

Research Article

An Efficient Reconfigurable Architecture for Fingerprint Recognition

Satish S. Bhairannawar,¹ K. B. Raja,² and K. R. Venugopal³

¹Department of Electronics and Communication, SDMCET, Dharwad 580002, India

²Department of Electronics and Communication, UVCE, Bangalore 560001, India

³UVCE, Bangalore University, Bangalore, India

Correspondence should be addressed to Satish S. Bhairannawar; satishbhairannawar@gmail.com

Received 7 December 2015; Revised 26 April 2016; Accepted 9 May 2016

Academic Editor: Lazhar Khriji

Copyright © 2016 Satish S. Bhairannawar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The fingerprint identification is an efficient biometric technique to authenticate human beings in real-time Big Data Analytics. In this paper, we propose an efficient Finite State Machine (FSM) based reconfigurable architecture for fingerprint recognition. The fingerprint image is resized, and Compound Linear Binary Pattern (CLBP) is applied on fingerprint, followed by histogram to obtain histogram CLBP features. Discrete Wavelet Transform (DWT) Level 2 features are obtained by the same methodology. The novel matching score of CLBP is computed using histogram CLBP features of test image and fingerprint images in the database. Similarly, the DWT matching score is computed using DWT features of test image and fingerprint images in the database. Further, the matching scores of CLBP and DWT are fused with arithmetic equation using improvement factor. The performance parameters such as TSR (Total Success Rate), FAR (False Acceptance Rate), and FRR (False Rejection Rate) are computed using fusion scores with correlation matching technique for FVC2004 DB3 Database. The proposed fusion based VLSI architecture is synthesized on Virtex xc5vlx30T-3 FPGA board using Finite State Machine resulting in optimized parameters.

1. Introduction

The reliable personnel authentication [1, 2] based on biometrics has significant importance in the present digital world and can be achieved by human and computer interface activities. The evolution of biometrics in recent years from single mode to multiple mode closed systems has made it possible to consider for Big Data processing [3, 4]. The development of new algorithms and parallel processing architectures has an impact for Big Data processing response time. The interoperable feature and many sources and avenues for collection of biometric samples have made the biometric evolution of Big Data possible. The biometric physiological or behavioural samples are captured using sensors or devices, which are further processed in the next level of vetting through Office for Personal Management (OPM) which can be either verification or identification. Fingerprint based identification is one of the most important biometric

technologies, which have drawn a substantial amount of attention recently since the process of acquiring fingerprint samples are easy and simple. A fingerprint is seen as a set of interleaved ridges and valleys on the surface of the finger. The most fingerprint matching approach relies on the fact that the uniqueness of a fingerprint can be determined by minutiae, which are represented by either bifurcation or termination of ridges. The quality and enhanced minutiae [5–7], which influence recognition rates are discussed in literature.

The features of a fingerprint can be derived using the following: (i) *Spatial domain*: the features of an image are carried out directly on pixel value. Examples are Local Binary Pattern [8], Complete Linear Binary Pattern [9], and Singular Value Decomposition [10]. (ii) *Transform domain*: in this any transform is applied to an original image to get a transformed image on which further processing is done. Examples are Fast Fourier Transform [11], Discrete Cosine Transform [12], Discrete Wavelet Transform [13], and Dual Tree Complex

Wavelet Transform [14]. (iii) *Fusion*: in this technique [15, 16] it combines the advantages of both spatial and transform domain.

The automated fingerprint recognition system is used for both identification and verification against standard database law enforcement agencies to identify the suspect for committing crime or for attendance verification process to verify the claimed identity. The performance speed of fingerprint system is a critical factor to be addressed while dealing with large databases. The real-time processing of a fingerprint recognition system is its ability to process the large data and produce the results within certain time constraints in the order of milliseconds and sometimes microseconds depending on the application and the user requirements. In this category, Field Programmable Gate Array (FPGA) outperforms other processors. The FPGAs are specially built hardware optimized for speed and are suitable for real-time biometric data processing. Multicores and HPC clusters have reasonable real-time processing capabilities, but not efficient as FPGA with many processing cores and high bandwidth memory.

In real time, speed of the algorithm becomes crucial which in turn defines the throughput. The efficient FPGA architectures [17–20] for fingerprint processing and existing algorithms to identify a fingerprint based on minutiae [21], ridge, multiresolution features, and Hough transform were discussed.

Vatsa et al. [22] proposed Redundant Discrete Wavelet Transform based on local image quality assessment algorithm followed by extraction algorithm using Level 3 features. These features are combined with Level 1 and Level 2 in the fingerprint identification scheme. Finally, the matching performance was improved by using quality based likelihood ratios. Govan and Buggy [23] proposed effective matching solution that addresses security and privacy issues. This technique eliminates the requirement to release biometric template data into an open environment which uses embedded applications such as smart cards. The effective disturbance rejection methodology which is able to differentiate between equivalent and insignificant structure models was discussed.

Nain et al. [24] proposed an algorithm to classify fingerprint images into four different classes using High Ridge Curvature (HRC) algorithm involving two stages. In the first stage, HRC region was extracted, which avoids core point detection. In the second stage, ridges inside HRC region were considered for matching. The global distribution structure and the local matching similarities [25] between fingerprints were considered for matching using Hidden Markov Model (HMM) [26]. Nikam and Agarwal [27] proposed spoof fingerprint detection using ridgelet transform. The comparisons of individual ridgelet energy and cooccurrence signatures were analysed and also testing was done using diverse classifiers. Masmoudi et al. [28] proposed an algorithm which used the rotation invariant measured as local phase and was combined with Linear Binary Pattern Features to improve the performance accuracy. Stewart et al. [29] proposed the test technique to determine the effects of outdoor and cold weather effects on chip versus optical fingerprint scanner, fingerprint recognition quality, and device interaction. The

results suggested that performance has no dependence on temperature and humidity. Cao and Dai [30] proposed finger-print segmentation for online process using frame difference technique. Further the segmented foreground was used for identification.

UmaMaheswari et al. [31] proposed finger-print classification and recognition using neuro-nearest neighbour based method which improves classification rate. This consists of different stages such as image enhancement, line detector base feature extraction, and neural network classification using back propagation networks. The results have shown the accurate estimation of orientation and ridge frequency which helps in better recognition. Conti et al. [32] proposed pseudo-singularity points based finger-print recognition. This technique uses additional parameters such as their relative distance and orientation around standard singularity points (core and delta) which enhances the matching performance. Ahmed et al. [33] proposed Compound Local Binary Pattern (CLBP) for rotation invariant texture classification. This combines magnitude information of the difference between two grey values with original LBP pattern and provides robustness. Paulino et al. [34] proposed an alignment algorithm (descriptor-based Hough transform) for latent finger-print matching. This technique measures similarity between finger-prints by considering both minutiae and orientation field information. The comparison was done between proposed and generalized Hough transform for large database.

Feng et al. [35] proposed a technique using orientation field estimation based on prior knowledge of finger-print structure. The dictionary of reference for orientation patches was constructed using a true set of orientation fields. The approach was applied to the overlapped latent finger-print database to achieve better performance compared to conventional algorithms.

Contribution. The contribution and novel aspects of the proposed techniques are listed as follows: (i) the computation of the novel matching score for CLBP and DWT features; (ii) the matching score values which are varied based on characteristics of images, that is, the values which are computed adaptively based on characteristics of the images; (iii) the fusion of matching scores with improvement factor; (iv) the implementation of FSM based VLSI architecture to improve the hardware performance.

2. Proposed Fingerprint Recognition System

An efficient finger-print recognition model using histogram of CLBP scores, DWT feature scores, and fusion of both scores is given in Figure 1.

2.1. Fingerprint Database. The DB3 of FVC2004 finger-print database [36] is considered for performance analysis. The size of each finger-print image is 300×480 with 512 dpi. The finger-print samples of ten different persons are shown in Figure 2.

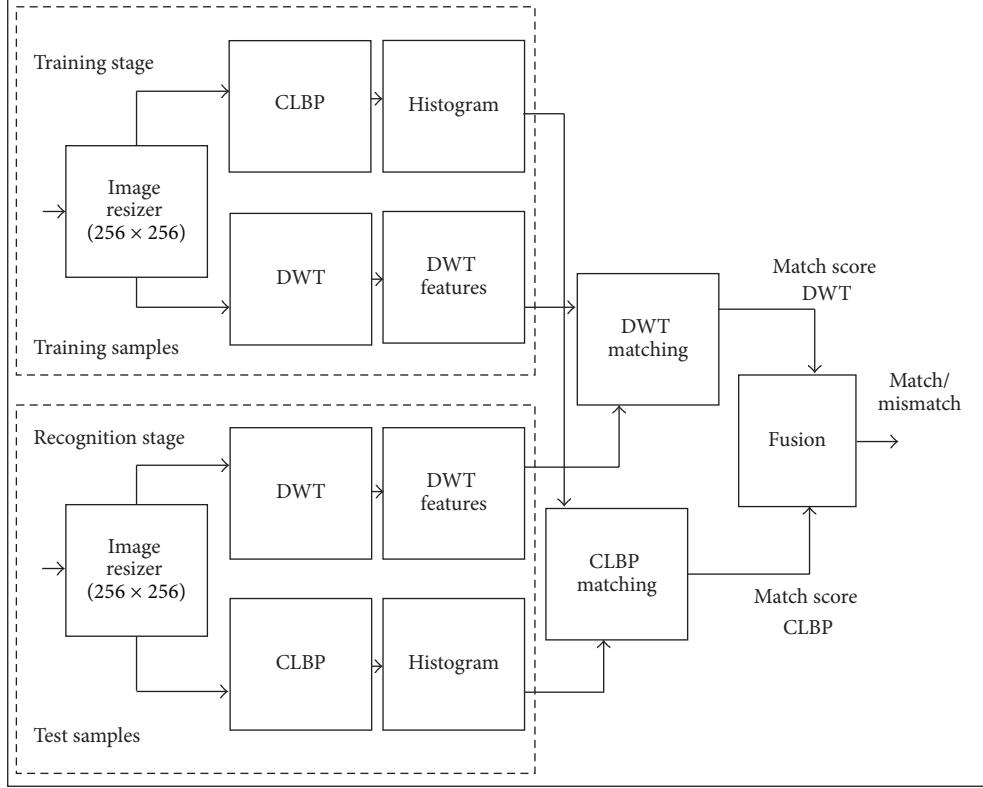


FIGURE 1: Block diagram of the proposed fingerprint recognition system.

2.2. Preprocessing. The original fingerprint image of size 300 × 480 is resized to 256 × 256, which is suitable for hardware implementation.

2.3. Complete Local Binary Pattern (CLBP). It is an extension of the Local Binary Pattern (LBP) [37] texture operator. The CLBP operator gives both sign CLBP_S and magnitude components CLBP_M for each pixel from its neighbouring pixels. If P is the number of neighbours of a centre pixel, then CLBP operator uses $2P$ bits to code centre pixel. The first P MSB bits represent sign and the next P LSB bits represent magnitude.

The binary bit patterns are generated for sign and magnitude components for each pixel. The fingerprint image is scanned from left to right and top to bottom and considering each pixel which is surrounded by 8 neighbouring pixels, that is, 3×3 matrix. The centre pixel intensity value is I_c and surrounded neighbouring pixel intensity values, say, I_p . The sign bit patterns for 3×3 matrices are generated using

$$S(I) = \begin{cases} 0, & I_p - I_c \leq 0 \\ 1, & I_p - I_c > 0. \end{cases} \quad (1)$$

The magnitude bit pattern is generated using

$$M(I) = \begin{cases} 0, & I_p - I_c \leq M_{\text{avg}} \\ 1, & I_p - I_c > M_{\text{avg}}, \end{cases} \quad (2)$$

where $M_{\text{avg}} = |m_1| + |m_2| \dots |m_8|/8$ and m_1 to m_8 are the magnitude values of the difference between respective I_p and I_c .

Each neighbourhood pixel is represented by two bits; that is, MSB bit represents sign and the LSB bit represents magnitude. Each centre pixel is represented by eight sign bits and eight magnitude bits. The example for CLBP is as shown in Figure 3. The arbitrary values for 3×3 matrix are considered in Figure 3(a). The values of neighbouring pixels are subtracted from centre pixel value and are given in Figure 3(b). The sign of each coefficient in Figure 3(b) is represented in Figure 3(c) as sign component of CLBP. The magnitude components of CLBP are shown in Figure 3(d) by considering only magnitude of Figure 3(b). The average value of the CLBP magnitude component is computed and is compared with neighbouring CLBP magnitude coefficient values and assigns binary values using (2) to generate CLBP magnitude pattern given in Figure 3(f). The numbers of centre pixels available for image size 256×256 are 64516 using 3×3 window matrix. The binary eight bits of sign and magnitude of each pixel are converted into decimal values for feature extraction. If the CLBP sign and magnitude coefficient features are considered directly for an image size of 256×256 , the algorithm requires 64516 for sign and 64516 for magnitude; that is, total number of features are 129032.

