiCAM make it an extremely good choice of memory for a pattern recognition system. The exact pattern matching capability of CAM, when combined with an LBP visual classifier, proposes a potentially brilliant pattern recognition system that can be used in a gamut of real-life applications ranging from biometric recognition to DNA sequence matching to motion detection [17], [18].

Major contributions of the proposed work are recorded below:

- To the best of our knowledge, this is the first ever pattern recognition system that uses an integrated approach of CAM and LBP. Since our proposed system uses CAM instead of RAM, faster recognition speeds are achieved compared to the systems that use RAM.
- The proposed pattern recognition system works equally well for all image types and sizes; a specialty it has got from LBP; hence, making it a general-purpose architecture.
- Moreover, the proposed system uses a CAM of a depth of only 256 for one pattern. Hence, a very small size of computer memory is required to implement the proposed system.

An integrated approach of LBP and CAM results in an extremely fast recognition system with respect to the systems that use RAM- or software-based approaches. In addition, our proposed system is cost-effective and flexible. A comparison with systems using alternate methods for pattern recognition is also done. These results are listed in Table 3. Our proposed architecture goes beyond the advantages of RAM- and software-based methods.

Rest of the paper is organized as follows: Section II discusses local binary patterns. Section III mentions the logic-based CAM. Section IV explains integration of LH-CAM and LBP. Section V has the implementation details and performance evaluation of the proposed system.

Section VI contains conclusions and discussion about future work.

II. LOCAL BINARY PATTERNS (LBPs)

The LBP operator is an image operator, which transforms an image into an array or an image of integer labels describing small-scale appearance of the image [19]. The LBP works on a 3×3 block of pixels as shown in Fig. 2. It assumes the gray scale value of that pixel. The gray scale value is then compared with 8 neighboring values of that pixel. If the gray scale value of the pixel is less than the neighbor pixels value then a binary 1 is obtained; otherwise, a binary value of 0 is assigned. The LBP value of a pixel is an 8 bit binary value. This proves that there must be 256 or less than 256 LBP values in a given texture. The LBP values are then used to construct a histogram, which depicts how frequently a value occurs. The histogram of all the pixels is then concatenated to create the complete LBP histogram of the texture.

In Fig. 3, the value of the center pixel, i.e. at position (1, 1) is compared to the values of all other neighboring pixels. $218 < 157 ?$, $218 < 178 ?$, $218 < 220 ?$, $218 < 219 ?$, $218 < 255 ?$, $218 < 215 ?$, $218 < 219 ?$, $218 < 255 ?$; any one of the above logical statements, yields ‘1’, if true and ‘0’ if false. So the LBP value of pixel (1, 1) is 00111011. This is equal to 59. The LBP of all pixels in the image is similarly obtained. After computing all the LBPs in an image, they are concatenated to generate a histogram. Fig. 4 shows an example of an LBP histogram of a face image. There are 256 possible LBPs in any image i.e. $2^8 = 256$. A histogram of all the available LBP is then created. The LBP value of a pixel is given by (1)

$$LBP(Xc, Yc) = \sum_{n=0}^7 \sigma(ln - lc)2^n \tag{1}$$

Pixel (0,0)	Pixel (0,1)	Pixel (0,2)
Pixel (1,0)	Pixel (1,1)	Pixel (1,2)
Pixel (2,0)	Pixel (2,1)	Pixel (2,2)

FIGURE 2. 3 × 3 pixel block representing pixel position.

157	178	220
219	218	255
215	219	255

FIGURE 3. Gray scale values of the pixels.

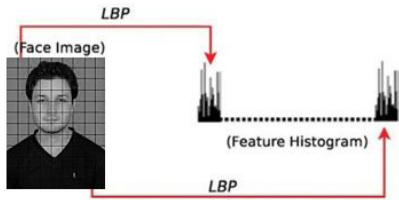


FIGURE 4. Example of an LBP histogram acquired from an image.

In a pixel block of 3×3 size, ln corresponds to central pixel value, while lc corresponds to the 8-neighbor pixel values. The 8 lc values are compared to ln one by one.

$$\sigma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2)$$

Equation (2) shows that a '1' is obtained if the value of center pixel value is lesser than the neighboring pixel value and a '0' is obtained otherwise.

