Latent Fingerprint Image Segmentation using Fractal Dimension Features and Weighted Extreme Learning Machine Ensemble

Jude Ezeobiejesi and Bir Bhanu
Center for Research in Intelligent Systems
University of California at Riverside, Riverside, CA 92521, USA
e-mail: jezeobie@cs.ucr.edu, bhanu@cris.ucr.edu

Abstract

Latent fingerprints are fingerprints unintentionally left at a crime scene. Due to the poor quality and often complex image background and overlapping patterns characteristic of latent fingerprint images, separating the fingerprint region-of-interest from complex image background and overlapping patterns is a very challenging problem. In this paper, we propose a latent fingerprint segmentation algorithm based on fractal dimension features and weighted extreme learning machine. We build feature vectors from the local fractal dimension features and use them as input to a weighted extreme learning machine ensemble classifier. The patches are classified into fingerprint and non-fingerprint classes. We evaluated the proposed segmentation algorithm by comparing the results with the published results from the state of the art latent fingerprint segmentation algorithms. The experimental results of our proposed approach show significant improvement in both the false detection rate (FDR) and overall segmentation accuracy compared to the existing approaches.

1. Introduction

Latent fingerprint segmentation is of fundamental importance to the automation of latent fingerprint processing. Automatic latent fingerprint segmentation with a high degree of accuracy can drastically reduce the time spent by latent examiners in processing latent fingerprints. By more accurately defining the region of interest (ROI), the number of false minutiae (ridge endings and bifurcations) that are extracted by automated systems can be reduced, leading to better fingerprint matching results. In recent years, the accuracy of latent fingerprint identification by latent fingerprint forensic examiners has been the subject of increased study, scrutiny, and commentary in the legal system and the forensic science literature. Errors in latent fingerprint matching can be devastating, resulting in missed opportunities to apprehend criminals or wrongful convictions of innocent people. Several high-profile cases in the United States and abroad during the past 20 years have shown that forensic examiners can sometimes make mistakes when analyzing or comparing fingerprints [7]. Compared to rolled and plain fingerprints, latent fingerprints have significantly poor quality ridge structure and large non-linear distortions. As shown in Figure 1, latent fingerprint images contain background structured noise such as speckles, stains, lines, arcs, and sometimes text. Due to the poor quality and often complex image background and overlapping patterns characteristic of latent fingerprint images, separating the fingerprint region of interest from complex image background and overlapping patterns is a very challenging problem [17].

To process latent fingerprints, latent experts manually mark the region-of-interest (*ROI*) in latent fingerprints and use the *ROI* to search large databases of reference full fingerprints and identify a small number of potential matches for manual examination. Given the large size of law enforcement databases containing rolled and plain fingerprints, it is very desirable to perform latent fingerprint processing in a fully automated way. As a step in this direction, our paper proposes an efficient technique for separating latent fingerprints from the complex image background using fractal dimension features and weighted extreme learning machine ensemble. To the best of our knowledge, no previous work has used this strategy to segment latent fingerprints.

The rest of the paper is organized as follows: Section 2 reviews recent works in latent fingerprint segmentation while section 2.1 describes the contributions of this paper. Section 3 highlights our technical approach and presents a discussion on fractal dimension features used in our approach. Section 4 presents a brief review of extreme learning machine, weighted extreme learning machine (*WELM*), and ensemble learning and segmentation using *WELM*. The experimental results and performance evaluation of our proposed approach are discussed in section 5.3. Section 6 contains the conclusion and future work.

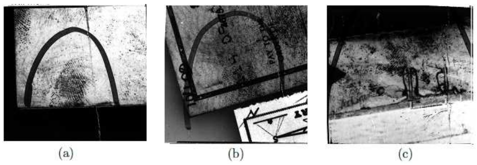

Figure 1. Sample latent fingerprints from NIST SD27 showing three different quality levels (a) good, (b) bad, and (c) ugly.

2. Related work

The goal of fingerprint segmentation is to decompose the input fingerprint image into fingerprint and non-fingerprint parts. Reliable feature extraction and matching require accurate segmentation [4].

A few recent studies have been done on latent fingerprint segmentation [4], [17], [16], [6], [2]. In [4], the authors used orientation tensor approach to extract the symmetric patterns of a fingerprint and removed the structured noise in background. They used local Fourier analysis method to estimate the local frequency in the latent fingerprint image and locate fingerprint region by considering valid frequency regions. They obtain candidate fingerprint (foreground) regions for each feature (orientation and frequency) and then localize the latent fingerprint regions using the intersection of those candidate fingerprint regions. Karimi et al. [6] estimated local frequency of the ridge/valley pattern based on ridge projection with varying orientations. They used the variance of frequency and amplitude of the ridge signal as features for the segmentation algorithm. They reported segmentation results for only two latent fingerprint images and provided no performance evaluation. Short et al. [16] proposed ridge template correlation method for latent segmentation. They generated an ideal ridge template and computed cross-correlation values to define the local fingerprint quality. They manually selected 6 different threshold values to assign a quality value to each fingerprint block. They neither provided the size and number for the ideal ridge template nor reported and evaluation criteria for the segmentation results. Zhang et al. [17] proposed an adaptive total variation (TV) model for latent segmentation. They adaptively determined the weight assigned to the fidelity term in the model based on the background noise level. They used this to remove the background noise in the latent fingerprint images. Arshad et al. [2] used K-means clustering to divide the latent fingerprint image into non-overlapping blocks and compute the standard deviation of each block. If the standard deviation of a block is greater than a predefined threshold, they consider it a foreground block otherwise, its is a background block. They perform morphological operations and apply mask to segment the latent fingerprint.

