



Learning Representations for Matching Fingerprint Variants

by

Anush Sankaran

Under the supervision of

Dr. Mayank Vatsa

Dr. Richa Singh

Indraprastha Institute of Information Technology Delhi
February, 2017

©Anush Sankaran, 2017.



Learning Representations for Matching Fingerprint Variants

by

Anush Sankaran

Submitted

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

to the

Indraprastha Institute of Information Technology Delhi

February, 2017

Certificate

This is to certify that the thesis titled “**Learning Representations for Fingerprint Variants**” being submitted by **Anush Sankaran** to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

February, 2017

Dr. Mayank Vatsa

February, 2017

Dr. Richa Singh

Indraprastha Institute of Information Technology Delhi

New Delhi 110020

Acknowledgment

A PhD journey that started six years back culminates with this thesis, cumulating my learnings and maturity over this journey. This thesis marks both the completion of an important phase in my life and also the beginning of my career. At this juncture, I want to show my gratitude to those who have supported me through this special journey.

Foremost, I want to express my sincere thanks and heartfelt gratitude to my advisors, **Dr. Mayank Vatsa** and **Dr. Richa Singh** for their relentless support, vision, and exemplary guidance. Their caring nature and tailor-made guidance gave me hope during the tough times of this journey. It is surely not an overstatement that both the thesis and I would not have obtained this stage, if not for their inspirational mentorship.

This journey would not have been possible, if not for the generous financial support from **Tata Consultancy Services (TCS)** under their research scholar funding program. Also, thanks to the **Department of Science and Technology (DST)**, Government of India and **NVIDIA** for providing funding and equipments necessary for research.

My profound thanks to the university director, **Prof. Pankaj Jalote** for inspiring me to take up this journey and providing a competitive research environment in the university. I would like to thank **Dr. Afzel Noore**, **Dr. Keith Morris**, and **Dr. Ajay Kumar** for having provided me with internship opportunities. These collaborative internships nourished me with international exposure and diversified my perspective towards research and life. Further, their kindness and patience was a humbling experience, which gave me strength and kept me going. I thank **Dr. Angshul Majumdar** for his valuable technical inputs while addressing some challenging problems in my thesis. My yearly evaluation committee members, **Dr. Debajyoti Bera** and **Dr. A. V. Subramanyam**, constantly provided constructive feedback and steered my thesis in a positive direction. I would also like to thank all my teachers for having selflessly transferred their knowledge and equipping me with necessary technical and non-technical skills.

The best part of this special journey had been the company of my lab-mates, **Himanshu**, **Samarth**, and **Tejas**. Their equal mixture of fun and hard work, made the lab life very memorable and exciting. Short technical discussions with them would solve weeks worth of confusion and bring more clarity to work. I acknowledge my collaborators and co-authors, **Naman**, **Daksha**,

and **Gaurav** for sharing their talent and time, when needed. Thanks to **Tarun, Aayush, Apoorva, Aakarsh,** and **Prateekshit** for making me appreciate the fact that teaching is one of the best forms of learning.

Friends outside of work have been the source of celebration and liveliness. **Monika**, who is as special to me as my thesis, shared the entire adventurous journey with me and a trivial thanks is not enough to value her love and support. A singular mention to **Garima, Paridhi, Siddarth, Niharika,** and **Komal** for making me embrace failures in equal proportion to success.

Finally, with utmost gratitude, I would like to thank my parents, **Sankaran** and **Raji**, my brother **Dinesh** and sister-in-law **Sowmya** for having faith in me. Their love, prayers, and sacrifice had been the biggest support through this adventurous journey.

Learning Representations for Matching Fingerprint Variants

by

Anush Sankaran

Abstract

Fingerprint recognition has evolved over the decades, providing innumerable applications for improving the modern day security. Based on the method of capture, fingerprints can be classified into four variants: inked, live-scan, latent, and fingerphoto. Extensive research has been undertaken for inked and live-scanned fingerprints. However, research on latent fingerprints and fingerphoto matching is still in nascent stages. These two capture methodologies are semi-controlled or uncontrolled which pose significant variations in the feature space and therefore warrant further exploration. The key research challenges involved in building an automated system for latent fingerprint and fingerphoto matching are as follows: (i) lack of publicly available large scale datasets with diverse variations to motivate reproducible research, (ii) segmentation of foreground regions from the complex background surface, and (iii) lack of robust feature models to represent the noisy and partial finger ridge information. Currently, there are limited end-to-end automated systems for latent fingerprint and fingerphoto matching. This thesis primarily focuses in contributing towards building a completely automated “lights-out” matching system for these two fingerprint variants. There are four contributions ranging from creating large databases to designing algorithms for segmentation and feature extraction for these two fingerprint variants.