2.3.1. Histogram of CLBP Features. The features obtained directly from CLBP are large in number and hence increase in

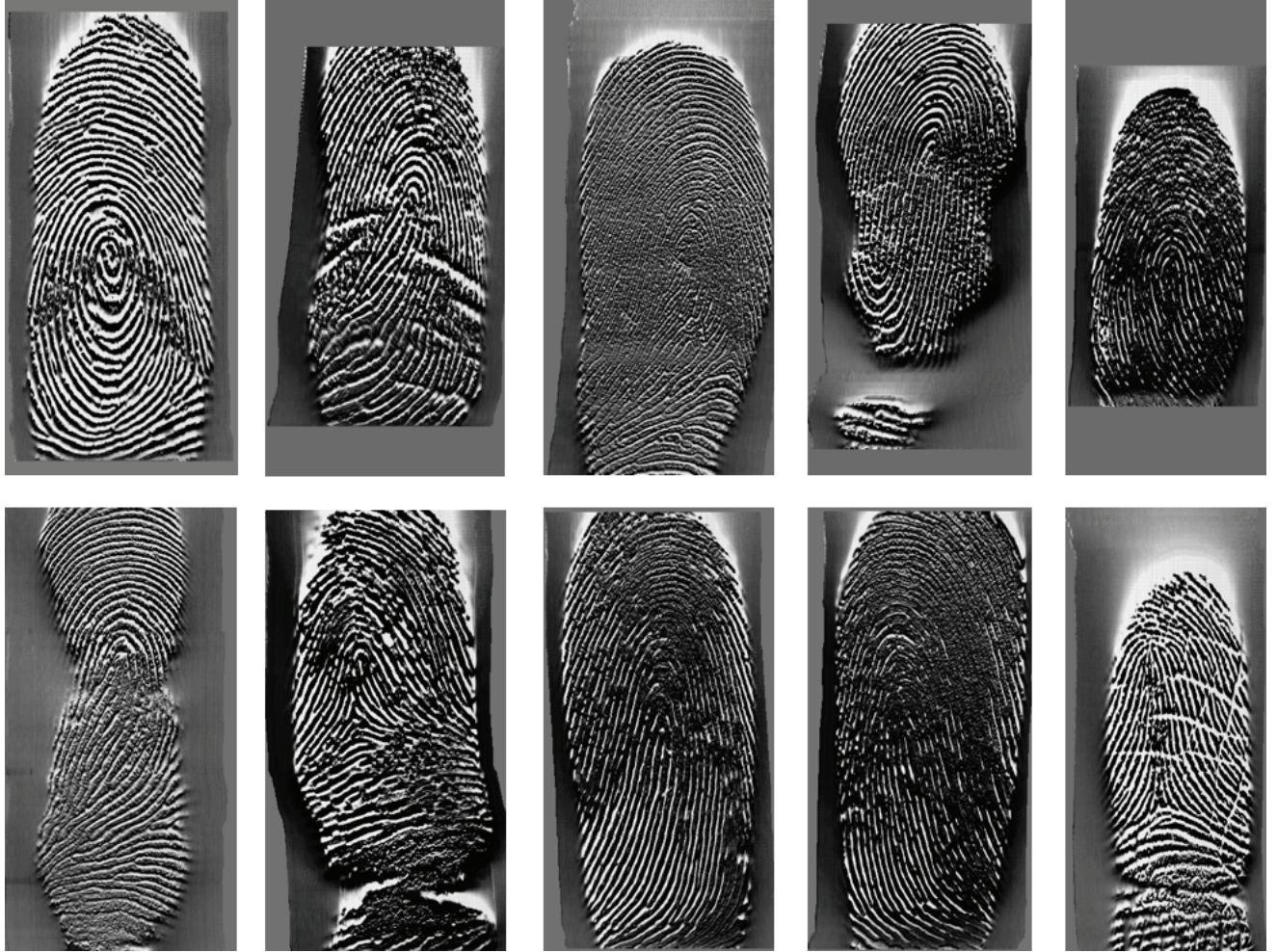


FIGURE 2: Sample images of ten different persons of FVC2004 database.

matching processing time and are a disadvantage in hardware implementation. The histogram on CLBP produces only 256 features for each sign and magnitude. Hence the number of features is reduced from 129032 to 512, that is, approximately 0.4% features compared to CLBP. The advantage of histogram on CLBP is that the number of features reduces and also features are more unique. The histograms of original fingerprint, sign, and magnitude components of CLBP are shown in Figure 4.

2.3.2. Proposed CLBP Matching Score. The CLBP histograms of test and database images are compared componentwise to compute CLBP match score C . The absolute sign component difference $CLBP_S_D_i$ between sign component $CLBP_S_T_i$ of test fingerprint and sign component $CLBP_S_D_{ji}$ of fingerprint images in the database is computed using

$$CLBP_S_D_i = |CLBP_S_T_i - CLBP_S_D_{ji}|, \quad (3)$$

where i is intensity values (0 to 255); j = number of persons in the database \times number of images per person.

The CLBP sign histogram coefficients match x_i is computed based on threshold sign difference value (14 for best match) given in

$$x_i = \begin{cases} 1, & CLBP_S_D_i < 14 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The absolute magnitude component difference $CLBP_M_D_i$ between magnitude component $CLBP_M_T_i$ of test fingerprint and magnitude component $CLBP_M_D_{ji}$ of fingerprint images in the database is computed using

$$CLBP_M_D_i = |CLBP_M_T_i - CLBP_M_D_{ji}|, \quad (5)$$

where i is intensity values (0 to 255); j = number of persons in the database \times number of images per person.

The CLBP magnitude histogram coefficient match y_i is computed based on threshold magnitude difference value (18 for best match) given in

$$y_i = \begin{cases} 1, & CLBP_M_D_i < 18 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

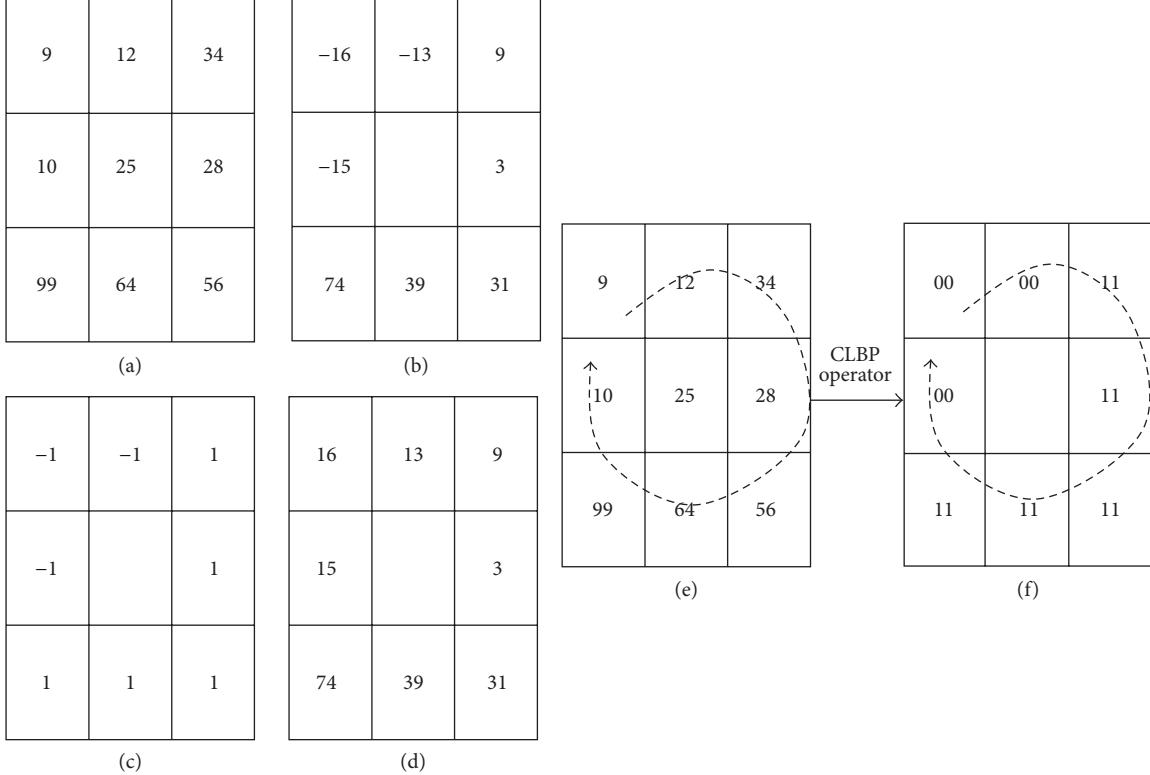


FIGURE 3: CLBP operator: (a) 3×3 sample block; (b) local difference; (c) sign components; (d) magnitude components; (e) original matrix; (f) CLBP matrix.

The overall CLBP match count by considering both sign and magnitude histogram coefficients is

$$\text{CLBP_Match_count} = \begin{cases} \text{Match count} + 1, & \text{if } x_i \cdot y_i = 1 \\ \text{Match count}, & \text{otherwise.} \end{cases} \quad (7)$$

The CLBP match score is computed using CLBP match count and number of histogram levels using

$$\% \text{ CLBP_Match_score } C = \frac{\text{CLBP_Match_count} * 100}{\text{Number of Histogram Levels}}. \quad (8)$$

The *first* and *eighth* samples of same person are considered as database and test image. The original fingerprint, CLBP magnitude component, and CLBP sign component images of database and test image are shown in Figures 5(a)–5(c) and 5(d)–5(f) respectively. The CLBP_Match_score is computed between database image and test image of the same person, which yields high value, that is, 67.9%.

The first and eighth samples of different person are considered as database and test image. The original fingerprint, CLBP magnitude component and CLBP sign component images of the database and test image are shown in Figures 6(a)–6(c) and 6(d)–6(f) respectively. The CLBP_Match_score is computed between the database image and test image of the different person, which yields low value, that is, 51.9531%.

2.4. DWT Algorithm. The DWT [38] provides spatial and frequency characteristics of an image. It has an advantage over Fourier transform in terms of temporal resolution where it captures both frequency and location information. The signal is translated into shifted and scaled versions of the mother wavelet to generate DWT bands. The fingerprint image is decomposed into multiresolution representation using DWT. The LL subband gives overall information of the original fingerprint image, the LH subband represents horizontal information of the fingerprint image, HL gives vertical characteristics of the fingerprint image, and HH gives diagonal details.