III. LOGIC-BASED CAM

Logic-based CAM is the latest trend in the FPGA based CAM design [13]–[15], [20]. Logic-based CAM has numerous advantages over other FPGA CAM designs [14], [15]. Because of the advantages that a logic-based CAM possess over the established designs, we have chosen a logic-based CAM for the proposed pattern recognition system. The logic-based CAM we chose for our system is called LH-CAM (Logic Based Higher Performance Binary CAM) [15]. Some of the advantages of LH-CAM over other CAM designs are as follows:

The throughput of an LH-CAM is higher because the registers are flexible [15]. Xilinx FPGAs do not have an on-board conventional BiCAM whereas the LH-CAM has been successfully implemented on FPGA [15]. LH-CAM writing is faster if multiple vectors are accessed in parallel provided enough I/Os are available on FPGA device, which is advantageous [15]. FPGA-based logic CAMs are easy to develop, integrate and much cheaper than conventional CAMs [15].

A. LH-CAM ARCHITECTURE

LH-CAM or logic-based high performance binary CAM is a type of logical CAM that can be implemented on an FPGA device [15]. LH-CAM works on the concept of matchlines (ML). Where ML is a combination of storage cells, referred to as logic cells (LC).

Each LC has the capability to store one bit and a dedicated comparison circuitry because of its BiCAM functionality. The comparison circuitry is used to compare the stored bit to the input bit. LCs connected to the same ML are grouped into N-bit vector, where N is the number of bits present in that vector. An LH-CAM uses flip flops (FF) as a storage element to store pattern bits. One FF stores one bit. An LH-CAM ML is a combination of many such FFs. One ML stores one pattern. The number of ML in an LH-CAM shows the number

of patterns it will store. The more the number of ML in an LH-CAM the deeper the LH-CAM will be.

When a match occurs all the LCs must output a logic '1'. When all the LCs output a logic '1' the content associated with that ML is taken to the output by the encoder. If one of the LCs outputs a logic '0', it specifies that a mismatch has occurred. The occurrence of a mismatch causes the whole ML to become a logic '0'. Fig. 5 shows the high level architecture of the modified LH-CAM with each of its LC, ML and encoder visible in the diagram.

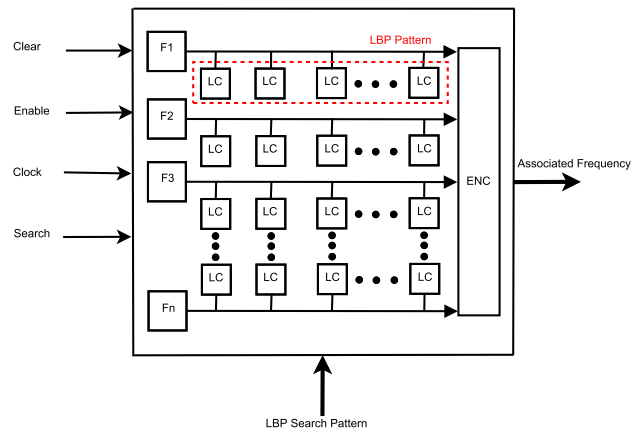


FIGURE 5. LH-CAM architecture with LBP frequencies. (Fn: Frequency of LBP pattern, LC: logic cell, and ENC: encoder).

IV. INTEGRATING LH-CAM AND LOCAL BINARY PATTERNS

The operation of the CAM to output address is subject to the provision of data, however in this case we do not have any address that is associated with the stored pattern. Since the stored pattern is an LBP pattern, the associated content is a frequency of occurrence of that pattern and not an address.

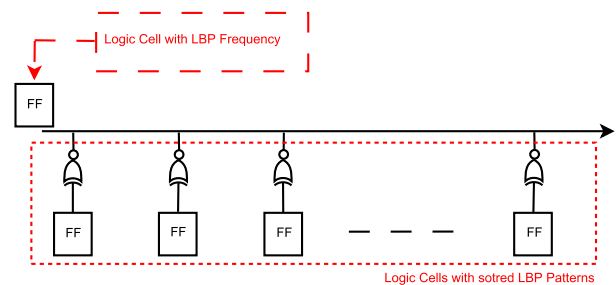


FIGURE 6. Modified LH-CAM matchline with LBP pattern frequency instead of address. (FF: Flip flop).