First, we create two benchmark datasets with diverse acquisition methods: (i) Multi-sensor Optical and Latent Fingerprint (MOLF) dataset containing 19,200 fingerprint images with large intra-class and capture variations and (ii) IIIT-D SmartPhone FingerPhoto Dataset version 2 (ISPF2-v2) containing 16,800 images from 300 classes captured under different environmental setup. The second contribution is designing an automated latent fingerprint segmentation algorithm that segments the fingerprint regions from background by distinguishing between ridge and non-ridge patterns. Latent fingerprint segmentation is usually affected by the *texture* of the surface and smudges are introduced during lifting. The proposed learning-based algorithm is generalizable and can accommodate for unseen texture noises. Further, a novel Spectral Image Validation and Verification based metric is proposed to measure the effect of the segmentation algorithm. Third, a minutiae extraction algorithm is proposed as a major contribution towards the “lights-out” latent fingerprint matching. A novel group (or class) sparsity based $\ell_{2,1}$ regularization method is proposed to improve the unsupervised features learnt using stacked autoencoders and Restricted Boltzmann Machines. Latent fingerprint minutiae extraction is then posed as a binary classification problem to classify patches as *minutia* or *non-minutia*. To the best of our knowledge, this is the first algorithm in literature for automated minutia extraction from latent fingerprints. The fourth contribution is towards fingerphoto recognition, in which a novel end-to-end fingerphoto matching algorithm is proposed that is invariant to the environmental factors such as background noise, illumination variation, and camera resolution. The ridge-valley pattern present in a fingerphoto is not as distinct as

a fingerprint image, thus making minutia extraction highly noisy. The matching pipeline consists of a segmentation technique to extract the fingerphoto region of interest from varying background, followed by enhancement to neutralize the illumination imbalance and increase the ridge valley contrast. For feature extraction, a deep scattering network based representation is introduced. The resultant fingerphoto features are robust and invariant to environmental variations. By addressing these challenging problems, this thesis improves the understanding and performance of automated matching systems for forensic latent fingerprints and fingerphoto images.

Table of Contents

1	Introduction	1
1.1	Latent Fingerprint Recognition	5
1.2	Smartphone Fingerphoto Recognition	10
1.3	Research Contributions	13
2	Literature Survey	17
2.1	ACE-V Method For Latent Fingerprint Comparison	18
2.2	Study of Human Performance for Latent Fingerprint Comparison	19
2.3	Automated Latent Fingerprint Recognition System	23
2.3.1	Latent Fingerprint Segmentation	23
2.3.2	Latent Fingerprint Quality Assessment and Enhancement	28
2.3.3	Latent Fingerprint Feature Extraction	32
2.3.4	Latent Fingerprint Comparison	38
2.4	Smartphone Fingerphoto Matching	42
3	Database Collection	45
3.1	Introduction	45
3.2	Multi-sensor Optical and Latent Fingerprint Database	50
3.2.1	Fingerprint Data Collected with Optical Sensors	51
3.2.2	Latent Fingerprint Collection and Annotation	52
3.2.3	Latent Fingerprint Annotation	53
3.2.4	Availability of Database	54
3.3	Research Applications of the Database	55

3.4	Experimental Evaluation for Baseline Results	56
3.4.1	Quality Analysis	56
3.4.2	Sensor Interoperability Analysis	57
3.4.3	Latent Fingerprint Matching	59
3.5	IIITD Smartphone Fingerphoto Database v2	64
4	Adaptive Latent Fingerprint Segmentation	69
4.1	Introduction	69
4.2	Proposed Segmentation Algorithm	72
4.2.1	Feature Extraction	72
4.2.2	Feature Selection	83
4.2.3	Classification using RDF	84
4.2.4	Representing Segmented Latent Prints	86
4.3	Evaluation Metrics for Latent Fingerprint Segmentation	87
4.4	Experimental Results	89
4.4.1	Datasets	89
4.4.2	Experimental Protocol	90
4.4.3	Importance of Saliency	91
4.4.4	Segmentation Performance	93
4.4.5	Comparison with Existing Feature Selection Algorithms	95
4.4.6	Comparison with Existing Latent Print Segmentation Algorithm	96
4.4.7	Matching Performance	97
4.4.8	Performance Evaluation using Latent Fingerprint Identification System with Very Large Gallery	100
4.5	Summary	101
5	Latent Fingerprint Minutiae Extraction	103
5.1	Introduction	103
5.2	Regularization in Deep Learning Algorithms	105
5.3	GSAE: Group Sparse AutoEncoders	109
5.3.1	Solution using Majorization-Minimization	110