The Haar wavelets are orthogonal and have simplest useful energy compression process. The Haar transformation on one-dimension inputs leads to a 2-element vector using

$$(y(1), y(2)) = T(x(1), x(2)), \quad (9)$$

where $T = (1/\sqrt{2}) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is the Haar operator and $y(1)$ and $y(2)$ are the sum and difference of $x(1)$ and $x(2)$ which produce low pass and high pass filtering, respectively, scaled by $1/\sqrt{2}$ to preserve energy. The Haar operator T is an orthonormal matrix since its rows are orthogonal to each other (their dot products are zero) and have unit lengths; therefore $T^{-1} = T^T$. Hence we may recover x from y using

$$(x(1), x(2)) = T^T (\gamma(1), \gamma(2)). \quad (10)$$

For 2D image, Let x be 2×2 matrix of an image; the transformation y is obtained by multiplying columns of x by

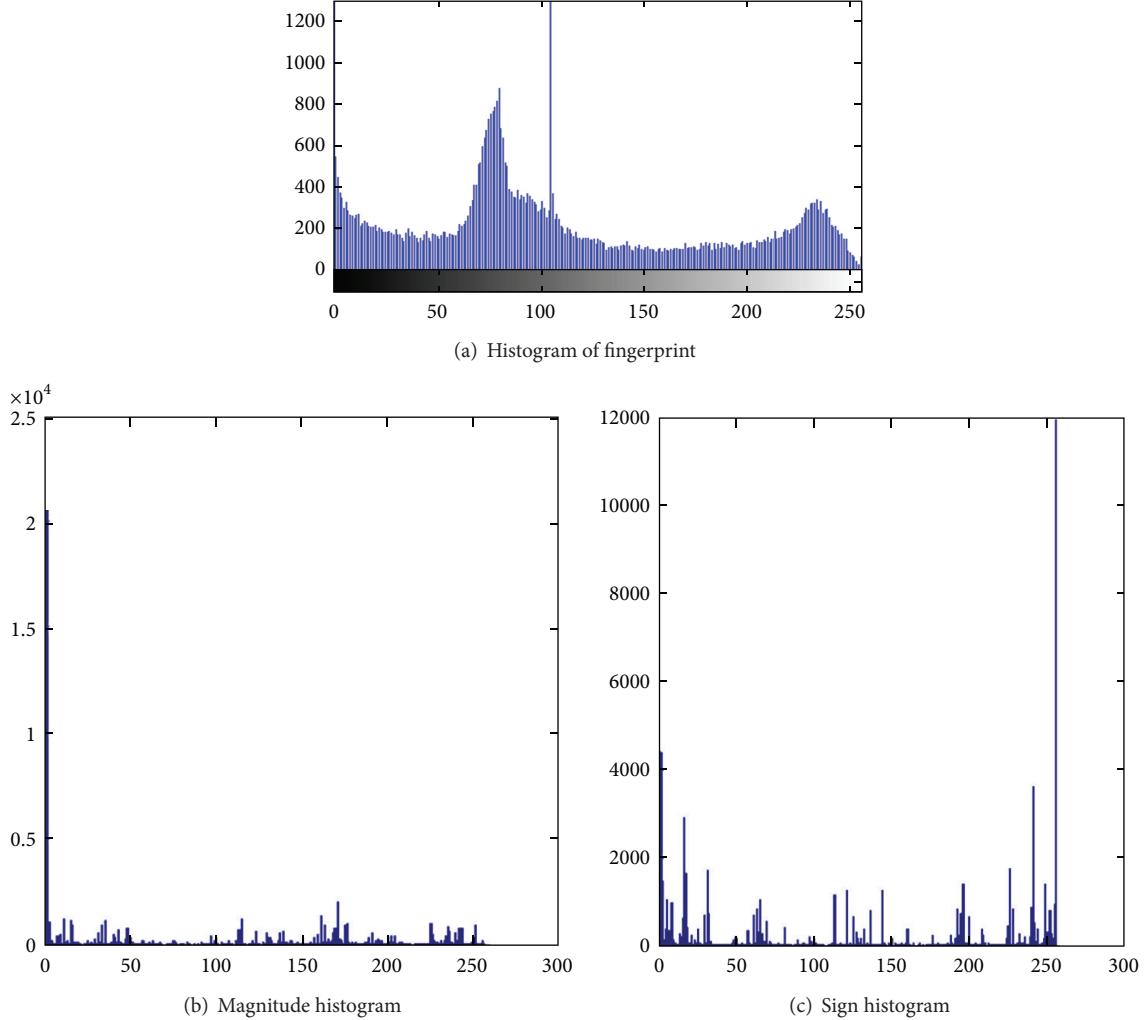


FIGURE 4: Histograms of original fingerprint and CLBP features.

T , and then the rows of the result are multiplied by T^T using

$$y = T * x * T^T. \quad (11)$$

The original values are recovered using

$$x = T^T * y * T. \quad (12)$$

An example of DWT is as follows.

If $x = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is the original matrix, then DWT is given in (13).

Then

$$y = \frac{1}{2} \begin{pmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{pmatrix}. \quad (13)$$

The level 2 DWT features can be obtained by applying Haar wavelet on LL subband of Level 1. The decomposition of fingerprint using DWT at two levels is shown in Figure 7.

The DWT bands correspond to the following filtering processes:

- (i) LL ($a + b + c + d$): low pass filtering in horizontal as well as vertical direction.
- (ii) HL ($a - b + c - d$): high pass filtering in horizontal direction and low pass filtering in vertical direction.
- (iii) LH ($a + b - c - d$): low pass filtering in horizontal and high pass filtering in vertical direction.
- (iv) HH ($a - b - c + d$): high pass filtering in both horizontal and vertical direction.

To use this transform to a complete image, the pixels are grouped into 2×2 blocks and transformations are obtained using (13) for each block. The 2-level DWT is applied to the fingerprint image of size 256×256 to obtain 128×128 coefficients after first level and 64×64 coefficients after second-level stage. The 64×64 LL subband coefficients are considered as DWT features.

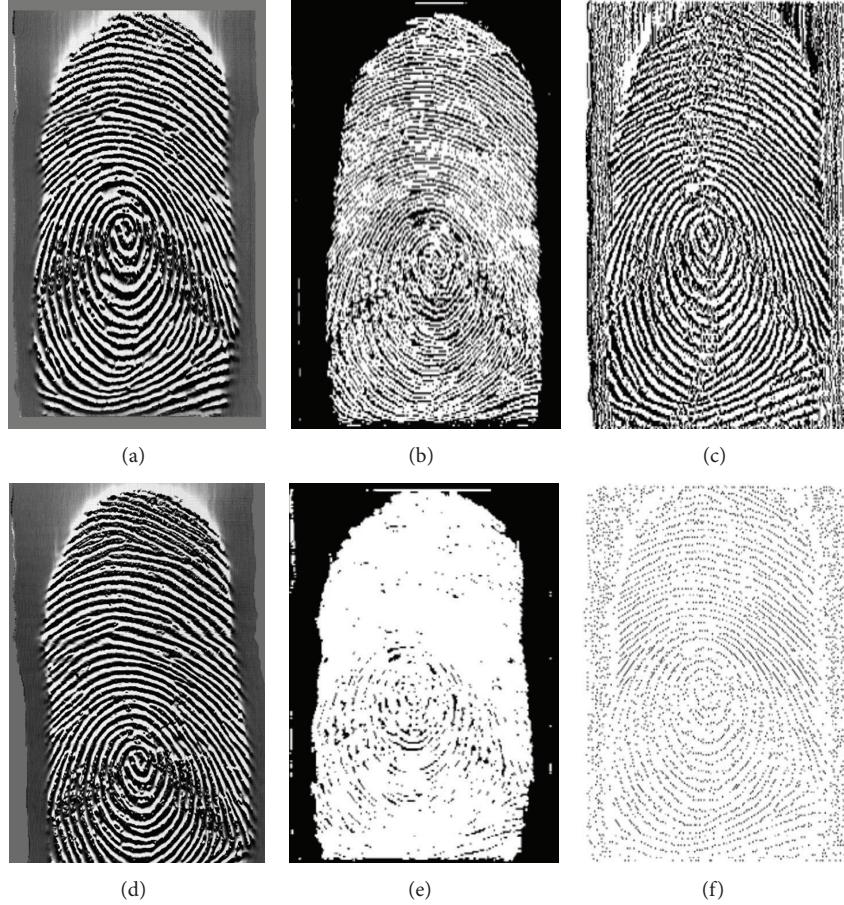


FIGURE 5: CLBP images of same person with matching score 67.9%.

2.4.1. Proposed DWT Matching Score. The LL subband coefficients of Level 2 DWT of the test fingerprint are compared with LL band coefficients of fingerprint images present in the database using difference formula between coefficients using

$$\text{DWT_}D_k = | \text{DWT_}T_k - \text{DWT_}D_{jk} |, \quad (14)$$

where K is the number of second-level subband coefficients, that is, 4096 for original image size of 256×256 .

The DWT coefficient match Z_K is given by

$$Z_k = \begin{cases} 1, & \text{DWT_}D_k < 45 \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

The DWT_Match_count by considering Level 2 coefficients is given in

$$\begin{aligned} \text{DWT_Match_count} \\ = \begin{cases} \text{Match count} + 1, & \text{if } Z_k = 1 \\ \text{Match count}, & \text{otherwise.} \end{cases} \end{aligned} \quad (16)$$

The DWT_Match_score is computed using DWT match count and total number of DWT coefficients using

$$\begin{aligned} \% \text{ DWT_Match_score } D \\ = \frac{\text{DWT_Match_count} * 100}{\text{Total No. of Level 2 DWT coefficients}}. \end{aligned} \quad (17)$$

The *first* and *eighth* samples of same person are considered as database image and the test image. The corresponding LL subband images of DWT database image and test image are shown in Figures 8(a)-8(b) and 8(c)-8(d) respectively. The matching score of 21.6768% is high, since the score is computed between two samples of the same person.

The *first* and *eighth* samples of different person are considered as database image and the test image. The corresponding LL subband images of DWT database image and test image are shown in Figures 9(a)-9(b) and 9(c)-9(d) respectively. The matching score of 14.2129% is low, since the score is computed between two samples of different person.

2.5. Fusion. The percentage CLBP match score is fused with percentage DWT matching score [39] to improve performance of the proposed algorithm using

$$\text{Final Score } F = P * D + (1 - P) * C, \quad (18)$$

where P is an improvement factor which varies from 0 to 1.

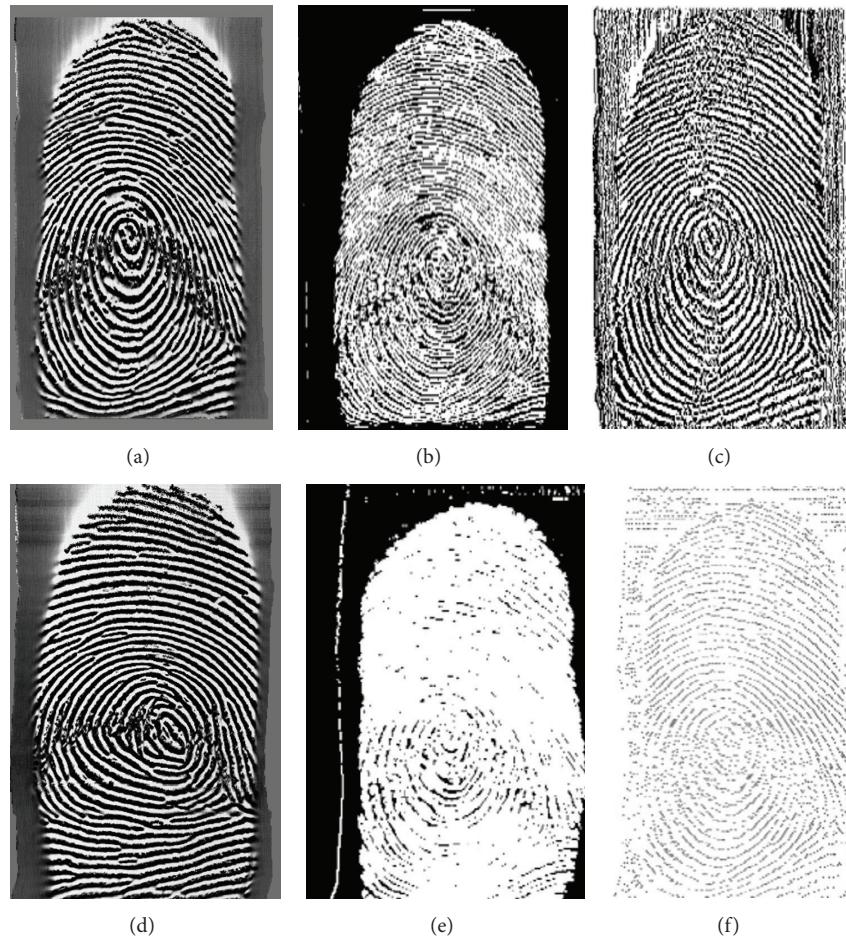


FIGURE 6: CLBP images of different persons with matching score 51.9531%.

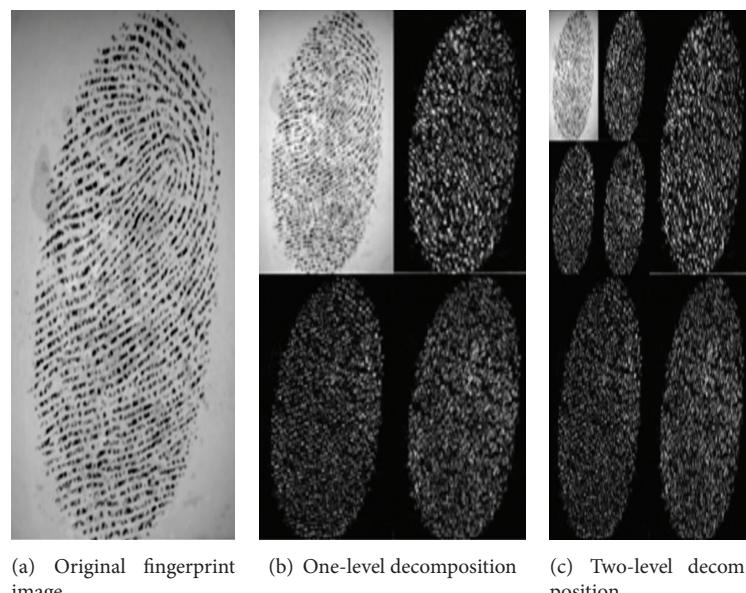


FIGURE 7: DWT decomposition.