To enable LH-CAM to perform the pattern recognition task, a slight modification is made to the design. Every ML in the LH-CAM is associated with an LBP frequency (F_n) instead of an address. The LBP frequency content of the ML is stored in a separate logic cell. When a match occurs, the LBP frequency stored in the logic cell associated with the ML is obtained at the output. Fig. 6 shows the modified

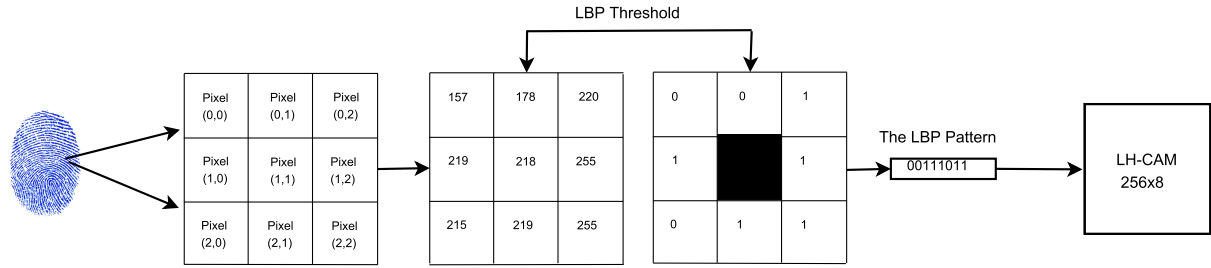


FIGURE 7. Writing pattern to LH-CAM.

LH-CAM matchline. The LBP frequency is stored in a separate logic cell.

A. MAPPING OPERATION

The LBP values of an image are calculated using MATLAB. For this purpose, the built-in MATLAB function ‘ExtractLBPfeatures’ is used. Fig. 7 shows the complete process of LBP extraction and then their mapping to LH-CAM. Algorithm 1 shows the steps that are required to map LBP patterns to LH-CAM vectors. The LBP frequencies of the pattern to be stored are first assigned to the specific LH-CAM vectors. The LBP frequencies of any pattern define its uniqueness.

To store LBP frequencies in their specific ML is pretty straightforward. Each pattern is mapped to its corresponding ML. During mapping, LBP0 is mapped to V0, LBP1 is mapped to V1 and LBP255 is mapped to V255. Each frequency is stored in the logic cell associated with the ML that contains the specific LBP pattern.

Algorithm 1 Mapping Algorithm

- 1: Input : LBP histogram of a pattern
- 2: Output: The LH-CAM words
- 3: **for** $i \leftarrow 0$ to 255 **do**
- 4: Map LBP_i to V_i of LH-CAM
- 5: **end for**

B. LOOKUP OPERATION

The LBP frequencies of the pattern that is to be tested against the stored pattern is obtained by using the same method as discussed in mapping operation. After the LBP patterns are obtained, the first search pattern LBP_search and its associated frequency LBP_frequency is applied to the system. The availability of LBP_search is checked in the memory. If the pattern is present in the memory, its associated frequency(F_n) is taken at the output; otherwise, a NO MATCH label is taken to the output.

If a match is found and an F_n of the pattern is obtained at the output, the F_n is then compared with the LBP_frequency of the input pattern. This task is performed by the comparator block in Fig. 8. If a match occurs in the comparator block, the pattern is deemed as recognized, otherwise a NO MATCH

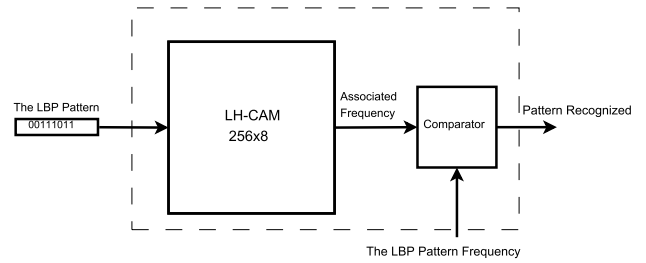


FIGURE 8. The proposed system architecture.

label is obtained at the output. Fig. 8 shows the block diagram of the complete pattern recognition system.