5.4	Class Sparsity based Restricted Boltzmann Machine	114
5.4.1	Restricted Boltzmann Machine (RBM)	114
5.4.2	Class Sparsity Signature in RBM	116
5.4.3	Constructing Deep Networks using cssRBMs	118
5.4.4	cssDBM and cssDBN with Dropout and Dropconnect	120
5.5	Latent Fingerprint Minutia Extraction	121
5.6	Experimental Protocol and Performance Analysis	124
5.6.1	Fingerprint Datasets	124
5.6.2	Evaluation Metrics	126
5.6.3	Latent Fingerprint Minutiae Extraction Performance	126
5.7	Latent Fingerprint Recognition Performance	130
5.8	Summary	131
6	Smartphone Fingerphoto Recognition	133
6.1	Proposed Fingerphoto Matching Algorithm	136
6.1.1	Fingerphoto Segmentation	136
6.1.2	Fingerphoto Enhancement	139
6.1.3	ScatNet based Feature Representation	139
6.1.4	Feature Matching	142
6.2	IIITD SmartPhone Fingerphoto Database v2	143
6.3	Results and Analysis	146
6.3.1	Experimental Setting	146
6.3.2	Comparison Algorithms	148
6.3.3	Implementation Details	148
6.3.4	Proposed Pipeline Performance Analysis	149
6.3.5	Minitiae and Non-minutiae Matching Pipeline	150
6.3.6	Existing Algorithms	151
6.3.7	Effectiveness of Enhancement	151
6.4	Summary	152

7	Conclusion and Future Research	157
7.1	Conclusion	158
7.2	Future Research	159
A	Results on Standard Image Datasets	161
A.1	Standard Image Datasets	161
A.2	Evaluation Metrics	162
A.3	Results Using GSAE	163
A.4	Results using cssDBM and cssDBN	166

List of Figures

1-1	Examples of imprints common on ceramic pots from various periods (discovered by Martin Hloek) [1].	2
1-2	Sample images showing high intra-class variation introduced during fingerprint capture. The right index finger is captured from two different subjects using different capture methods. (a) inked fingerprint, (b)-(d) live scan fingerprints: (b) CrossMatch sensor, (c) Secugen Hamster IV sensor, (d) Lumidigm multi-spectral sensor, (e) latent fingerprint lifted using black powder dusting method, and (f) smartphone camera captured fingerphoto image.	3
1-3	Sample latent fingerprint images from the Evaluation of Latent Fingerprint Technologies, Extend Feature Set (ELFT-EFS) database [2].	4
1-4	An example illustrating the challenge of latent fingerprints. A sample unknown latent fingerprint along with a pool of known exemplar prints (a)-(e) is shown. Of these five latent fingerprints, only two are mated with the exemplar fingerprint, while the remaining three are false matches.	5
1-5	A stepwise, semi-automated procedure for analyzing latent fingerprints obtained from a crime scene. The dotted cylinders represent the human intervention in the latent fingerprint identification process.	7
1-6	Challenges in latent fingerprint matching compared with their corresponding full fingerprints.	9

1-7	Sample images illustrating the variations in the ridge patterns and the noise levels between different types of fingerprint capture mechanisms: (a) live-scan fingerprints image captured using Secugen Hamster IV sensor, (b) contactless finger images captured using low resolution webcam under constrained environment [3], and (c) contactless fingerphoto images captured using three different smartphone cameras - Micromax Canvas Knight, Apple iPhone, and OnePlus One, respectively.	11
1-8	Examples showing minutiae extraction using VeriFinger Software Development Kit (SDK) (one of the state-of-the-art fingerprint matching systems) on fingerphoto images captured using smart phone camera.	12
2-1	Analysis, Comparison, Evaluation, and Verification (ACE-V) methodology for manual matching of latent fingerprint.	17
2-2	The overall schema of an automated latent fingerprint matching system.	23
2-3	Sample latent fingerprints demonstrating different ways of segmenting a latent fingerprint image. (a) original latent fingerprint image, (b) segmentation of the outline of the entire latent fingerprint, and (c) segmentation of the outline of latent fingerprint (yellow full lines) and marking the structured noise (blue dashed lines) and smudgy region (red dotted lines) overlapping with the print.	24
2-4	Example latent fingerprint images from National Institute of Standards and Technology, Special Database 27 (NIST SD-27) showcasing two specific challenges in latent fingerprint segmentation. (a) Overlapped fingerprints result in overlapped ridge information (multiple touches in the same location) making it difficult to determine the ridge flow of either of the fingerprints, and (b) the presence of structured noise in latent fingerprint background that often resemble ridge like patterns.	25