FIGURE 8: DWT images of same person with matching score 21.6768%.

3. Algorithm

The proposed efficient algorithm is shown as follows.

Proposed Algorithm

Input. This includes fingerprint database and test image.

Output. This includes fingerprint authentication.

- (i) The DB3 of FVC2004 fingerprint database is considered.
- (ii) Resize to 256×256 .
- (iii) The CLBP is applied on fingerprints to obtain CLBP sign and magnitude coefficient.
- (iv) The histogram of CLBP sign and magnitude are obtained to form features.
- (v) The 2-level DWT is applied on the fingerprint and second-level LL band coefficients are considered as features.

(vi) The CLBP sign and magnitude histogram of the test and database fingerprint images are compared using difference formula to compute CLBP match score.

- (vii) The LL subband coefficient test and fingerprint images are compared using difference formula to compute DWT score.
- (viii) The matching scores of CLBP and DWT are fused using an arithmetic equation

$$F = P * D + (1 - P) * C. \quad (19)$$

- (ix) The performances parameters are computed using fused matching scores.

The fingerprint identification to authenticate a person effectively on FPGA with optimized parameters is discussed. The spatial domain CLBP and transform domain DWT are used to extract features. The arithmetic fusion is employed on CLBP and DWT match score to compute performance parameters. The algorithm is implemented on Virtex 5 FPGA board. The main objective is to increase TSR, decrease FRR and FAR, and improve hardware optimization parameters.



FIGURE 9: DWT images of different person with matching score 14.2129%.

4. Performance Analysis and Results

In this section, the definitions of performance parameters and performance analysis are discussed.

4.1. Definitions of Performance Parameters

4.1.1. False Rejection Rate (FRR). False Rejection Rate (FRR) is the measure of the number of authorized persons rejected. It is computed using

$$\begin{aligned} \text{\% FRR} \\ = \frac{\text{No. of authorized persons rejected} \times 100}{\text{Total No. of persons in database}}. \end{aligned} \quad (20)$$

4.1.2. False Acceptance Rate (FAR). False Acceptance Rate (FAR) is the measure of the number of unauthorized persons accepted and is computed using

$$\begin{aligned} \text{\% FAR} \\ = \frac{\text{No. of unauthorized persons accepted} \times 100}{\text{Total No. of persons outside the database}}. \end{aligned} \quad (21)$$

4.1.3. Total Success Rate (TSR). Total Success Rate (TSR) is the number of authorized persons successfully matched in the database and is computed using

$$\% \text{ TSR}$$

$$= \frac{\text{No. of authorized persons correctly matched} \times 100}{\text{Total No. of persons in the database}}. \quad (22)$$

4.1.4. Equal Error Rate (EER). Equal error rate (EER) is the point of intersection of FRR and FAR values at particular threshold value. The EER is the tradeoff between FRR and FAR. The value of EER must be low for better performance of an algorithm.

4.2. MATLAB Experimental Results. The performance parameters are computed by running a computer simulation using MATLAB 12.1 version. The performance improvements are explained in this section. The DB3 of FVC2004 fingerprint database is considered for performance analysis. The DB3_A database has one hundred persons with eight samples per person. The size of each fingerprint image is 300×480 with 512 dpi. The database is created by considering the number of persons inside database varied from 30 to 50 with 7 fingerprint samples per person in the database;

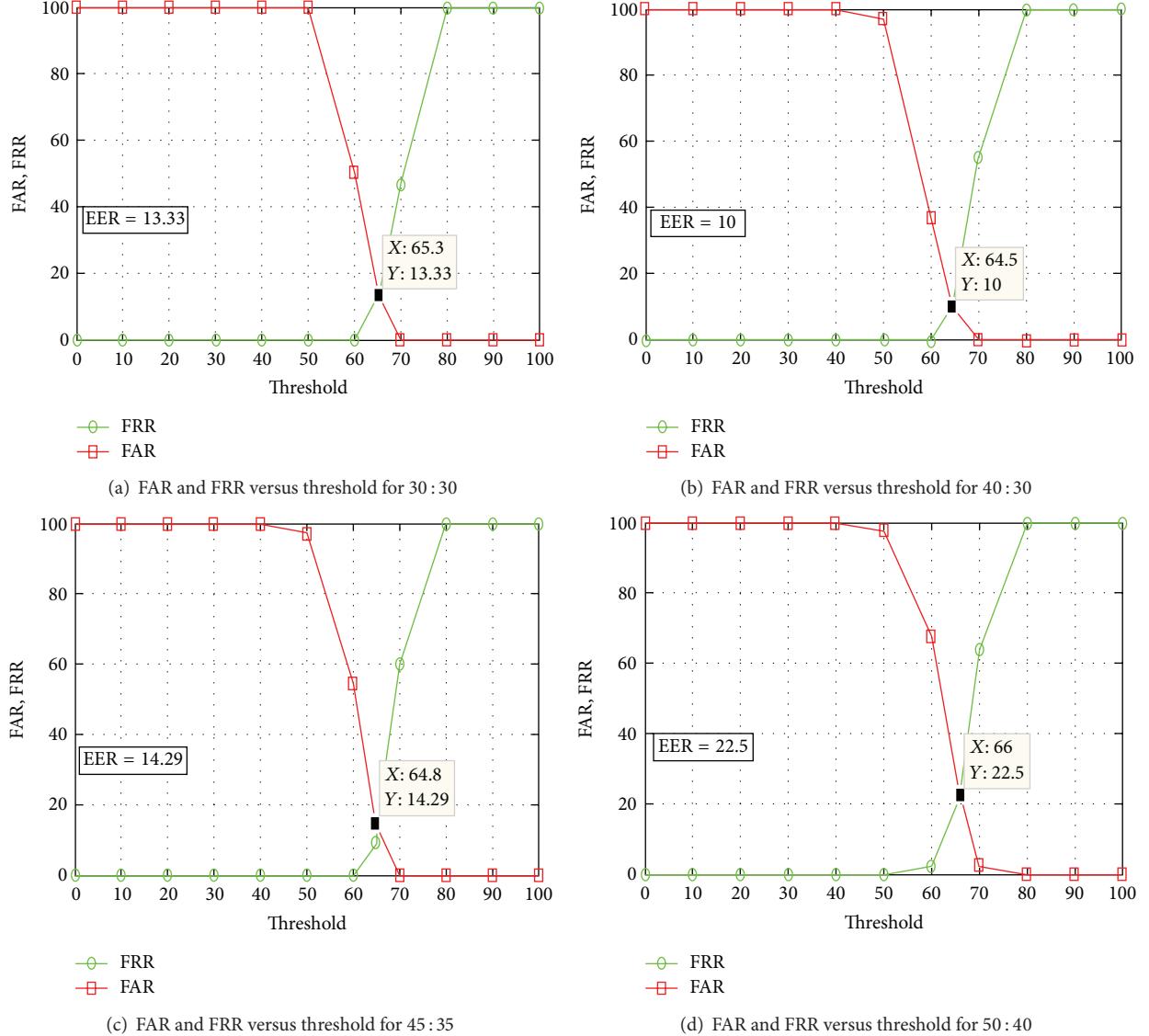


FIGURE 10: FAR and FRR versus threshold for various PID : POD.

that is, number of images varies between 210 and 350 to compute FRR and TSR. The eighth sample of each person is considered as test fingerprint.

4.2.1. CLBP Algorithm. In this section the performance analysis is discussed for features extracted using only CLBP by substituting power factor $P = 0$ in (18). Consider

$$F = C. \quad (23)$$

The variations of FRR and FAR with a threshold for PID : POD combinations of 30 : 30, 40 : 30, 45 : 35, and 50 : 40 are shown in Figure 10. It is observed that for lower threshold values FAR is high and FRR is low. As threshold value increases, FAR decreases from higher values, whereas FRR increases from lower to higher values. The computed values of EERs for different PID and POD combinations of 30 : 30,

40 : 30, 45 : 35, and 50 : 40 are 13.33, 10, 14.29, and 22.5, respectively.

The variations of percentage TSR with threshold for different combinations of PID and POD are given in Table 1. The value of % TSR decreases from higher values to zero as threshold increases. The value of TSR is zero for higher threshold value since the correlation technique is used for matching. It is also observed that as PID increases, the % TSR decreases.

4.2.2. DWT Algorithm. The performance parameters are computed by considering only DWT features by substituting power factor $P = 1$ in (18) to obtain

$$F = D. \quad (24)$$

The variations of FRR and FAR with a threshold for PID : POD of 30 : 30, 40 : 30, 45 : 35, and 50 : 40 are shown

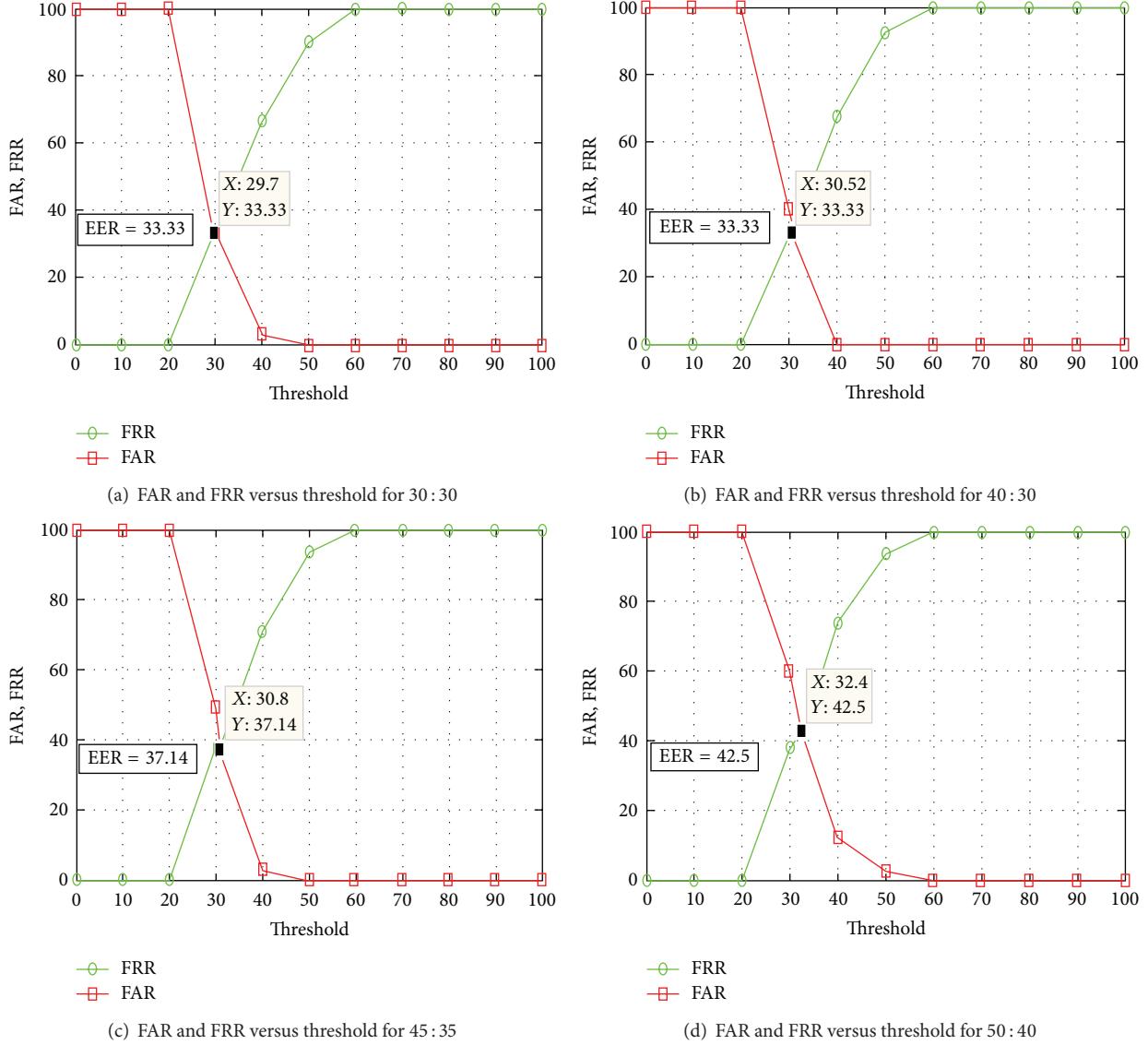


FIGURE 11: FAR and FRR versus threshold for various PID : POD.