Algorithm 2 Searching Algorithm

- 1: Input : 8-bit LBP search pattern (LBP_search) & Frequency of occurrence of LBP_search (LBP_frequency)
- 2: Output: Pattern is recognized/NO MATCH
- 3: Stored LBP pattern $LBP_i, i=1,2,3\hat{A}e,256$
- 4: **if** LBP_search matches any LBP_i **then**
- 5: Associated LBP frequency F_n is sent to output
- 6: **if** F_n matches LBP_frequency **then**
- 7: Pattern is recognized
- 8: **else**
- 9: Pattern is not recognized
- 10: **end if**
- 11: **else**
- 12: Mismatch occurs and NO MATCH is sent to output
- 13: **end if**

Algorithm 2 shows the steps taken by the system to complete one search operation. There are two inputs to the system. One is an 8 bit LBP pattern (LBP_search) and the other one is the frequency associated with that pattern, referred to as LBP_frequency here, and only one output specifying whether a pattern is recognized or not. The LBP_search is simultaneously compared with all 256 memory locations of the LH-CAM. There is only one possible match. After a successful match occurs, the frequency associated (F_n) with that matchline is obtained. The LBP_frequency is then compared with the F_n . A match in this case completes the recognition process and the pattern is recognized. The time latency of this system is three clock cycles.

V. FPGA IMPLEMENTATION RESULTS AND PERFORMANCE EVALUATION

The proposed system was implemented on Xilinx Virtex-7 FPGA with a speed grade -2. Xilinx ISE Suite 14.5 was used as the software tool to implement the proposed system’s design. Verilog HDL was used as the programming language to develop the proposed system. Table 1 shows the FPGA resources used for one pattern by the proposed system. The slice registers (SRs) used by the proposed system for one pattern are 513, while the LUTs used by the proposed system for one pattern are 899. It is clear from Table 1 that one pattern uses resources as low as 1% of the total available FPGA resources.

TABLE 1. Resource Consumption—FPGA: Xilinx Virtex-7 XC7VX330T—speed grade -2.

Resources	Used	Available	Utilization
SRs	513	408000	1%
LUTs	899	204000	1%

The proposed system was tested on a total of 6 patterns of different types and sizes. The number of LBP patterns varies from pattern to pattern. The number of LBP patterns can never exceed 256 and hence the match time of patterns with all 256 LBPs is always the same. The match time is reduced for patterns with LBPs less than 256, as seen in Fig. 10. It is also observed that the number of bits present in an image has no direct relationship to the number of LBPs present in that image because the number of LBP patterns present in an image depends only on gray scale variations. Hence, the match time for an image does not depend on the number of bits but on the number of LBP patterns.

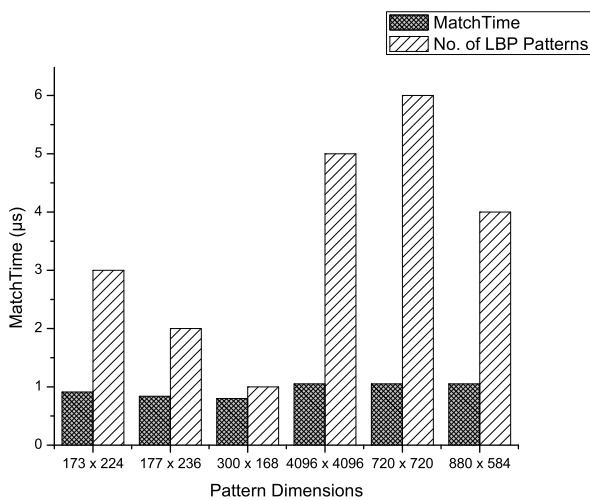


FIGURE 9. Match time vs. pattern dimensions.