2-5	Sample latent fingerprints from the NIST SD-27 fingerprint database [4] showing segmentation results. (a) Original latent fingerprint images, (b) manually segmented output with just a bounding box around the fingerprint region, (c) manually segmented output with exact boundary around the fingerprint region, (d) manually segmented output with only the useful ridge information rejecting all the smudgy and noisy (non-informative) regions, (e) segmented output from <i>nfseg</i> module of NIST Biometric Image Software (NBIS) [5], and (f) segmented output from Choi et al. algorithm [6] (implemented by the authors).	26
2-6	Sample latent fingerprints from NIST SD-27 enhanced using VeriFinger SDK 6.0	29
2-7	Different types of features extracted from fingerprint. (a) shows level 1 and level 2 features and (b) shows level 3 features.	34
2-8	Sample latent fingerprints from NIST SD-27 showing spurious minutiae extracted by (a) NBIS and (b) VeriFinger 6.0 SDK.	34
2-9	Sample images showing high intra-class variation in latent fingerprints captured from the same finger. Images obtained from the ELFT-EFS public challenge dataset [2].	37
3-1	Sample fingerprints captured of a subject representing capture variations in the MOLF database: (a) 500Pixels Per Inch (PPI) fingerprint set from Secugen live scan sensor, (b) multi-spectral fingerprint set from Lumidigm live scan sensor, (c) slap fingerprint set from CrossMatch L-Scan Patrol live scan sensor, (d) latent fingerprint set, (e) simultaneous latent fingerprint set of subject's right hand, and (f) simultaneous latent fingerprint set of subject's left hand.	49
3-2	(a) The latent fingerprint capture setup utilizing a 1/2" Complementary metal-oxide-semiconductor (CMOS) sensor with a 8mm focal length Closed-circuit television (CCTV) lens mounted on a Manfrotto magic arm that yields an image of size 3840 × 2748 and (b) a sample screen shot of the Graphical User Interface (GUI) based software tool developed for fingerprint feature annotation.	52

3-3 [NIST Fingerprint Image Quality \(NFIQ\)](#) quality score distribution of (a) *DB1* (Lumidigm) images, (b) *DB2* (Secugen) images, (c) *DB3_A* (CrossMatch) images, and (d) *DB4* (latent) images. In NFIQ measure, 1 denotes the best quality score while 5 denotes the worst. 57

3-4 Sample images showing quality variations across the three sensors (a) Secugen Hamster IV, (b) CrossMatch L-Scan Patrol, and (c) Lumidigm Venus. It can be observed that some of the images captured using Secugen Hamster IV has a poor capture quality because of its unconstrained capture mode. 58

3-5 [Cumulative Match Characteristics \(CMC\)](#) curves using NBIS for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability. 61

3-6 [CMC](#) curves using VeriFinger for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability 61

3-7 ROC curves using NBIS for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability. 62

3-8 ROC curves using VeriFinger for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability 62

3-9 [CMC](#) curves for experiments II(a), II(b) and III(b). The results are computed with (a) manually marked minutiae matched using Minutiae Cylinder Code, (b) manually marked features matched using BOZORTH3, (c) [NBIS](#), and (d) VeriFinger. . . 63

3-10 The environment and the mount setup used to collect fingerphoto images in the proposed [IIIT-D SmartPhone FingerPhoto Dataset version-2 \(ISPF-D-v2\)](#). 65

3-11 Sample images showing various challenges addressed in IIITD SmartPhone Fingerphoto Database v2. (a)-(b) illumination variation with white background, (c)-(d) background variation, and (e) camera resolution variation. Multiple samples showing the intra-class variations and noise present in the database. 66