TABLE 1: Variations of TSR with threshold for different values of PID and POD.

Threshold	% TSR				
	PID : POD	30 : 30	40 : 30	45 : 35	50 : 40
40		96.7	90	88.9	88
50		96.7	90	88.9	88
60		96.7	90	88.9	88
65		86.67	80	80	68
70		53.3	42.5	37.8	34
80		0	0	0	0

in Figure 11. It is observed that for lower threshold values FAR is high and FRR is low. As threshold value increases, FAR decreases from higher values, whereas FRR increases

from lower to higher values. The computed values of EERs in percentage for different PID and POD combinations of 30 : 30, 40 : 30, 45 : 35, and 50 : 40 are 33.33, 33.33, 37.14, and 42.5, respectively. The variations of percentage TSR with threshold for different combinations of PID and POD are given in Table 2. The value of % TSR decreases from higher values to zero as threshold increases. The value of TSR is zero for higher threshold value since the correlation technique is used for matching. It is also observed that as PID increases the percentage TSR value decreases.

4.2.3. Fusion of CLBP and DWT. The performance parameters are computed considering fusion based given by (18).

The variations of FRR and FAR with a threshold for PID : POD of 30 : 30, 40 : 30, 45 : 35, and 50 : 40 are shown in Figure 12. It is observed that for lower threshold values FAR is high and FRR is low. As threshold value increases,

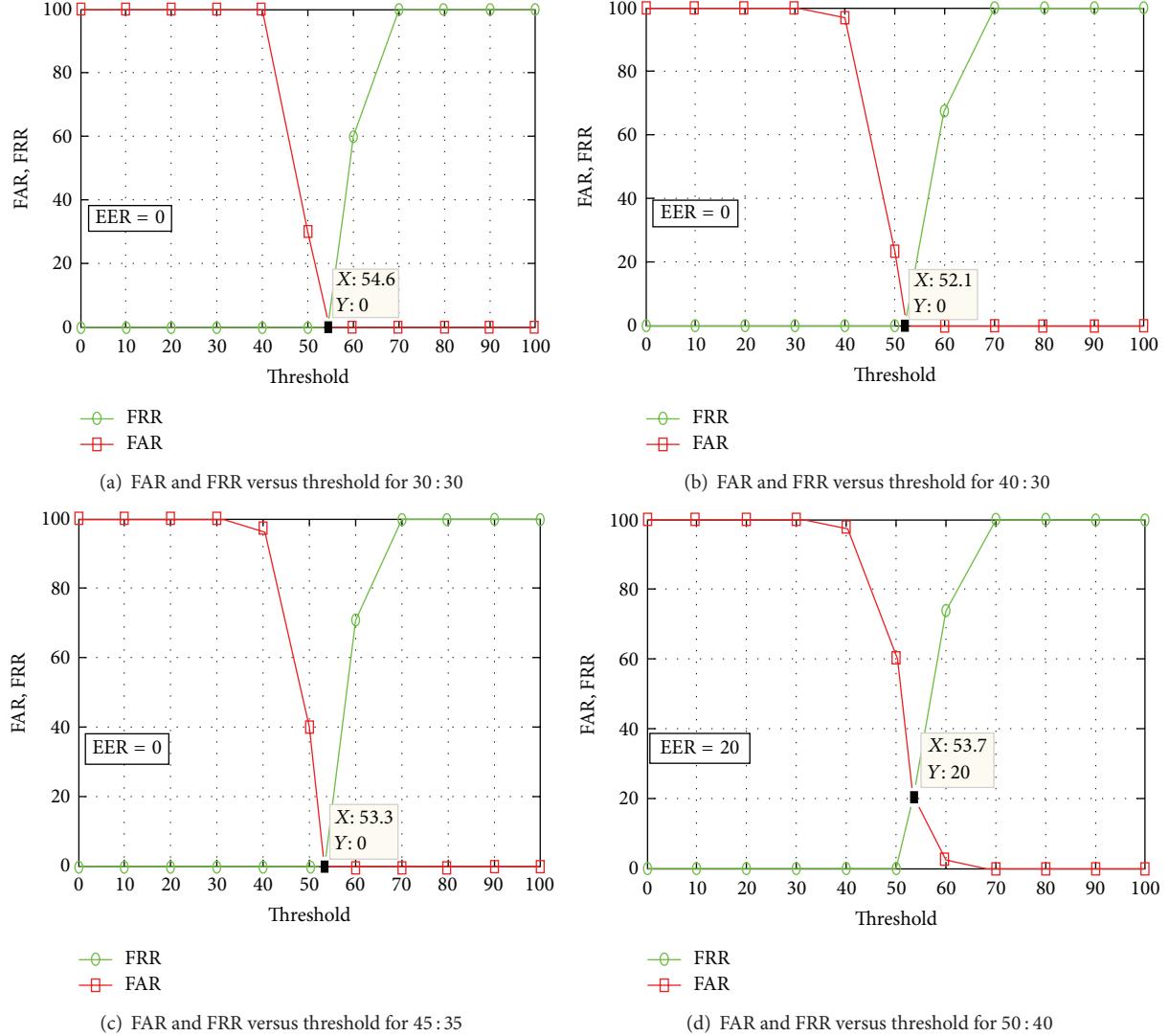


FIGURE 12: FAR and FRR versus threshold for various PID : POD.

TABLE 2: Variations of TSR with threshold for different values of PID and POD.

Threshold	% TSR				
	PID : POD	30 : 30	40 : 30	45 : 35	50 : 40
20		36.7	27.5	24.44	26
30		23.33	27.5	20	20
40		13.33	12.5	11.11	10
50		0	0	0	0
60		0	0	0	0

FAR decreases from higher values, whereas FRR increases from lower to higher values. The computed values of EERs in percentage for different PID and POD combinations of 30:30, 40:30, 45:35, and 50:40 are 0, 0, 0, and 20, respectively.

TABLE 3: Variations of TSR with threshold for different values of PID and POD.

Threshold	% TSR				
	PID : POD	30 : 30	40 : 30	45 : 35	50 : 40
40		100	100	97.7	98
50		100	100	97.7	98
55		100	100	97.7	78
60		40	32.5	28.3	26
70		0	0	0	0
80		0	0	0	0

The variations of percentage TSR with threshold for different combinations of PID and POD are given in Table 3. The values of % TSR decrease from higher values to zero as threshold increases. The value of TSR is zero for higher threshold value since the correlation technique is used for

TABLE 4: Variations of EER and TSR for CLBP, DWT, and proposed model.

PID	POD	CLBP		DWT		Proposed model	
		EER (%)	TSR (%)	EER (%)	TSR (%)	EER (%)	TSR (%)
30	30	13.3	86.67	33.33	23.33	0	100
40	30	13.5	80	33.33	22.5	0	100
45	35	14.29	80	37.14	20	0	97.78
50	40	22	68	42.5	20	20	80

TABLE 5: Comparison of percentage EER and TSR existing techniques with proposed fusion based method.

Author	Techniques	% EER	% TSR
Karki and Sethu Selvi [40]	Curvelet-Euclidean	—	95.02
Bartunek et al. [41]	Adaptive fingerprint enhancement	6.2	—
Ouzounoglou et al. [42]	Self-Organizing Maps (SOM) algorithm	5.86	—
Medina-Pérez et al. [43]	Minutiae triplets	5.1	—
Proposed method	Fusion of CLBP and DWT	0	97.78

matching. It is also observed that as PID increases the percentage TSR value decreases.

4.2.4. Comparison between CLBP, DWT, and Proposed Model. The values of EER and TSR with different combinations of PID and POD are tabulated in Table 4. The value of %TSR decreases and EER increases with increase in PID and POD. The proposed model achieves reduced EER and increased TSR compared to individual technique of CLBP and DWT implementation. The performance parameters such as EER and TSR are compared with existing techniques published by Karki and Sethu Selvi [40], Bartunek et al. [41], Ouzounoglou et al. [42], and Medina-Pérez et al. [43] for FVC2004 DB3 Database given in Table 5. The proposed model achieves reduced EER and increased TSR.

5. FPGA Implementation of Proposed Model

The proposed architectures are implemented on FPGA device using Virtex xc5vlx30T [44] with speed grade 3 and designed to work with external SRAM memory [45] which is used to store the database and test images. This SRAM has been required since the on-chip memory of FPGA is small to store the database and test images during algorithm execution.

5.1. CLBP Architecture. The CLBP algorithm is synthesized using CLBP VLSI architecture shown in Figure 13. The nine shift registers of eight bits along with two shift registers each of length 2008 bits are used to form First In First Out (FIFO) architecture to implement 3×3 matrix. The outputs p0, p1, and p2 are three pixels of *first* row, p3, p4, and p5 are three

pixels of *second* row, that is, exactly below first three pixels, and p6, p7, and p8 are three pixels of *third* row, that is, exactly below *second* row three pixels to form 3×3 matrix for sign and magnitude computations with CLBP. In the next rising edge of the clock, pixels are shifted to the right by one to form a new 3×3 matrix. The control unit along with 10-bit and 8-bit counters is used to create new matrices which are sent to compute CLBP_S and CLBP_M blocks to obtain CLBP sign and magnitude components. These components are further used to obtain the histogram magnitude and sign CLBP features using two counter banks of 16×256 .

5.1.1. Finite State Machine (FSM) of CLBP. The MODELSIM FSM view window is used to display state diagram. The FSM of control unit to compute CLBP_M and CLBP_S is shown in Figure 14. The st0 is the initial state of control unit and is continued in this state until 10-bit counter counts 515 clock cycles to allow the FIFO architecture to store pixel values of the *first* row, *second* row, and first three pixels of *third* row. The st0 state shifts to st1 state after 515 clock cycles. In st1 state first 3×3 matrices of *first*, *second*, and *third* rows are considered, sending q1 from 10-bit counter to control unit to activate s1 to compute CLBP_S and s2 to compute CLBP_M. The sign and magnitudes of CLBP for successive 3×3 matrices of the first *three* rows are computed in st1 till 8-bit counter count reaches 253 and shifts to st2. The 8-bit counter count is reset and 10-bit counter count is incremented in state st2 to store *fourth* row by eliminating first row and shifts to st3 to create clock cycles delay before shifts to st1. The processes of CLBP_S and CLBP_M for every 3×3 matrix for all rows of an image are computed in st1, st2, and st3 states. Once all rows are processed state shifts st0 to read next image.