Observation of Fig. 9 and Fig. 10 shows that the match time has no relationship with the dimensions of the pattern. Fig. 9 demonstrates two quantities varying with the pattern dimensions. One is the number of LBP patterns and the other

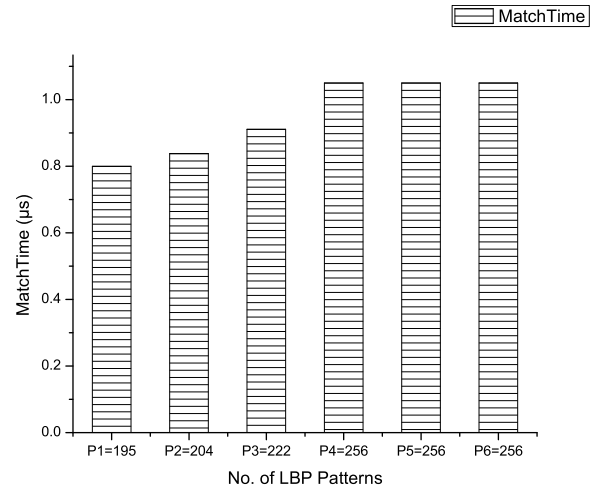


FIGURE 10. Match time vs. # of LBP patterns.

one is the match time. It is clearly seen that the match time does not increase or decrease with pattern dimensions but it is directly proportional to LBP patterns. This means that the dimensions of a pattern does not specify whether match time of a pattern would be more or less. The match time is only specified by the number of LBP patterns. Fig. 10 shows that as the number of LBP patterns increases in an image, the match time also increases with it. The more the LBP patterns in an image, the greater the match time and vice versa as shown in Table 2.

TABLE 2. Test Patterns (RGB images of different sizes and types).

Dimensions	Total No. of Bits	Total No. of LBPs	Match Time (µs)
300 × 168	1209600	195	0.80
177 × 236	1002528	204	0.838
173 × 224	930042	222	0.911
720 × 720	12441600	256	1.05
880 × 584	12334080	256	1.05
4096 × 4096	402653184	256	1.05

A. PERFORMANCE EVALUATION

The choice of LBP in our proposed system has given it a unique quality of being equally efficient for patterns of any size and type. LBP, when compared with other feature extracting image classifier such as Histogram of oriented gradients (HOG) and Scale-invariant feature transform (SIFT), proves advantageous. HOG even though a useful image classifier, has exceeding histogram dimensions. For a pattern of size 64 × 128 the HOG histogram has 3780 dimensions. To accommodate a HOG histogram of 3780 dimensions, we need a CAM of 3780 memory locations deep.

Moreover, the dimensions of HOG are not fixed, with an increasing size in pattern the dimensions of HOG histogram increases. HOG for this reason is not a good choice of feature extractor for our proposed system. SIFT on the other hand, is a

TABLE 3. Performance comparison with the prior work.

References	Ref [5]	Ref [2]	Ref [13]	Ref [21]	Proposed System
Match Time	45ms	5.173ms	1.67 μ s	0.143ms	1.05 μ s

TABLE 4. Performance comparison with the prior work [13].

Resources	Ref[13]	Proposed Work
LUTs	6983	899
Speed (MHz)	238.83	243.5

local feature descriptor. Since the mechanism of our proposed system is suited only for global feature descriptor, it would not be a wise choice to use a local feature descriptor in it. LBP is a global feature descriptor. A comparison is made with certain pattern recognition systems based on LBP using different approaches. In Table 3, [5] uses a software based approach for LBP using Euclidean distance method to find the similarity between two faces. Reference [2] is also an LBP based system. It is implemented on an FPGA device and uses RAM for comparisons. Reference [13] makes use of FPGA CAM but does not use LBP.

Reference [21] is an image recognition system that is implemented on FPGA based CAM. It does not make use of LBP but instead uses a combination of logic circuits, i.e. shift register, AND gate and a Finite State Automation (FSA) controller that monitors the whole system. Reference [21] is designed for an image size of 32 \times 32 only, whereas our proposed system works equally fine for an image of any dimension. This is the reason that the size of CAM and the logical resources used by [21] increases as the dimensions of image increases. Moreover, our proposed system performs better in terms of matching speeds as well. It could be observed in Table 3 that our proposed system by combining LBP and LH-CAM outperforms all other pattern recognition techniques in terms of speed.