4-1	(a) Sample latent fingerprint images from the NIST SD-27 database [4] demonstrating the effect of background information on ridge information and (b) latent fingerprint samples illustrating the problem of segmentation.	69
4-2	Sample latent fingerprints illustrating the distinct nature of foreground ridge patterns and background. (a) inked fingerprints, (b) live-scan fingerprints, and (c) latent fingerprints.	70
4-3	Illustrating the steps involved in the proposed Random Decision Forest (RDF) based latent fingerprint segmentation algorithm. A composite set of features is extracted from every local block and a random decision forest based binary classifier is used to classify the foreground regions from background.	71
4-4	(a) A sample image from the NIST SD-27 database with corresponding saliency-based (b) intensity feature (f_1), and (c) orientation feature (f_2).	74
4-5	Sample local blocks from the NIST SD-27 database with (a) foreground ridge blocks and (b) noisy background blocks. Varying image intensity patterns can be observed between the foreground and the background blocks.	74
4-6	Sample local square blocks from the NIST SD-27 database with (a) dominant orthogonal orientation observed in a foreground ridge block, (b) no dominant direction found in a noisy background block.	76
4-7	Sample images from the NIST SD-27 database showing noise in latent fingerprint images. The yellow dotted lines are the actual fingerprint regions while the red full lines are ridge like noisy regions in the background.	80
4-8	An example showing the post-processing performed on the classification output to arrive at the final segmentation output. (a) original input latent fingerprint, (b) classifier output of segmentation, (c) final segmentation boundary obtained after post-processing, and (d) an elliptical window fitted over the segmented region. . . .	86
4-9	Illustrating the segmentation result of the proposed algorithm using two sample images from NIST SD-27 by using (a) only saliency features (f_1, f_2), (b) all other features except saliency ($f_3 - f_{23}$), and (c) all features (f_1, f_{23}).	92
4-10	Comparing the performance of the proposed segmentation algorithm with the algorithm recently proposed by Zhang et al. [7], on the NIST SD-27 dataset.	97

4-11	CMC curves showing the average matching performance of unsegmented and segmented images on the NIST SD-27 database.	99
4-12	CMC curves showing the average matching performance of unsegmented and segmented images on the IIITD Combined Latent Fingerprint (IIITD-CLF) database. .	100
5-1	Sample latent fingerprints from NIST SD-27 database [4] showing spurious minutiae extracted by (a) NBIS and (b) VeriFinger 6.0 SDK	104
5-2	(a) High quality fingerprint patches illustrating the difference in the ridge structure between minutia and non-minutia patches, (b) Local patches from latent fingerprints illustrating the lack of well defined structures and noisy ridge patterns. . . .	104
5-3	An overview of different kinds of regularization techniques used in the literature for an network based deep architecture. Note that the number of nodes in each hidden later is varying and may be dense or sparse depending on the application, data characteristics, and architecture.	108
5-4	Block diagram to explain the various stages in the proposed algorithm. I: Pre-training stage where the group sparse deep autoencoder is learnt from labelled high quality fingerprints, II: Fine-tuning stage where the feature learner and classifier are trained with labelled latent fingerprints, and III: Testing stage in which local patches from unknown latent fingerprint are classified as minutia and non-minutia patches.	122
5-5	Sample example latent fingerprints from NIST SD-27 database showing correct results of the proposed algorithm. Red dots denote manually annotated minutiae and green patches represent the minutia patches predicted using the Group Sparse Autoencoder (GSAE) algorithm.	128
5-6	Sample latent fingerprints from NIST SD-27 database showing some incorrect predictions obtained using the proposed algorithm. Red dots represent the annotated minutiae, green patches represent the correct minutiae patches predicted using the GSAE algorithm and blue patches show the incorrect ones (false positive). It can be observed that no genuine minutiae is rejected by the proposed algorithm.	129

5-7 Recognition performance on the [Multi-sensor Optical and Latent Fingerprint \(MOLF\)](#) dataset using a latent fingerprint identification system. The in-built minutiae extractor is compared against the proposed minutiae extraction algorithms including [GSAE](#), [Class Sparsity Specific Deep Boltzmann Machine \(cssDBM\)](#), and [Class Sparsity Specific Deep Belief Network \(cssDBN\)](#), while the matching algorithm is kept constant. 130

6-1 Illustrating the different steps involved in the proposed fingerphoto verification pipeline. 135

6-2 Example images showing the segmentation results of using only skin color based features or only saliency based features. 137

6-3 Illustration of the segmentation of algorithm combining the saliency based and skin color based segmentation maps. 137

6-4 A segmented fingerphoto image and its enhanced image using the proposed algorithm, along with the output of applying a band pass filter, removing most of the high-frequency components. 141

6-5 ScatNet feature representation of the enhanced image where R_1 to R_{209} represent the responses obtained from a wavelet filter bank of 209 filters, which is up to a depth of two in the scattering network. 141

6-6 Sample images showing various challenges addressed in [ISPF2-v2](#). (a)-(b) illumination variation with white background, (c)-(d) background variation, and (e) camera resolution variation. Multiple samples showing the intra-class variations and noise present in the database. 145

6-7 Analyzing the effect of enhancement on matching fingerphoto images with illumination variations. 152