2.1. Contributions

Our approach differs significantly from the existing approaches because we perform the segmentation by extracting patches from the latent fingerprint image and classifying them into fingerprint and non-fingerprint patches. We assemble the fingerprint patches to build the fingerprint portion (segmented ROI) of the original image, without any post processing. Existing approaches for latent fingerprint segmentation rely on the analysis of the ridge frequency and orientation properties of the ridge valley patterns to determine the area within a latent fingerprint image that contains the fingerprint. Our approach performs classification of image patches extracted from the fingerprint image using fractal dimension features which are persistent geometric quantities. Moreover, as can be seen from Figure 18 which shows the results of training our model on NIST SD27 and testing on NIST SD27, WVU, and IIITD latent databases, the performance of our approach is invariant to the choice of database used for training the weighted extreme learning machine model.

3. Technical approach

Our approach involves partitioning a latent fingerprint image into 8×8 non overlapping blocks called patches and compute fractal dimension features from the patches. We use the features to train and test weighted extreme learning machine ensemble (*WELME*) classifier. The *WELME* classifies the patches into fingerprint and non-fingerprint classes. We use the fingerprint patches to reconstruct the latent fingerprint image and discard the non-fingerprint patches which contain the structured noise in the original latent fingerprint. The block diagram of our proposed approach is shown in Figure 2.

3.1. Fractal dimension

Fractal dimension is an index used to characterize texture patterns by quantifying their complexity as a ratio of the change in detail to the change in the scale used. It was defined by Mandelbrot [12] and was first used in texture analysis by Keller et al. [8]. Fractal dimension offers a quantitative way to describe and characterize the complexity of image texture composition [9].

We compute the fractal dimension of an image patch *P* using a variant of differential box-counting (DBC) algorithm [1, 15]. We consider *P* as a *3-D* spatial surface with *(x,y)* axis as the spatial coordinates and *z* axis for the gray level of the pixels. Using the same strategy as in *DBC*, we partition the $N \times N$ matrix representing *P* into non-overlapping $d \times d$ blocks where $d \in [1, N]$. Each block has a column of boxes of size $d \times d \times h$, where *h* is the height defined by the relationship $h = \frac{\mathcal{T}d}{N}$, where $\mathcal{T}$ is the total gray levels in *P*, and *d* is an integer. Let $\mathcal{T}_{min}$ and

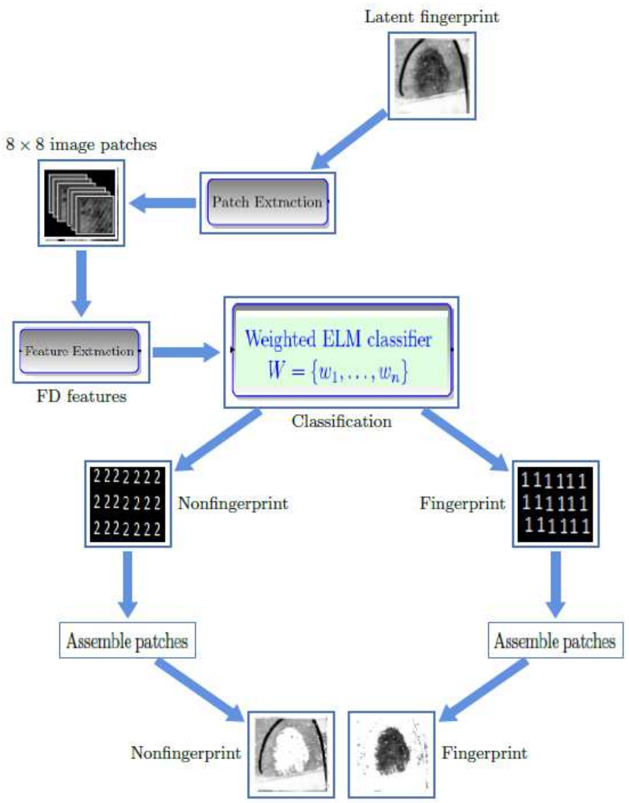

Figure 2. Proposed approach

$\mathcal{T}_{max}$ be the minimum and maximum gray levels in grid *(i, j)*, respectively. The number of boxes covering block *(i, j)* is given by:

$$n_d(i,j) = floor[\frac{\mathcal{T}_{max} - \mathcal{T}_{min}}{r}] + 1, \tag{1}$$

where $r = 2, \ldots, N-1$, is the scaling factor and for each block *r*=*d*. The number of boxes covering all $d \times d$ blocks is:

$$N_d = \sum_{i,j} n_d(i,j) \tag{2}$$

We compute the values N_d for all $d \in [1, N]$. The fractal dimension of each pixel in *P* is by given by the slope of a plot of the logarithm of the minimum box number as a function of the logarithm of the box size. We obtain a fractal dimension image patch P' represented by an $M \times N$ matrix whose entry *(i,j)* is the fractal dimension FD_{ij} of the pixel at *(i,j)* in *P*.

$$FD_P = \sum_{i=1,j=1}^{MN} FD_{ij}, \tag{3}$$

Figure 3 shows FD for sample fingerprint and non-fingerprint image patches and highlights the discriminative potential of FD for separating fingerprint from non-fingerprint image patches.