5.2. DWT Architecture. The DWT algorithm is synthesized using DWT architecture as shown in Figure 15. In case of DWT, 2×2 nonoverlapping matrix is required. The four shift registers of eight bits along with one shift register of length 254 bits are used for FIFO architecture to form 2×2 matrix. The outputs p0 and p1 are two pixels of the *first* row and p2 and p3 are pixels exactly below the *first* row to form 2×2 matrix for DWT computation. The control unit controls all the timing issues using a 9-bit counter. Its operation is based on the state diagram shown in Figure 16. In st0, the entire *first* row and two pixels of the *second* row of an image are read using 9-bit counter and shifts to st1. The LL of 2×2 matrix is computed in st1 and continued to compute LL coefficient with nonoverlapping pixels of the *second* row. Once *first* and *second* rows were completed, then st1 shifts to st2 and back

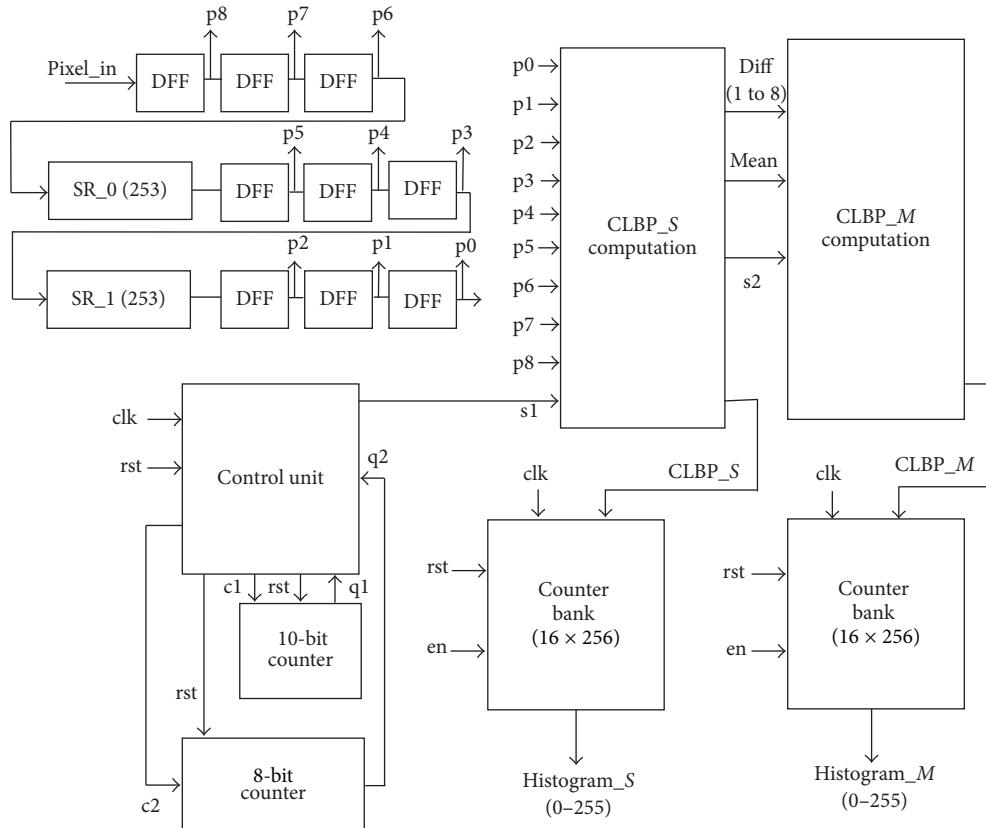


FIGURE 13: VLSI architecture of CLBP.

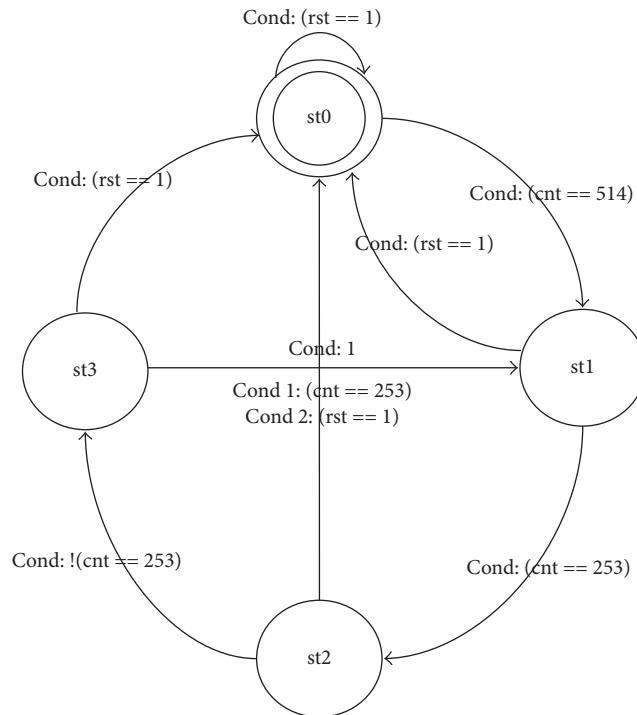


FIGURE 14: FSM of CLBP control unit.

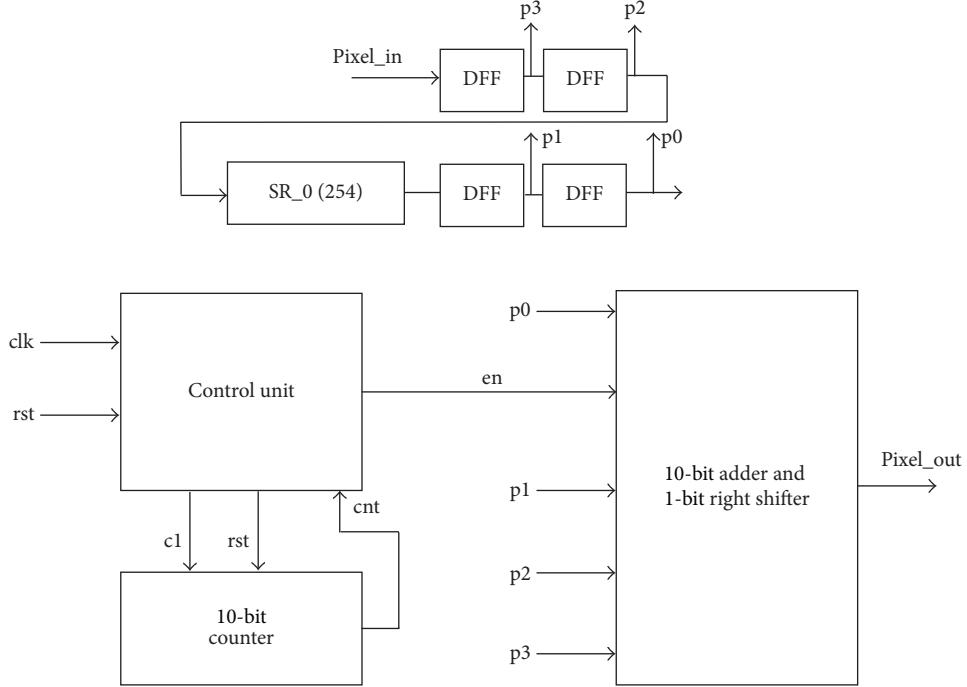


FIGURE 15: VLSI architecture of Level 1 DWT.

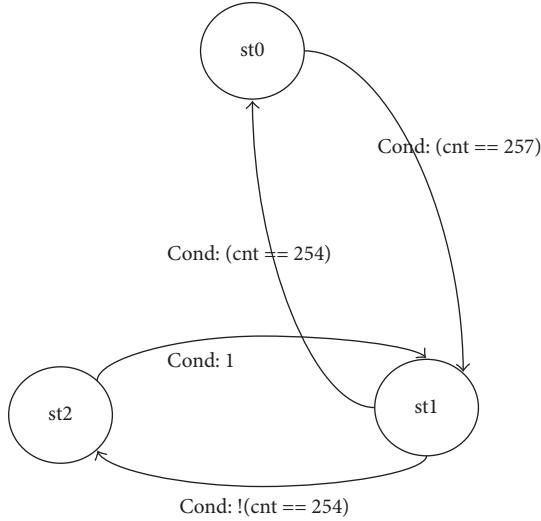


FIGURE 16: FSM of DWT Level 1 control unit.

to st1 after two clock cycles' delay to compute LL coefficients for *third* and *fourth* rows and is continued till all rows are completed in an image. The entire image of 256×256 will give rise to 128×128 DWT coefficients for 65536 clock cycles.

The algorithm for DWT Level 2 remains the same as DWT Level 1 but is applied only on the LL component of DWT Level 1 coefficients. The 128×128 LL coefficients generated by Level 1 are again processed here to generate 64×64 coefficients. Now, instead of waiting for Level 1 to complete its processing and then executing Level 2, the pipelining has been done in both the stages to achieve better speed. The LL

coefficients for every overlapping and nonoverlapping 2×2 matrix in an image are generated in Level 1 using moving window architecture and are connected to registers of Level 2 architecture shown in Figure 17. The LL coefficients of a nonoverlap 2×2 matrix of Level 1 are considered in Level 2 for further decomposition. The architecture uses a 10-bit counter, 11-bit adder, and 1 right shift register. The controller uses the counter to keep track of time and all the LL coefficient values P0, P1, P2, and P3 are added using 11-bit adder since all the values are about 9 bits. The result of the addition is scaled down by 2 by using one-bit right shift operation.

The FSM of Level 2 DWT to generate LL coefficients is shown in Figure 18. In state st0, the 2×2 LL coefficients of Level 1 are read and jump to st1. The LL coefficients of Level 2 are computed in st1 by adding and shift technique. The states st2, st3, and st4 are used to create a delay to compute Level 2 coefficient of next nonoverlapped 2×2 windows. The process is continued until all nonoverlapped 2×2 windows are exhausted.

5.3. Matching Score Architecture for CLBP and DWT. The architecture for computation of the matching score in percentage for both CLBP and DWT is shown in Figure 19. The feature of the test image is subtracted from that of database feature and if the difference is less than the threshold, then it is considered to be a match and the counter is updated. Similarly, after comparing all the features, the control unit asserts cnt_out signal to use the content of counter to calculate the match score; also this signal is used to reset the counter to zero. The match score in percentage is obtained by multiplying the number of matched features by 100 and then dividing it by the total number of features as given in (8) and (17). This

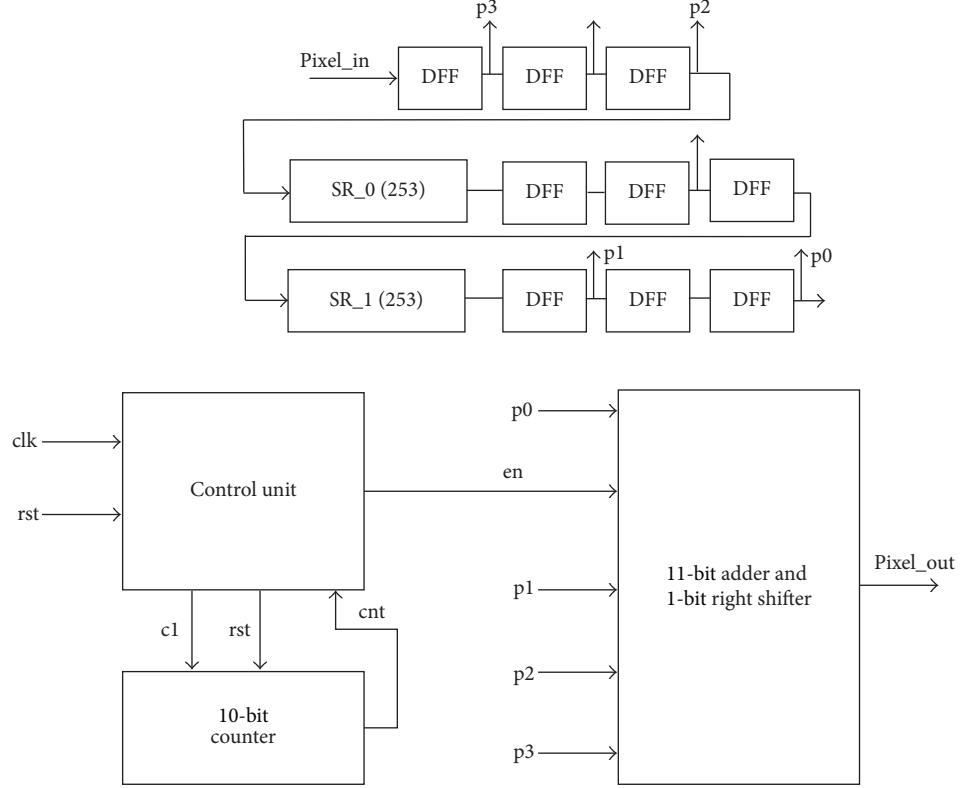


FIGURE 17: VLSI architecture of Level 2 DWT.

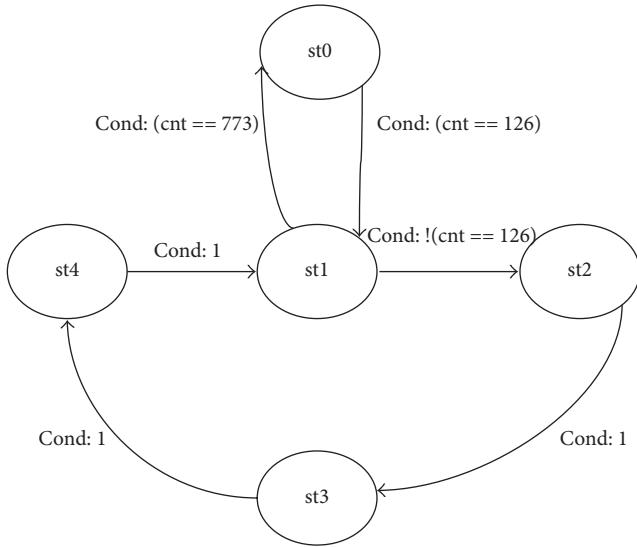


FIGURE 18: FSM of DWT Level 2 control unit.

requires a dedicated multiplier and divider which consumes more hardware and decreases the speed. In our architecture multiplication and division operation can be performed using shift registers reducing the area and increasing the speed. The numbers of matched features are shifted left by 6, 5, and 2 bits and then add to achieve multiplication by 100. Similarly the

division is performed by shifting right by 9 bits and 14 bits for CLBP score and DWT score, respectively.