6-8 Analyzing the effect of enhancement on matching fingerphoto images with background variations. 153

6-9 Analyzing the effect of enhancement on matching fingerphoto images with background and illumination variations. 153

6-10	ROC curves illustrating the effect of the proposed enhancement algorithm on the verification performance of different fingerphoto recognition algorithms with background and illumination variations (Exp. 2): (a) without enhancement and (b) with enhancement.	154
6-11	ROC curves illustrating the effect of the proposed enhancement algorithm on the verification performance of different fingerphoto recognition algorithms with resolution variations (Exp. 2): (a) without enhancement and (b) with enhancement. . .	155
A-1	Sample images from the CIFAR-10 and MNIST databases without any preprocessing.	163
A-2	Visualization of the features learnt in the first hidden layer of the autoencoder on MNIST dataset with (a) standard autoencoder using only KL-divergence based sparsity, (b) proposed GSAE learning algorithm.	164
A-3	Sample representation showing the features learnt in MNIST database using (a) standard RBM and (b) proposed cssRBM.	168

List of Tables

2.1	Consequences and implications of human performance in matching latent fingerprints. It can be clearly shown that the human examiners can be inconsistent by the introduction of bias. Also, some contradictory results can be obtained from experiments performed in different setup, further adding to the inconsistency in human performance.	20
2.2	Review of existing latent fingerprint segmentation techniques.	25
2.3	A comparison of different descriptors used in the literature to describe latent fingerprint ridge structure.	35
2.4	Steps involved in latent fingerprint analysis, the features used in literature for the individual steps and corresponding evaluation metric. Majority of the features are extended from full fingerprint analysis literature. Along with the metrics specific to evaluation the performance for individual stages, rank- K matching performance if also used for evaluation.	41
2.5	A literature survey of existing algorithms for processing and matching fingerphoto images captured using mobile phones.	44
3.1	Comparison of publicly available fingerprint databases in terms of capture methodology, size, and the types of research challenges that can be addressed.	47
3.2	Different subsets of the MOLF database along with fingerprint type, capture protocol, and its properties.	50
3.3	The nomenclature followed for the five subsets of the MOLF database.	54
3.4	Rank-1 identification accuracy and equal error rate (for verification) pertaining to experiment I (sensor interoperability analysis).	59

3.5	Rank-50 identification accuracy (%) of experiment II (latent matching with manually marked minutiae) and experiment III (latent matching with automatically extracted minutiae).	60
3.6	A summary of the multiple subsets and their variations in the ISPFV-v2	65
4.1	Summary of features used to represent the foreground ridge features.	83
4.2	Experimental protocol for the NIST SD-4, NIST SD-27 , and IIITD-CLF databases. Inked fingerprints from 2000 classes of the NIST SD-4 database are added to extend the gallery.	90
4.3	RELIEF algorithm based feature analysis on NIST SD-4, NIST SD-27 , and IIITD-CLF databases. The most and least contributing features for segmentation on each database are also obtained.	91
4.4	Segmentation accuracy (SA), FSA and BSA (in %) of the proposed and existing segmentation algorithms on the NIST SD-4, NIST SD-27, and IIIT-D combined latent fingerprint databases.	94
4.5	The SIVV-TPR improvement on the three databases, before and after segmentation.	94
4.6	Segmentation accuracy (SA), foreground segmentation accuracy (FSA), and background segmentation accuracy (BSA) (in %) of the proposed and existing feature selection algorithms on the NIST SD-4, NIST SD-27 , and IIITD-CLF databases.	96
4.7	Rank-50 identification accuracy (in %) of the proposed segmentation algorithms using MCC descriptor. Manually marked minutiae are used for NIST SD-27 while VeriFinger SDK is used to extract minutiae from IIITD-CLF database.	99
4.8	Average number of minutiae extracted in the fingerprint images after segmentation.	99
5.1	A comparison of different types of regularization methods used in the literature for network based architectures	107
5.2	Summary of the composition and characteristics of the heterogenous fingerprint database. This database is used as the pre-training dataset for the proposed deep learning approach.	125