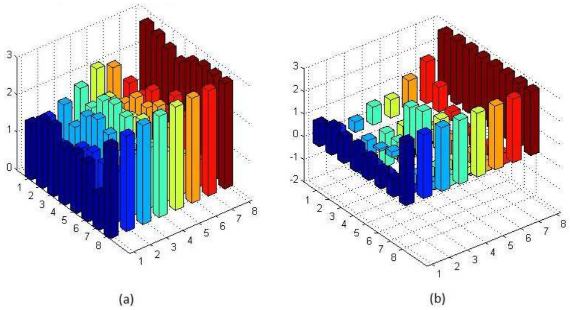

Figure 3. Fractal dimension for sample image patches : (a) fingerprint , (b) non-fingerprint

3.2. Fractal Dimension Features

We implemented a variant of the *DBC* algorithm to compute the following statistical features from the fractal dimension image P'.

3.3. Average Fractal Dimension

$$FD_{avg} = \frac{1}{MN} \sum_{i=1,j=1}^{MN} FD_{ij}, \tag{4}$$

3.4. Standard Deviation Fractal Dimension

The standard deviation of the gray levels in an image provides a degree of image dispersion and offers a quantitative description of variation in the intensity of the image plane. Therefore

$$FD_{std} = \frac{1}{MN} \sum_{i=1,j=1}^{MN} (FD_{ij} - FD_{avg}), \tag{5}$$

3.5. Fractal Dimension Spatial Frequency

This refers to the frequency of change per unit distance across an fractal dimension (FD) processed image. We compute it using the formula for (spatial domain) spatial frequency as in [11]. Given an $N \times N$ FD processed image patch P', let *G(x,y)* be the FD value of the pixel at location *(x,y)* in P'. The row frequency R_f and column frequency C_f are given by

$$R_f = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} [G(x,y) - G(x,y-1)]^2}, \tag{6}$$

$$C_f = \sqrt{\frac{1}{MN} \sum_{y=0}^{M-1} \sum_{x=1}^{N-1} [G(x,y) - G(x-1,y)]^2}, \tag{7}$$

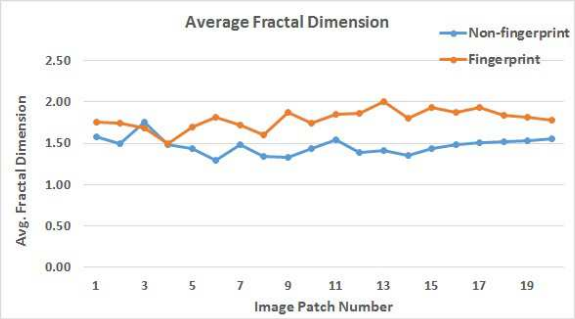

Figure 4. Average fractal dimension for some fingerprint and non-fingerprint 8×8 patches

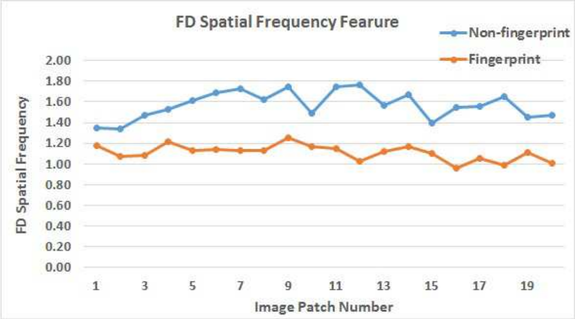

Figure 5. FD spatial frequency for sample fingerprint and non-fingerprint image patches

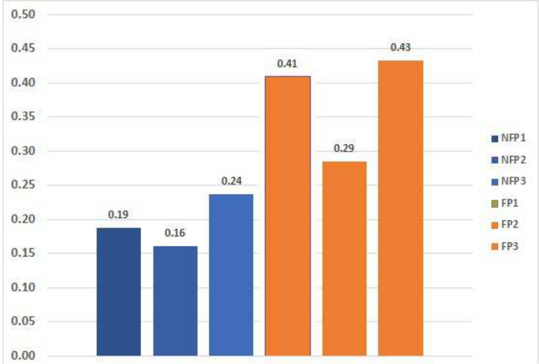

Figure 6. Lacunarity: FP1, FP2, FP2 are fingerprint patches; NFP1, NFP2, NFP3 are non-fingerprint patches

The FD spatial frequency FD_{sf} of P' is defined as

$$FD_{sf} = \sqrt{R_f^2 + C_f^2}, \tag{8}$$

From signal processing perspective, equations 6 and 7 favour high frequencies and yield values indicative of patches with fingerprint.