A thorough comparison in terms of speed and resources usage is made with [13], a state of the art pattern recognition system. Reference [13] is chosen for comparison with the proposed system because it achieves one of the faster pattern matching speeds among the existing pattern recognition systems. Reference [13] and our proposed system both work on binary data. Moreover, the proposed system and [13] are both implemented over the same FPGA device with the same speed grade.

Our proposed system has a match time of 1.05 μ s, which is 37.12% less than [13]. Our proposed system is not only faster but also uses minimum logical resources as compared to [13] as shown in Table 4. The proposed system has a speed of 243.5 MHz, which is 1.96% faster than [13].

Different performance parameters for the proposed system are: Speed of the proposed system is 243.5 MHz, latency is 3 clock cycles, and the power consumed by the proposed

system at 100 MHz clock speed is 6 mW. Key benefits of the proposed system are summarized as below:

- The proposed system requires very little preprocessing as LBP operator can be applied to any type of image.
- The proposed system works just the same for any type and size of two-dimensional image.
- The proposed system requires memory of only 256 locations deep.

VI. CONCLUSIONS AND FUTURE WORK

A fast pattern recognition system is proposed that uses an integrated approach of CAM and LBP. The proposed system uses LBP frequencies instead of redundant pixel data and is deployed over an FPGA logic-based CAM. The proposed system outperforms other pattern recognition systems in terms of speed. Our system is capable of working equally on patterns of any size and type, which makes it flexible and versatile. Furthermore, the proposed system requires 256 CAM locations for one stored pattern, which makes it a general purpose pattern recognition system.

Future work includes extension and modification of this architecture for approximate pattern matching.

REFERENCES

- [1] S. Liao and A. C. S. Chung, "Texture classification by using advanced local binary patterns and spatial distribution of dominant patterns," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 1, Apr. 2007, pp. 1-1221-1-1224.
- [2] N. Stekas and D. van den Heuvel, "Face recognition using local binary patterns histograms (LBPH) on an FPGA-based system on chip (SoC)," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2016, pp. 300-304.
- [3] Y. Ding, Q. Zhao, B. Li, and X. Yuan, "Facial expression recognition from image sequence based on LBP and Taylor expansion," *IEEE Access*, vol. 5, pp. 19409-19419, 2017.
- [4] Y. Zhang, W. Cao, and L. Wang, "Implementation of high performance hardware architecture of face recognition algorithm based on local binary pattern on FPGA," in *Proc. IEEE 11th Int. Conf. ASIC (ASICON)*, Nov. 2015, pp. 1-4.
- [5] G. Xiang, Z. Qiuyu, W. Hui, and C. Yan, "Face recognition based on LBPH and regression of local binary features," in *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)*, Jul. 2016, pp. 414-417.
- [6] I. R. P. Selvam and M. Karupiah, "Gender recognition based on face image using reinforced local binary patterns," *IET Comput. Vis.*, vol. 11, no. 6, pp. 415-425, Sep. 2017.
- [7] N. I. Rafla and I. Gauba, "A reconfigurable pattern matching hardware implementation using on-chip RAM-based FSM," in *Proc. 53rd IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2010, pp. 49-52.
- [8] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, Mar. 2006.
- [9] R. Karam, R. Puri, S. Ghosh, and S. Bhunia, "Emerging trends in design and applications of memory-based computing and content-addressable memories," *Proc. IEEE*, vol. 103, no. 8, pp. 1311-1330, Aug. 2015.
- [10] A. X. Liu, C. R. Meiners, and E. Torng, "Packet classification using binary content addressable memory," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1295-1307, Jun. 2016.
- [11] Z. Ullah, M. K. Jaiswal, R. C. C. Cheung, and H. K. H. So, "UE-TCAM: An ultra efficient SRAM-based TCAM," in *Proc. TENCON IEEE Region 10 Conf.*, Nov. 2015, pp. 1-6.
- [12] D.-H. Le, M. Sowa, C.-K. Pham, and K. Inoue, "A fully-parallel information detection hardware system employing content addressable memory," in *Proc. 4th Int. Conf. Commun. Electron. (ICCE)*, Aug. 2012, pp. 447-452.