5.3	Summary of latent fingerprint databases used in our experiments, including the number of train patches and test patches used in each of the three cross validation experiments.	125
5.4	Classification results of the proposed and existing algorithms on the NIST SD-27 latent fingerprint dataset. The results are reported in terms of CCA (%). MDA is the Minutia Detection Accuracy and NMDA is the Non-Minutia Detection Accuracy.	127
5.5	Classification results of the proposed and existing algorithms on the MOLF latent fingerprint dataset. The results are reported in terms of correct classification accuracy (%), minutia detection accuracy, and non-minutia detection accuracy. . . .	127
6.1	Summarizing the characteristics and the variations captured in the ISPFV2	143
6.2	The algorithmic pipeline of the different techniques used for comparison along with the proposed pipeline.	146
6.3	Performance comparison of different feature extraction and matching algorithms in terms of Equal Error Rate (EER) (%). The preprocessing pipeline (segmentation + enhancement) is consistent across all feature extractors.	147
6.4	Understanding the effect of resolution on fingerphoto matching with <i>Exp.2</i> . The results are reported in terms of EER (%).	150
A.1	Overview of the standard image data sets used in the experiments.	162
A.2	The performance of the proposed GSAE algorithm using different regularizers on the standard image data sets. The architecture of the autoencoder used for each dataset is also provided.	164
A.3	Comparing the performance of different variates of autoencoder based algorithms and the proposed algorithm, GSAE, on the MNIST dataset [8].	166
A.4	Classification error on the MNIST database obtained using Deep Boltzmann Machines and Deep Belief Networks. The table shows the error (%) of the classifier, where 0.53 means 53 out of 10,000 images are misclassified.	167
A.5	Results on CIFAR-10 database using Deep Boltzmann Machines and Deep Belief Networks. The table shows the matching accuracy (%) of the classifier.	168

Abbreviations

ACE Analysis, Comparison, and Evaluation.

ACE-V Analysis, Comparison, Evaluation, and Verification.

AFIS Automated Fingerprint Identification System.

BMP Bitmap Image.

CCTV Closed-circuit television.

CDEFES Committee to Define Features for Fingerprint Systems.

CJIS Criminal Justice Information Services.

CMC Cumulative Match Characteristics.

CMOS Complementary metal-oxide-semiconductor.

CMYK Cyan Magenta Yellow Key (black).

COTS Commercial Off-The-Shelf.

cssDBM Class Sparsity Specific Deep Boltzmann Machine.

cssDBN Class Sparsity Specific Deep Belief Network.

cssRBM Class Sparsity Specific Restricted Boltzmann Machine.

DBM Deep Boltzmann Machine.

DBN Deep Belief Network.

DISR Double Input Symmetrical Relevance.

DNA DeoxyriboNucleic Acid.

DTV Directional Total Variational.

EER Equal Error Rate.

EFS Extended Feature Set.

ELFT-EFS Evaluation of Latent Fingerprint Technologies, Extend Feature Set.

FAR False Accept Rate.

FTE Failure To Enroll.

FTP Failure To Process.

FTR Failure To Register.

GAR Genuine Accept Rate.

GPU Graphics Processing Unit.

GSAE Group Sparse Autoencoder.

GUI Graphical User Interface.

IAFIS Integrated Automated Fingerprint Identification System.

IIITD-CLF IIITD Combined Latent Fingerprint.

IIITD-SLF IIITD Simultaneous Latent Fingerprint.

IRLS Iterative Reweighted Least Squares.

ISPF-D-v1 IIIT-D SmartPhone FingerPhoto Dataset version-1.

ISPF-D-v2 IIIT-D SmartPhone FingerPhoto Dataset version-2.

JMI Joint Mutual Information.

KLD KL-Divergence.

LBP Local Binary Pattern.

LED Light Emitting Diode.

LITS Latent Inter-operability Transmission Specifications.

LQAS Latent Quality Assessment Software.

MCC Minutiae Cylinder Code.

MOLF Multi-sensor Optical and Latent Fingerprint.

MRMR Max-Relevance Min-Redundancy.

NBIS NIST Biometric Image Software.

NFIQ NIST Fingerprint Image Quality.

NGI Next Generation Identification.

NIST National Institute of Standards and Technology.

NIST SD-27 National Institute of Standards and Technology, Special Database 27.

NV No Value.

OBIM Office of Biometric Identity Management.

PCA Principal Component Analysis.

PIN Personal Identification Number.

PPI Pixels Per Inch.

RBF Radial Basis Function.

RBM Restricted Boltzmann Machines.

RDF Random Decision Forest.

RDWT Redundant Discrete Wavelet Transform.

RGB Red Green Blue.

ROC Receiver Operating Characteristic.

ROI Region of Interest.

SDK Software Development Kit.

SDSAE Stacked Denoising Sparse AutoEncoder.

SIVV Spectral Image Validation and Verification.

STFT Short-Time Fourier Transform.

SVM Support Vector Machine.

SWGFAST Scientific Working Group on Friction Ridge Analysis, Study and Technology.