3.6. Lacunarity

Lacunarity is a second-order statistic that provides a measure of how patterns fill space. Patterns that have more or larger gaps have higher lacunarity. It also quantifies rotational invariance and heterogeneity. A spatial pattern that has a high lacunarity has a high variability of gaps in the pattern, and indicates a more heterogeneous texture [3]. Lacunarity (FD_{lac}) is defined in terms of the ratio of variance over mean value[1].

$$FD_{lac} = \frac{\frac{1}{MN}(\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} P(i,j)^2)}{\{\frac{1}{MN}\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} P(i,j)\}^2} - 1, \tag{9}$$

where *M* and *N* are the sizes of the fractal dimension image patch *P*. As can be seen from Figures 4, 5, 6, the FD features we used have the required discriminative potential for separating fingerprint from non-fingerprint patches.

4. Extreme learning machine

Extreme Learning Machine (ELM) is a machine learning algorithm for single-hidden layer feed-forward neural networks (SLFNs). ELM randomly chooses hidden nodes and analytically determines the output weights of SLFNs. The algorithm tends to provide good generalization performance at extremely fast learning speed. It can learn thousands of times faster than conventional popular learning algorithms for feed-forward neural networks, such as back-propagation (BP) algorithm [5]. The ELM theory states that contrary to conventional learning methods, the hidden node parameters can be generated before seeing the training data. Given *N* distinct samples (x_i, y_i), where $x_i = [x_{i1}, \cdots, x_{in}]^T \in \mathbb{R}^n$ and $y_i = [y_{i1}, \cdots, y_{im}]^T \in \mathbb{R}^m$, the output of ELM with *L* hidden nodes is given by

$$\sum_{i=1}^{L} \beta_i(w_i x_j + b_i), j \in [1, N], \tag{10}$$

where $w_i = [w_{i1}, \cdots, w_{im}]^T$ is the weight vector connecting the *i*th hidden node and the input nodes, $\beta_i = [\beta_{i1}, \cdots, \beta_{im}]^T$ is the weight vector that connects the *i*th hidden node and the output nodes, and b_i is the threshold (bias) of the *i*th hidden node.
ELM can solve the following learning problem

$$H\beta = T, \tag{11}$$

where $T = [y_1, \cdots, y_N]^T$ are target labels, and H is the hidden layer output matrix defined as $H = [h^T(x_1), \cdots, h^T(x_N)]^T$. The output weight matrix β is calculated as

$$\beta = H\dagger T, \tag{12}$$

where $H\dagger$ is the Moore-Penrose generalized inverse of matrix H [5].

4.1. Weighted extreme learning machine

Weighted ELM provides a solution to the classification accuracy paradox associated with imbalance class distribution while maintaining the advantages of the unweighted ELM. It allows for weights to be assigned to each training sample and generalizes to cost sensitive learning [18]. Specifically, we define an $N \times N$ diagonal weight *W* and associate it with every training sample x_i. The W assigned to a training sample x_i from a minority class is larger than that assigned to an x_k from a majority class. The weight assignment scheme assigns larger weight to the minority samples thereby strengthening the impact of the minority class and weakening the relative impact of the majority class. Generally, the goal of *ELM* is to minimize the training errors and maximize the marginal distance between the classes [18]:

$$Minimize : \|H\beta - T\|^2 \ and \ \|\beta\| \tag{13}$$

This is written mathematically as:

$$L_{P_{ELM}} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\xi_i\|^2, \tag{14}$$

$$Subject\ to : h(x_i)\beta = t_i^T - \xi_i^T, i = 1, \ldots, N \tag{15}$$

where $\xi_i = [\xi_{i,1}, \cdots, \xi_{i,m}]^T$ is the training error vector of *m* output nodes with respect to the training sample x_i, *C* is the regularization parameter that indicates the trade-off between the minimization of the training errors and maximization of the marginal distance between classes, $h(x_i)$ if the feature mapping vector in the hidden layer with respect to sample x_i, and β is the output weight vector connecting the hidden layer and the output layer.
For the *WELM*, the optimization problem becomes [18]:

$$L_{P_{ELM}} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{i=1}^{N}\|\xi_i\|^2, \tag{16}$$

$$Subject\ to : h(x_i)\beta = t_i^T - \xi_i^T, i = 1, \ldots, N \tag{17}$$

4.2. Weighted extreme learning machine ensemble (WELME) learning

The random initialization of both the bias and input weights of the weighted extreme learning machine causes fluctuations in model performance between runs. By using *WELME*, we are able to handle large patch image data sets and minimize the model's empirical risk. Inspired by [10] and [13], we implemented the *WELME* for fixed size block-by-block learning and segmentation. We analytically determined the optimal block size, which is also the number of neurons *L* in the hidden layer of the WELM to be 720 as shown in Figure 12.

We learn a set of hypothesis:

$$H = \{f(x_i, w_k), w_k \in WELME\} \tag{18}$$

where x_i is the *i-th* training sample and w_k is the *k-th WELM* in the *WELME*. We use a non-negative real-valued loss function $L(\hat{y}, y)$ to measure how different the prediction $\hat{y}$ of a w_k in the ensemble is from the true outcome *y*. We compute the empirical risk associated with each hypothesis $h(x)$ by averaging the loss function on the training and validation sets :

$$E_l(h) = \frac{1}{m}\sum_{i=1}^{m} L(h(x_i), y_i) \tag{19}$$

We select optimal set H^* of hypotheses that minimizes the total empirical risk (TER). In our experiments we let H^* be the top 5% hypotheses that have minimal risk $E_l(h)$ for final prediction.