5.4. Architecture for Fusion of CLBP and DWT Match Scores. The architecture for fusion of CLBP and DWT match scores using improvement factor using (18) is as shown in Figure 20. The process of multiplication with a fractional part like 0.7 and 0.3 is carried-out shift operation in three steps. To obtain 0.7, the parallel combination of 1 bit, 3 bits, and 4 bits right shift registers, this yields a value of $0.5 + 0.125 + 0.0625 = 0.6875$ resulting in an error of 1.8%. Similarly to obtain 0.3, the parallel combination of 2 bits, 5 bits, and 6 bits right shift registers, this yields a value of $0.25 + 0.0313 + 0.016 = 0.2969$ resulting in an error of 2.3%. These errors are negligible since the threshold used for decision circuit is not hard. Finally, the fusion match score is compared with a threshold and the decision is made whether the test sample is matched with database or not. This method of implementation eliminates the use of dedicated floating point and fixed point multiplier and divider circuits consuming more clock cycles and area.

5.5. Hardware Results. The performance parameters based on FPGA for CLBP, DWT, and fusion based architectures are given in Table 6. The limitation of fusion technique is that it requires more number of slice registers and LUTs as compared to individual technique.

The RTL schematics of the proposed fusion based design with CLBP and DWT architecture, run in parallel, is shown

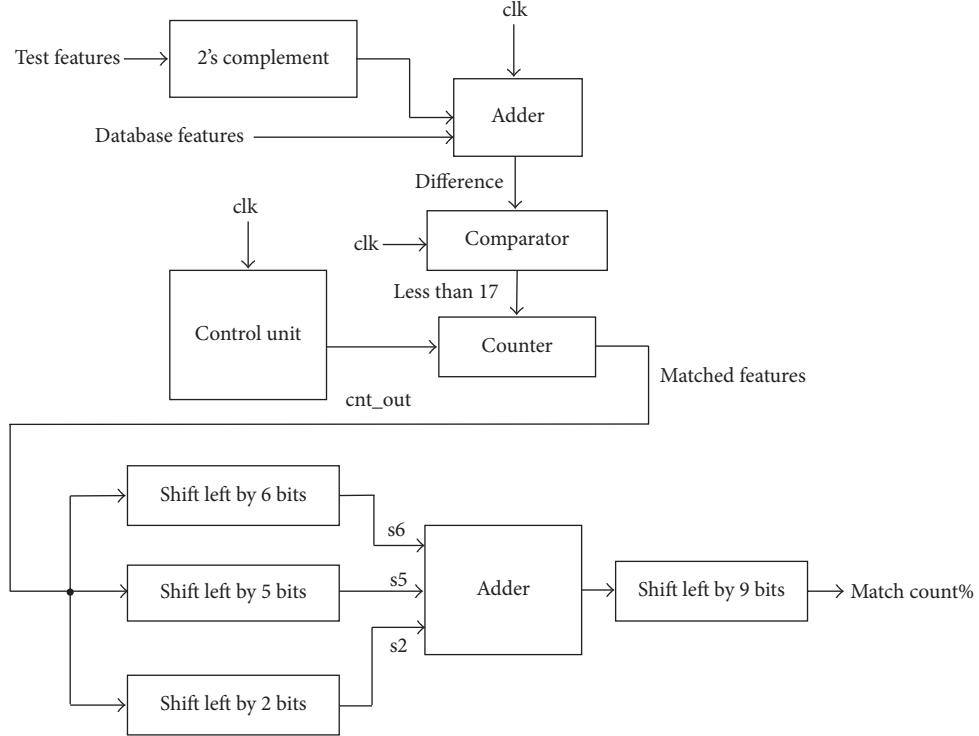


FIGURE 19: VLSI architecture for CLBP and DWT matching.

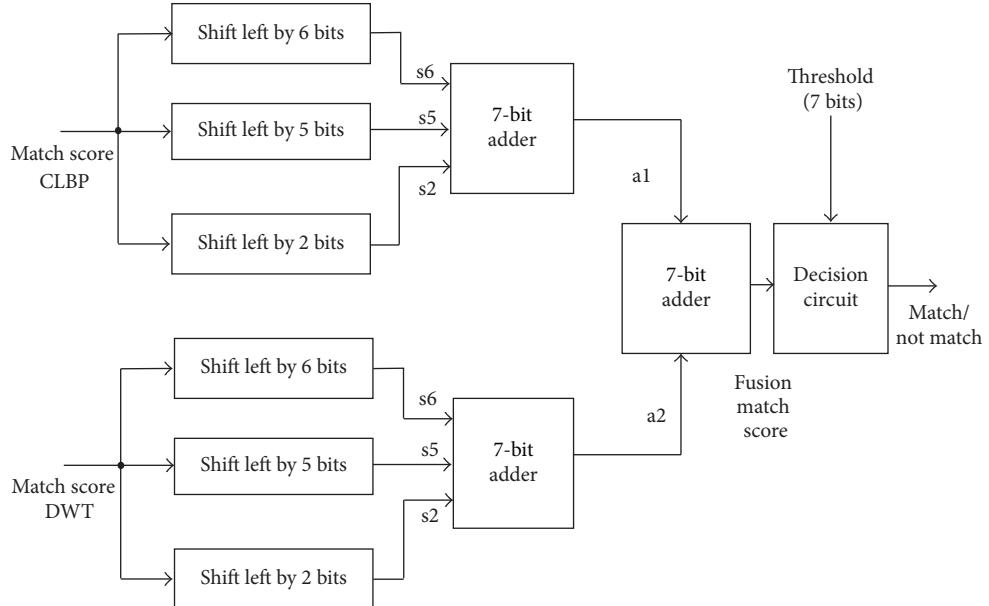


FIGURE 20: VLSI architecture for fusion.

in Figure 21. The CLBP system consists of CLBP module and CLBP matching with its output CLBP match score. The DWT system consists of 2-level decomposition modules and finally DWT matching module with its output DWT match score. The two levels of DWT decomposition are pipelined in order to achieve high speed. Finally, using fusion module, both

the match scores are combined using strength factors and the fusion match score is computed upon which threshold is applied to decide whether a match has occurred or not.

The routing design on the FPGA connecting several CLBs and block RAM is shown in Figure 22. The blue streaks indicate the connection between the logical blocks. Similarly,

TABLE 6: Performance parameters on FPGA.

Block	CLBP system	DWT system	Fusion
Number of slice registers	8435	359	8776
Number of slice LUTs	20243	440	20706
Max freq (MHz)	68.036	298.77	68.004

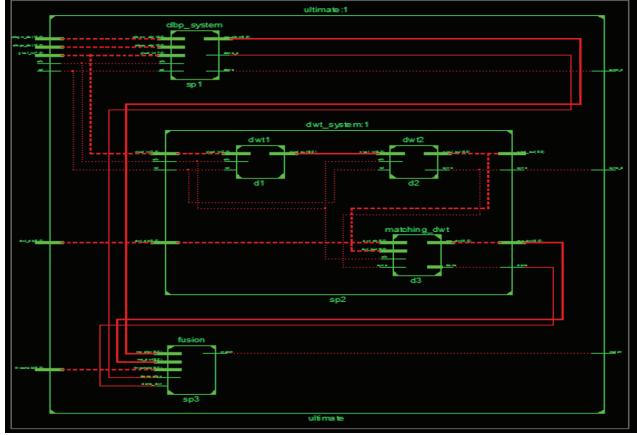


FIGURE 21: RTL schematic of the entire system.

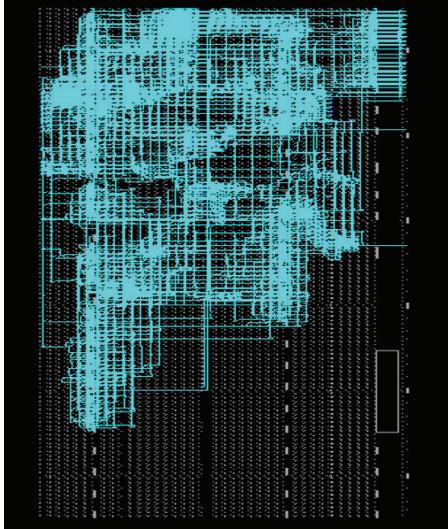


FIGURE 22: Routed design view.

a schematic showing all interconnections between the LUTs, BLOCK RAM, and IOBs (input/output buffers) is shown in Figure 23.

The schematic floor of our proposed design using Virtex 5 device is as shown in Figure 24. This snapshot is taken from a tool known as Xilinx PlanAhead. The small violet rectangular boxes represent CLBs (combinational logic block). The CLB consists of two slices and each slice has 4 LUTs, 3 multiplexers, 1 dedicated arithmetic logic (two 1-bit adders and a carry chain), and four 1-bit registers that can be configured either as flip-flops or as latches shown in Figure 25. This technique of hardware implementation does not require dedicated

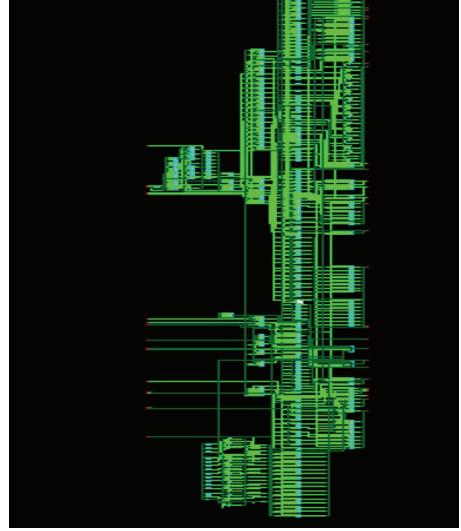


FIGURE 23: Schematic of the proposed design.

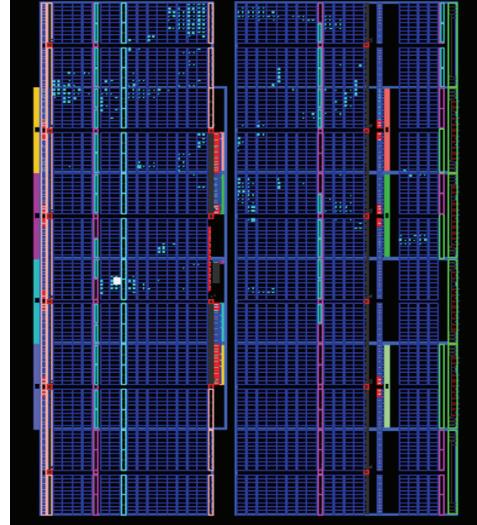


FIGURE 24: Floor plan of the design.

multiplier and divider; hence it consumes less hardware to build and it is faster.

5.5.1. Comparison between Existing Fingerprint Architectures and Proposed Architecture. The area and total execution time estimated using FPGA for the proposed algorithm and existing algorithm are presented in Table 7. In case of comparing the proposed results with previous related work, it is better with respect to different aspects.

- (a) In [46] authors presented a hardware-software code-sign of fingerprint recognition system. The coprocessors were used to speed up the execution time of algorithm resulting in 988 ms. The microblaze soft core processor along with coprocessor limits the speed of the entire system. In our proposed method the implementation of full flexible parallel and pipelined

TABLE 7: Comparison of area and execution time of existing method with proposed method on FPGA.

Author	Board	Slices (% utilization)	Total execution time
Lopez and Canto [46]	Spartan 3 XC3S2000	3507/20480 (17.12%)	987.8 ms at 40 MHz
Fons et al. [47]	Altera ARM + FPGA EPXA10 DDR	29327/38400 (76.37%)	955.842 ms
Conti et al. [48]	Xilinx Virtex II XC2V3000	12,863/14,336 (89.72%)	183.32 ms at 22.5 MHz
Proposed method	Xilinx Virtex xc5vlx30T	8776/19200 (45.70%)	1.644 ms at 68 MHz

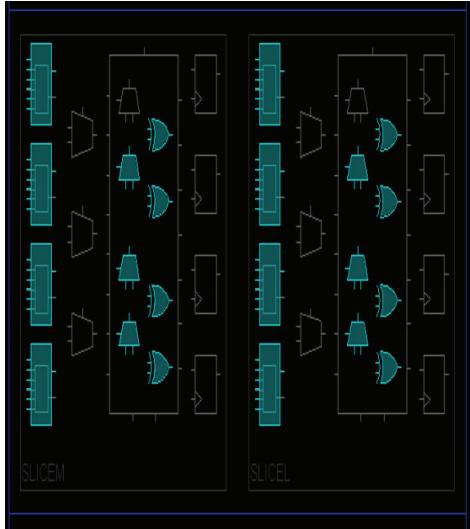


FIGURE 25: CLB consisting of two slices.

architecture using on-chip slices of FPGA improves the system matching speed.