- [13] T. Harbaum, M. Seboui, M. Balzer, J. Becker, and M. Weber, "A content adapted FPGA memory architecture with pattern recognition capability for L1 track triggering in the LHC environment," in *Proc. IEEE 24th Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM)*, May 2016, pp. 184–191.
- [14] M. Irfan and Z. Ullah, "G-AETCAM: Gate-based area-efficient ternary content-addressable memory on FPGA," *IEEE Access*, vol. 5, pp. 20785–20790, 2017.
- [15] Z. Ullah, "LH-CAM: Logic-based higher performance binary CAM architecture on FPGA," *IEEE Embedded Syst. Lett.*, vol. 9, no. 2, pp. 29–32, Jun. 2017.
- [16] Z. Ullah, M. K. Jaiswal, Y. C. Chan, and R. C. C. Cheung, "FPGA implementation of SRAM-based ternary content addressable memory," in *Proc. IEEE 26th Int. Parallel Distrib. Process. Workshops PhD Forum*, May 2012, pp. 383–389.
- [17] E. I. Junior, L. M. Garces, T. C. Pimenta, and A. J. Cabrera, "FPGA-based EMD assist block for motion detection in critical environments," *IEEE Latin Amer. Trans.*, vol. 15, no. 10, pp. 1856–1863, Oct. 2017.
- [18] S. Jin, D. Kim, T. T. Nguyen, D. Kim, M. Kim, and J. W. Jeon, "Design and implementation of a pipelined datapath for high-speed face detection using FPGA," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 158–167, Feb. 2012.
- [19] R. Li, X. Li, and T. Kurita, "Soft local binary patterns," in *Proc. 7th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Nov. 2015, pp. 70–75.
- [20] A. Annovi et al., "A XOR-based associative memory block in 28 nm CMOS for interdisciplinary applications," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2015, pp. 392–395.
- [21] D.-H. Le, T.-B.-T. Cao, K. Inoue, and C.-K. Pham, "A CAM-based information detection hardware system for fast image matching on FPGA," *IEICE Trans. Electron.*, vol. E97-C, no. 1, pp. 65–76, Jan. 2014.



OMER MUJAHID received the B.Sc. degree in electrical engineering from CECOS University of IT & Emerging Sciences in 2013, where he is currently pursuing the M.S. degree.

He is currently with the Signals Processing Lab, CECOS University of IT & Emerging Sciences. His research interest includes pattern recognition, embedded systems, and FPGA-based systems.



ZAHID ULLAH received the B.Sc. degree (Hons.) in computer system engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2006, the M.S. degree in electronic, electrical, control, and instrumentation engineering from Hanyang University, South Korea, in 2010, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2014.

He is currently serving as a Chairman with the Department of Electrical Engineering, CECOS University of IT & Emerging Sciences, Peshawar, Pakistan. He holds prestigious journal and conference papers, and patents in his name in the field of FPGA-based CAM. His research includes low power/high speed CAM design on FPGA, low power/high speed VLSI design, pattern matching, and embedded systems.



HASSAN MAHMOOD received the B.Sc. degree (Hons.) in computer system engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2010, and the M.S. degree in electrical communication engineering from the CECOS University of IT & Emerging Sciences, Peshawar.

He is currently a Manager Technical Support in Canadian-based firm PLC GROUP. He has over seven years of embedded systems, end-to-end product development, and research experience. His research includes m2m communication, embedded system, and FPGA-based CAM design.



ABDUL HAFEEZ received the B.Sc. degree in computer systems engineering from the University of Engineering and Technology (UET), Peshawar, Pakistan, in 2006, and the Ph.D. degree from Virginia Tech with a focus in high-performance computing and machine learning in 2014. Furthermore, he has been a Post-Doctoral Fellow with Georgia Tech, where he involved in collaborative platforms for materials scientists and data scientists.

He is currently an Assistant Professor with the Department of Computer Science, UET Jalozai Campus. He has published his work in prestigious journal and conference papers. His research focuses on parallel processing, machine learning, and data science and its applications in different domains, especially, biomedical computing. He has collaborated with the University of Texas at Arlington, USA and IBM Research Almaden and IBM Research at Dublin, Ireland.

...