4.3. Voting

The final prediction from the ensemble is obtained through voting. To obtain near optimal prediction from the ensemble, we require that Each $h(x) \in H^*$ meet the ensemble analytically determined confidence threshold ξ by satisfying:

$$c = \sum_{i=1}^{FN} W_p + \sum_{i=1}^{FP} W_n \leq \xi. \tag{20}$$

$$\xi = min\{R_v\}, v = 1, \ldots, K \tag{21}$$

where R_v is the total empirical risk of *WELM v*, *K* is the number of *WELM*s in the ensemble, *c* is the total misclassification penalty for each ensemble, *FN* and *FP* are the number of false negative and false positive predictions, respectively and the sum is over all the misclassification on the test data set by a given member of the ensemble. W_p and W_n are the weights assigned to fingerprint and non-fingerprint patches, respectively. The weights W_p and W_n account for the imbalance in the data set class distribution and were determined analytically. In our experiments, we set ξ to the value 0.16 which was obtained from experiments.

5. Experimental Results

We implemented our algorithms in Matlab R2014a running on Intel Core i7 CPU with 8GB RAM and 750GB hard drive.

5.1. Latent fingerprint databases

We tested our model on the following databases.
NIST SD27: This database was published by the National Institute of Standards and Technology in collaboration with

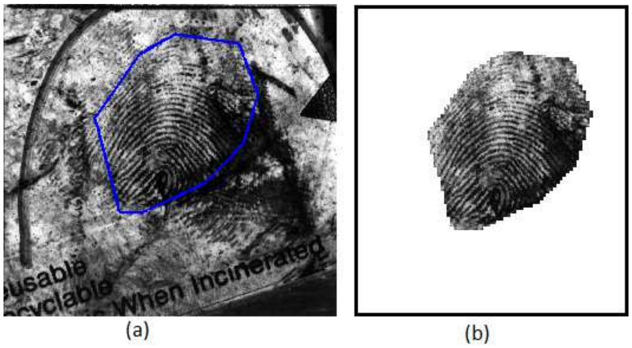

Figure 7. Building and validating the ground truth patch dataset

the FBI. It contains images of **258** latent crime scene fingerprints and their matching rolled tenprints. The images in the database are classified as good, bad or ugly based on the quality of the image. The latent prints and rolled prints are at **500** ppi.

WVU database: This database is jointly owned by West Virginia University and the FBI. It has **449** latent fingerprint images and matching **449** rolled fingerprints. All images in this database are at **1000** ppi.

IIITD database: The IIIT-D was published by the Image Analysis and Biometrics lab at the Indraprastha Institute of Information Technology, Delhi, India [14]. There are **150** latent fingerprints and **1,046** exemplar fingerprints. Some of the fingerprint images are at **500** ppi while others are at **1000** ppi.

5.2. Building the latent fingerprint patch dataset

Since there is no existing patch based latent fingerprint Ground-truth dataset, we created one and made it available on our website for others to use. (The reader can obtain the datasets by sending a request to bhanu@cris.ucr.edu or jezeobie@cs.ucr.edu). We built ground-truth dataset by extracting 8×8 image patches from GOOD, BAD, and UGLY latent fingerprint images from NIST SD27 database. For each latent fingerprint, we manually mark the region containing the fingerprint using a bounding *ROI* polygon. Our algorithm splits the latent fingerprint into 8×8 non-overlapping patches (blocks). A patch is labeled a fingerprint patch if it overlaps with the polygon and non-fingerprint otherwise. We say that a patch overlaps with the *ROI* polygon if it lies within the polygon or 25% of its pixels are inside the polygon or at the edge of the polygon. Figure 7(a) shows a latent fingerprint image with a bounding *ROI* polygon, while 7(b) shows the latent fingerprint constructed with the patches labeled as fingerprint.

5.3. Performance evaluation and metrics

We used the following metrics to evaluate the performance of our model.

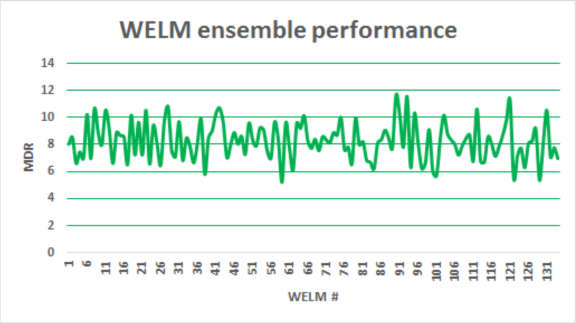

Figure 8. *WELME* performance: *MDR*

- **Missed Detection Rate (*MDR*):** This is the percentage of fingerprint patches classified as non-fingerprint patches. It is defined as.

$$MDR = \frac{FN}{TP + FN}, \tag{22}$$

  where *FN* is the number of false negatives and *TP* is the number of true positives.