- (b) In [47] authors proposed a solution of fingerprint recognition using a combination of ARM and FPGA. The use of full FPGA reconfiguration in our proposed method using Virtex 5 with SRAM is better in speed compared to reconfiguration latencies achieved using a combination of ARM and FPGA (EPXA10 DDR).
- (c) In [48] a sensor has been prototyped via FPGA to improve the speed of the system with best elaboration time of 183.32 ms and a working frequency of 22.5 MHz. In our method the elaboration time of 1.644 ms is achieved at the working frequency of 68 MHz, since the external SRAM is used to port via FPGA.

Limitations. There exist some limitations in the proposed method despite the improvement of speed as it requires more area since the CLBP and DWT fused technique is used. The on-chip moving window FIFO architecture designed has initial clock latencies.

6. Conclusion

In this paper, efficient FSM based reconfigurable architecture for fingerprint recognition implemented using Virtex

5 FPGA board is proposed. The novel matching score of CLBP is computed using histogram CLBP features of test image fingerprint images in the database. Similarly the DWT matching score is computed using DWT features of test image and fingerprint images in the database. The arithmetic fusion equation with improvement factor is used to combine the matching scores generated by histogram CLBP features and DWT features. The performance parameters are computed using fusion scores with correlation technique. It is observed that the values of EER, FAR, FRR, TSR, and hardware parameters such as area and delay are better in the case of proposed method compared to existing methods.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to Biometric*, Springer, Berlin, Germany, 2004.
- [2] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.
- [3] D. Hagan, "Biometric systems & big dat," in *Proceedings of the Big Data Conference*, 2012.
- [4] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *Journal of Big Data*, vol. 2, no. 1, article 8, 20 pages, 2014.
- [5] H. Xu and R. N. J. Veldhuis, "Spectral minutiae representations of fingerprints enhanced by quality data," in *Proceedings of the IEEE 3rd International Conference on Biometrics: Theory, Applications and Systems (BTAS '09)*, pp. 1–5, Washington, DC, USA, September 2009.
- [6] D. Mulyono and H. S. Jinn, "A study of finger vein biometric for personal identification," in *Proceedings of the IEEE International Symposium on Biometrics and Security Technologies (ISBAST '08)*, pp. 1–8, Islamabad, Pakistan, April 2008.
- [7] R. Cappelli, D. Maio, and D. Maltoni, "Semi-automatic enhancement of very low quality fingerprints," in *Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis (ISPA '09)*, pp. 678–683, September 2009.
- [8] M. H. V. Talele, P. V. Talele, and S. N. Bhutada, "Study of local binary pattern for partial fingerprint identification," *International Journal of Modern Engineering Research*, vol. 4, no. 9, pp. 55–61, 2014.
- [9] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE*

- Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.
- [10] M. K. Shinde and S. A. Annadate, “Study of different methods for gender identification using fingerprints,” *International Journal of Application or Innovation in Engineering & Management*, vol. 3, no. 10, pp. 194–199, 2014.
 - [11] K. Tewari and R. L. Kalakoti, “Fingerprint recognition using transform domain techniques,” in *Proceedings of the International Technological Conference*, pp. 136–140, 2014.
 - [12] M. ShabanAl-Ani and W. M. Al-Aloosi, “Biometrics fingerprint recognition using Discrete Cosine Transform (DCT),” *International Journal of Computer Applications*, vol. 69, no. 6, pp. 44–48, 2013.
 - [13] A. Pokhriyal and S. Lehri, “A new method of fingerprint authentication using 2D wavelets,” *Journal of Theoretical and Applied Information Technology*, vol. 13, no. 2, pp. 131–138, 2010.
 - [14] J. D. B. Nelson and N. G. Kingsbury, “Enhanced shift and scale tolerance for rotation invariant polar matching with dual-tree wavelets,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 814–821, 2011.
 - [15] U. Gawande, M. Zaveri, and A. Kapur, “A novel algorithm for feature level fusion using SVM classifier for multibiometrics-based person identification,” *Applied Computational Intelligence and Soft Computing*, vol. 2013, Article ID 515918, 11 pages, 2013.
 - [16] L. Nanni, C. Casanova, S. Brahnam, and A. Lumini, “Empirical tests on enhancement techniques for a hybrid fingerprint matcher based on minutiae and texture,” *International Journal on Artificial Intelligence Tools*, vol. 4, no. 1, pp. 1–10, 2012.
 - [17] G. Danese, M. Giachero, F. Leporati, G. Matrone, and N. Nazzicari, “An FPGA-based embedded system for fingerprint matching using phase-only correlation algorithm,” in *Proceedings of the 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools (DSD ’09)*, pp. 672–679, Patras, Greece, August 2009.
 - [18] A. H. A. Razak and R. H. Taharim, “Implementing Gabor filter for fingerprint recognition using Verilog HDL,” in *Proceedings of the 5th International Colloquium on Signal Processing and Its Applications (CSPA ’09)*, pp. 423–427, IEEE, Kuala Lumpur, Malaysia, March 2009.
 - [19] L. Hermanto, S. A. Sudiro, and E. P. Wibowo, “Hardware implementation of fingerprint image thinning algorithm in FPGA device,” in *Proceedings of the International Conference on Networking and Information Technology (ICNIT ’10)*, pp. 187–191, Manila, Philippines, June 2010.
 - [20] M. Fons, F. Fons, and E. Cantó, “Fingerprint image processing acceleration through run-time reconfigurable hardware,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 12, pp. 991–995, 2010.
 - [21] S. Yoon, J. Feng, and A. K. Jain, “Altered fingerprints: analysis and detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 451–464, 2012.
 - [22] M. Vatsa, R. Singh, A. Noore, and S. K. Singh, “Quality induced fingerprint identification using extended feature set,” in *Proceedings of the IEEE 2nd International Conference on Biometrics: Theory, Applications and Systems (BTAS ’08)*, pp. 1–6, Arlington, Va, USA, October 2008.
 - [23] M. Govan and T. Buggy, “A computationally efficient fingerprint matching algorithm for implementation on smartcards,” in *Proceedings of the 1st IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS ’07)*, pp. 1–6, Crystal City, Va, USA, September 2007.
 - [24] N. Nain, B. Bhadviya, B. Gautam, D. Kumar, and B. M. Deepak, “A fast fingerprint classification algorithm by tracing ridge-flow patterns,” in *Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems (SITIS ’08)*, pp. 235–238, Bali, Indonesia, December 2008.
 - [25] J. Li, S. Tulyakov, and V. Govindaraju, “Verifying fingerprint match by local correlation methods,” in *Proceedings of the 1st IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS ’07)*, pp. 1–5, IEEE, Crystal City, VA, USA, September 2007.
 - [26] D. Singh, D. P. Singh, and D. R. Shukla, “Fingerprint recognition system based on mapping approach,” *International Journal of Computer Applications*, vol. 5, no. 2, pp. 1–5, 2010.
 - [27] S. B. Nikam and S. Agarwal, “Fingerprint anti-spoofing using ridgelet transform,” in *Proceedings of the 2nd International Conference on Biometrics: Theory, Applications and Systems (BTAS ’08)*, pp. 1–6, Arlington, Va, USA, October 2008.
 - [28] A. D. Masmoudi, R. B. Trabelsi, and D. S. Masmoudi, “A new biometric human identification based on fusion fingerprints and fingerveins using MonoLBP descriptor,” *World Academy of Science, Engineering and Technology*, vol. 78, pp. 1658–1663, 2013.
 - [29] R. F. Stewart, M. Estevao, and A. Adler, “Fingerprint recognition performance in rugged outdoors and cold weather conditions,” in *Proceedings of the IEEE 3rd International Conference on Biometrics: Theory, Applications and Systems (BTAS ’09)*, pp. 1–6, Washington, DC, USA, September 2009.
 - [30] J.-Z. Cao and Q.-Y. Dai, “A novel online fingerprint segmentation method based on frame-difference,” in *Proceedings of the International Conference on Image Analysis and Signal Processing (IASP ’09)*, pp. 57–60, April 2009.
 - [31] K. Umamaheswari, S. Sumathi, S. N. Sivanandam, and K. K. N. Anburajan, “Efficient finger print image classification and recognition using neural network data mining,” in *Proceedings of the International Conference on Signal Processing, Communications and Networking (ICSCN ’07)*, pp. 426–432, IEEE, Chennai, India, February 2007.
 - [32] V. Conti, C. Militello, F. Sorbello, and S. Vitabile, “Introducing Pseudo-Singularity points for efficient fingerprints classification and recognition,” in *Proceedings of the 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS ’10)*, pp. 368–375, Krakow, Poland, February 2010.
 - [33] F. Ahmed, E. Hossain, A. S. M. Hossain Bari, and Md. Sakhawat Hossen, “Compound Local Binary Pattern (CLBP) for rotation invariant texture classification,” *International Journal of Computer Applications*, vol. 33, no. 6, pp. 5–9, 2011.
 - [34] A. A. Paulino, J. Feng, and A. K. Jain, “Latent fingerprint matching using descriptor-based hough transform,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 31–45, 2013.
 - [35] J. Feng, J. Zhou, and A. K. Jain, “Orientation field estimation for latent fingerprint enhancement,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 925–940, 2013.
 - [36] Third Fingerprint Verification Competition, *FVC 2004 Database Reference*, 2004, <http://bias.csr.unibo.it/fvc2004/download.asp>.
 - [37] C. Wen, T. Guo, and Y. Zhou, “A novel and efficient algorithm for segmentation of fingerprint image based on LBP operator,” in *Proceedings of the International Conference on Information Technology and Computer Science (ITCS ’09)*, pp. 200–204, Kiev, Ukraine, July 2009.

- [38] D. V. Jadhav and P. K. Ajmera, "Multi resolution feature based subspace analysis for fingerprint recognition," *International Journal of Computer Applications*, vol. 1, no. 13, pp. 1–4, 2010.
- [39] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [40] M. V. Karki and S. Sethu Selvi, "Multimodal biometrics at feature level fusion using texture features," *International Journal of Biometrics and Bioinformatics*, vol. 7, no. 1, pp. 58–72, 2013.
- [41] J. S. Bartunek, M. Nilsson, B. Sallberg, and I. Claesson, "Adaptive fingerprint image enhancement with emphasis on preprocessing of data," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 644–656, 2013.
- [42] A. N. Ouzounoglou, P. A. Asvestas, and G. K. Matsopoulos, "A new approach in fingerprint matching based on a competitive learning algorithm," *International Journal of Information and Communication Technology Research*, vol. 2, no. 9, pp. 723–731, 2012.
- [43] M. A. Medina-Pérez, M. García-Borrotto, A. E. Gutierrez-Rodríguez, and L. Altamirano-Robles, "Improving fingerprint verification using minutiae triplets," *Sensors*, vol. 12, no. 3, pp. 3418–3437, 2012.
- [44] Xilinx Veritrix-5 Family Datasheet, 2012, http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf.
- [45] SRAM Datasheet, 2012, [http://www.cypress.com/search/all?f\[0\]=meta_type%3Atechnical_documents](http://www.cypress.com/search/all?f[0]=meta_type%3Atechnical_documents).
- [46] M. Lopez and E. Canto, "FPGA implementation of a minutiae extraction fingerprint algorithm," in *Proceedings of the Biometric Signal Processing Conference*, pp. 21–25, September 2008.
- [47] M. Fons, F. Fons, E. Cantó, and M. López, "FPGA-based personal authentication using fingerprints," *Journal of Signal Processing Systems*, vol. 66, no. 2, pp. 153–189, 2012.
- [48] V. Conti, S. Vitabile, G. Vitello, and F. Sorbello, "An embedded biometric sensor for ubiquitous authentication," in *Proceedings of the AEIT Annual Conference: Innovation and Scientific and Technical Culture for Development (AEIT '13)*, pp. 1–6, Palermo, Italy, October 2013.