- **False Detection Rate (*FDR*):** This is the percentage of non-fingerprint patches classified as fingerprint patches.It is defined as

$$FDR = \frac{FP}{TN + FP} \tag{23}$$

  where *FP* is the number of false positives and *TN* is the number of true negatives.

- **Total Empirical Error (*TER*):** The *TER* is defined as:

$$TER = \sum_{i=1}^{FN} W_p + \sum_{i=1}^{FP} W_n. \tag{24}$$

  where W_p and W_n are as defined in section 4.3. The *TER* conveys the balance between *MDR* and *FDR*.

- **G-mean:** The square root of the positive class accuracy × negative class accuracy [18].

$$G_{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{25}$$

  It gives a good indication of the effectiveness of the model.

5.4. Model parameter selection

We ran experiments to select optimal weight vector (*W*) for the classes, *L2* penalty constant *C*, and number of nodes (*L*) in the hidden layer of the *WELM* network. The results of the experiments are depicted in Figures 12, and 13. The

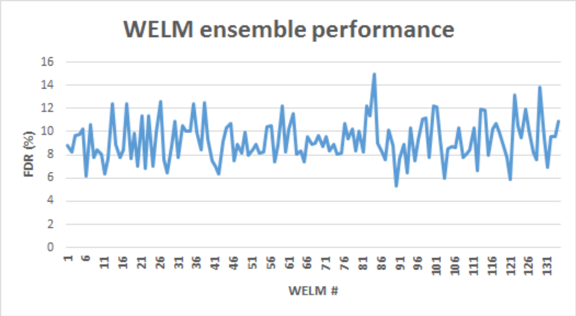

Figure 9. *WELME* performance: *FDR*

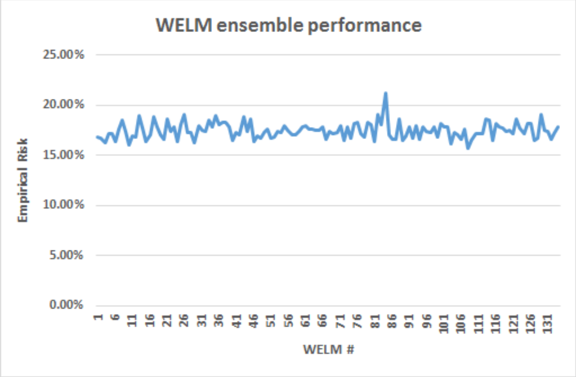

Figure 10. *WELME* performance: *Empirical risk*

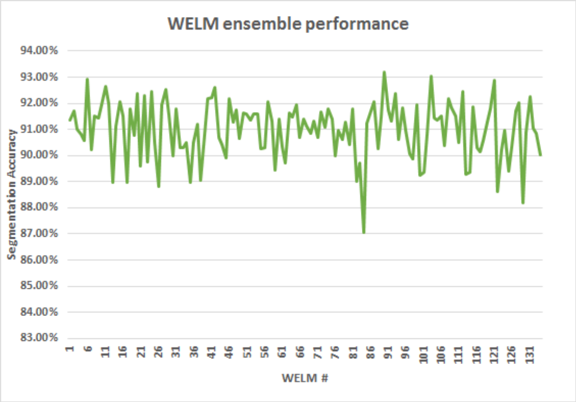

Figure 11. *WELME* performance: *Segmentation accuracy*

optimal values are $C = 10^{5.5}$, $L = 720$, $W_n = 3.75$ and $W_p = 7.15$ for good and bad quality images. For ugly quality images, $W_n = 0.001$ and $W_p = 0.093$. Figures 8, 9, 10, and 11 show the performance of the 134 WELMs in the WELME in terms of MDR, FDR, empirical risk and segmentation accuracy, respectively.

5.5. Confusion matrix

Training was done with 96,000 patches from images from the NIST SD27 database. The test dataset consists of 9,600 patches from the NIST SD27, WVU and IIITD latent databases. The confusion matric in Figure 14 shows the TP, TN, FP, and FN results and a segmentation accuracy 91.43%

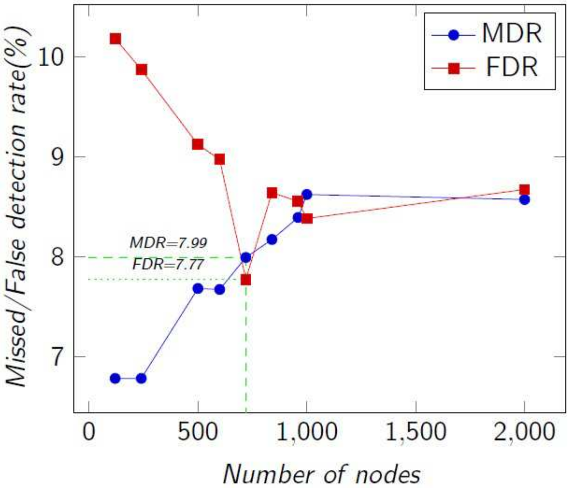

Figure 12. Hyper parameter selection: accuracy in the (L) subspace

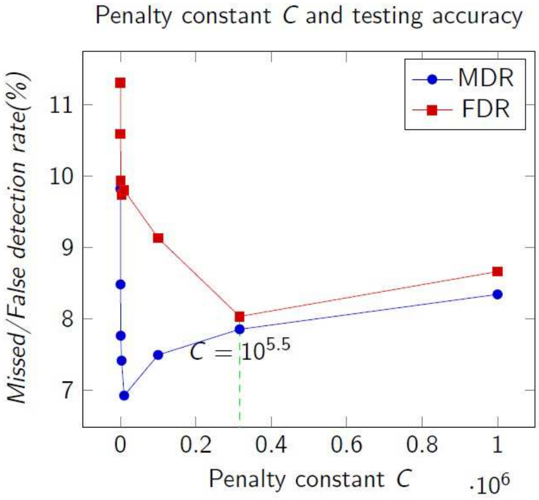

Figure 13. Hyper parameter selection: accuracy in the (C) subspace

		Predicted Patch Class	
		Fingerprint	Non-Fingerprint
Actual Patch Class	Fingerprint	**2061**	**643**
	Non-Fingerprint	**180**	**6716**

Figure 14. Confusion matrix

5.6. Segmentation results

Figures 15, 16, 17 show the segmentation results of our proposed method on sample good, bad and ugly quality images from the NIST SD27 database. The original latent fingerprint images are (a), (c), (e), while (b), (d), (f) are the segmented fingerprints constructed using patches classified as fingerprint, without any post classification processing.

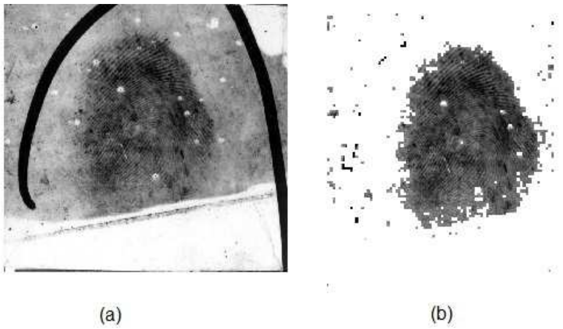

Figure 15. Good Latent fingerprint image and segmentation result

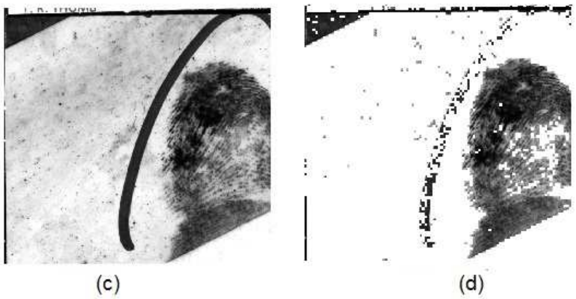

Figure 16. Bad Latent Fingerprint Image and segmentation result.

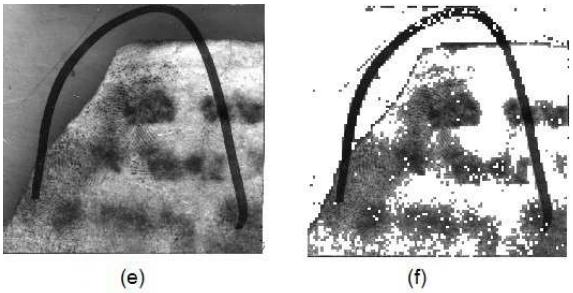

Figure 17. Ugly Latent Fingerprint Image and segmentation result.

Figure 18 shows the performance of our approach on different latent fingerprint databases.

5.7. Performance comparison with other algorithms

Figure 19 highlights the superior performance of our segmentation approach on the good, bad and ugly quality latent fingerprints from NIST SD27 compared to the results in [2], while Figure 20 shows that our approach performs better than existing approaches with respect to total empirical risk minimization and segmentation accuracy. Taking both MDR and FDR into consideration, our approach is 43.72% better than that of Zhang et al. [17] on the NIST SD27 database and 10.46% better that that of Arshad et al. [2] on the NIST SD27 database. On the WVU database, our approach is about 2 times better than that of Choi et al. [4].

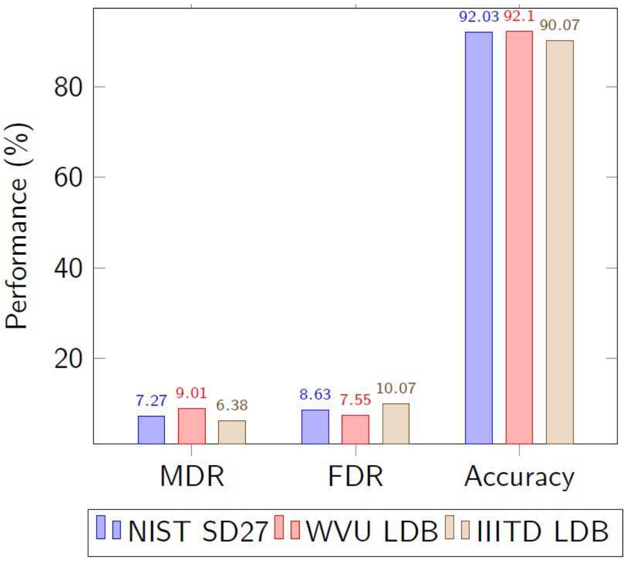

Figure 18. Segmentation reliability in different databases for good quality images

	Our approach			Arshad et al.		
	MDR	FDR	Average	MDR	FDR	Average
Good	7.27%	8.63%	7.95%	1.80%	17.85%	9.83%
Bad	12.77%	7.72%	10.24%	4.75%	26.28%	15.52%
Ugly	7.62%	39.75%	23.68%	22.50%	34.05%	28.28%

Figure 19. NIST SD27: Segmentation results in terms of MDR and FDR: Good, Bad, Ugly

Author	Approach	**Database**	**MDR %**	**FDR %**	AVERAGE
Choi et al.	Ridge orientation and frequency computation	NIST SD27 WVU LDB	14.78 40.88	47.99 5.63	31.38 23.26
Zhang et al.	Adaptive Total Variation model	NIST SD27	14.10	26.13	20.12
Arshad et al.	K-means clustering	NIST SD27	4.77	26.06	15.42
Our approach	Fractal Dimension & Weighted ELM	NIST SD27 (Good, Bad, Ugly) WVU LDB (Good,Bad,Ugly) IIITD LDB (Good)	9.22 15.54 6.38	**18.7** 9.65 10.07	**13.96** **12.60** 8.23

Figure 20. Performance comparison of segmentation approaches

6. Conclusions and future work

We have proposed a new technique based on fractal dimension and weighted extreme learning machine (*WELM*) for latent fingerprint segmentation using image patches. Experimental results using latent fingerprint images from the NIST SD27, WVU and IIITD latent fingerprint databases showed the promise of our method. Our future work involves incorporating additional features to improve classification accuracy of the fingerprint patches from bad and ugly latent fingerprints.

References

[1] O. Al-Kadi and D. Watson. Texture analysis of aggressive and nonaggressive lung tumor ce ct images. *Biomedical Engineering, IEEE Transactions on*, 55(7):1822–1830, July 2008.

[2] I. Arshad, G. Raja, and A. Khan. Latent fingerprints segmentation: Feasibility of using clustering-based automated approach. *Arabian Journal for Science and Engineering*, 39(11):7933–7944, 2014.

[3] M. Barros Filho and F. Sobreira. Accuracy of lacunarity algorithms in texture classification of high spatial resolution images from urban areas. In *XXI congress of international society of photogrammetry and remote sensing*, 2008.

[4] H. Choi, a. I. B. M. Boaventura, and A. Jain. Automatic segmentation of latent fingerprints. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 303–310, Sept 2012.

[5] G. Huang, Q. Zhu, and C. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(13):489 – 501, 2006.

[6] S. Karimi-Ashtiani and C.-C. Kuo. A robust technique for latent fingerprint image segmentation and enhancement. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1492–1495, Oct 2008.

[7] D. Kaye, T. Busey, M. Gische, G. LaPorte, C. Aitken, S. Ballou, L. B. ..., and K. Wertheim. Latent print examination and human factors: Improving the practice through a systems approach. *NIST Interagency/Internal Report (NISTIR) - 7842*, Jan 2012.

[8] J. Keller, S. Chen, and R. Crownover. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing*, 45(2):150 – 166, 1989.

[9] A. D. K. T. Lam and Q. Li. Fractal analysis and multifractal spectra for the images. In *Computer Communication Control and Automation (3CA), 2010 International Symposium on*, volume 2, pages 530–533, May 2010.

[10] Y. Lan, Y. C. Soh, and G.-B. Huang. Ensemble of online sequential extreme learning machine. *Neurocomputing*, 72(1315):3391 – 3395, 2009. Hybrid Learning Machines (HAIS 2007) / Recent Developments in Natural Computation (ICNC 2007).

[11] S. Li, J. T. Kwok, and Y. Wang. Combination of images with diverse focuses using the spatial frequency. *Information Fusion*, 2(3):169 – 176, 2001.

[12] B. Mandelbrot. *The Fractal Geometry of Nature*. Einaudi paperbacks. Henry Holt and Company, 1983.

[13] B. Mirza, Z. Lin, and K.-A. Toh. Weighted online sequential extreme learning machine for class imbalance learning. *Neural Processing Letters*, 38(3):465–486, 2013.

[14] A. Sankaran, M. Vatsa, and R. Singh. Hierarchical fusion for matching simultaneous latent fingerprint. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 377–382, Sept 2012.

[15] N. Sarkar and B. B. Chaudhuri. An efficient differential box-counting approach to compute fractal dimension of image. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(1):115–120, Jan 1994.

[16] N. Short, M. Hsiao, A. Abbott, and E. Fox. Latent fingerprint segmentation using ridge template correlation. In *Imaging for Crime Detection and Prevention 2011 (ICDP 2011), 4th International Conference on*, pages 1–6, Nov 2011.

[17] J. Zhang, R. Lai, and C.-C. Kuo. Latent fingerprint segmentation with adaptive total variation model. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 189–195, March 2012.

[18] W. Zong, G.-B. Huang, and Y. Chen. Weighted extreme learning machine for imbalance learning. *Neurocomputing*, 101:229–242, 2013.