UIDAI Unique Identification of Authority of India.

VEO Value for Exclusion.

VID Value for Individualization.

WSQ Wavelet Scalar Quantization.

Research Dissemination

JOURNALS

1. **A. Sankaran**, A. Malhotra, M. Vatsa, and R. Singh, Fingerphoto Authentication using Deep Scattering Networks, *IEEE Transaction on Information Forensics and Security* (under review). **Impact Factor: 4.332**
2. **A. Sankaran**, A. Majumdar, M. Vatsa, and R. Singh, Group Sparse Autoencoder, *Image and Vision Computing, Special Issue on Regularization Techniques for High-Dimensional Data Analysis*, Elsevier, vol. 60, pp. 64-74, 2017. **Impact Factor: 2.671**
3. **A. Sankaran**, A. Jain, T. Vashisth, M. Vatsa, and R. Singh, Adaptive latent fingerprint segmentation using feature selection and random decision forest classification, *Information Fusion*, Elsevier, vol. 34, pp. 1-15, 2017. **Impact Factor: 5.667**
4. **A. Sankaran**, G. Goswami, M. Vatsa, R. Singh, and A. Majumdar, Class Sparsity Signature based Restricted Boltzmann Machines, *Pattern Recognition, Special Issue on Deep Image Video*, Elsevier, vol. 61, pp. 674-685, 2017. **Impact Factor: 4.582**
5. **A. Sankaran**, M. Vatsa, R. Singh, Multisensor Optical and Latent Fingerprint Database, *IEEE Access*, vol. 3, pp. 653 - 665, 2015. **Impact Factor: 3.244**
6. **A. Sankaran**, M. Vatsa, R. Singh, Latent Fingerprint Matching: A Survey, *IEEE Access*, vol. 2, pp. 982-1004, 2014. (*Appeared as one of the top-10 highly viewed publication of 2014*). **Impact Factor: 3.244**

BOOK CHAPTERS

1. **A. Sankaran**, A. Malhotra, M. Vatsa, and R. Singh, Learning Representations for Uncontrolled Fingerprint Recognition, *Deep Learning in Biometrics*, CRC Press, 2017 (In Press).

PEER REVIEWED CONFERENCE ARTICLES

1. A. Taneja, A. Tayal, A. Malhotra, **A. Sankaran**, M. Vatsa, and R. Singh, Fingerphoto Spoofing in Mobile Devices: A Preliminary Study, *International Conference on Biometrics: Theory, Applications and Systems*, pp. 1-7, 2016.
2. **A. Sankaran**, A. Agarwal, R. Keshari, S. Ghosh, A. Sharma, M. Vatsa, and R. Singh, Latent Fingerprint from Multiple Surfaces: Database and Quality Analysis, *International Conference on Biometrics: Theory, Applications and Systems*, pp. 1-6, 2015.
3. **A. Sankaran**, P. Pandey, M. Vatsa, R. Singh, On Latent Fingerprint Minutiae Extraction using Stacked Denoising Sparse AutoEncoders, *International Joint Conference on Biometrics*, 2014, pp. 1-7, (**Best Poster Award**).
4. **A. Sankaran**, M. Vatsa, R. Singh, Automated Clarity and Quality Assessment for Latent Fingerprints: A Preliminary Study, *International Conference on Biometrics: Theory, Applications and Systems*, pp. 1-6, 2013 (**Best Poster Award**).
5. **A. Sankaran**, M. Vatsa, R. Singh, Hierarchical Fusion for Matching Simultaneous Latent Fingerprint, *International Conference on Biometrics: Theory, Applications and Systems*, pp. 377-382, 2012.
6. **A. Sankaran**, T.I. Dhamecha, M. Vatsa, R. Singh, On Matching Latent to Latent Fingerprints, *International Joint Conference on Biometrics*, pp. 1-6, 2011.

Chapter 1

Introduction

Person identification carries significance in all authentication systems ranging from access control systems to transaction systems. Biometrics is the science of uniquely identifying individuals by means of one or more biological traits an individual possess. Biometrics has the unique advantage of “something that you are” instead of “something that you possess” like a key, ID Card or “something that you know” such as passwords and [Personal Identification Number \(PIN\)](#) [9]. Hence, in the age of biometric authentication it is not required to carry an authentication device or remember additional passwords. Some of the most popular and commonly used biometric modalities include voice, face, fingerprint, iris, and signature.

Fingerprint is one of the most popular and oldest biometric used by mankind. Fingerprint is the ridges and furrow patterns found in the upper skin layer of fingers. The scientific and systematic study of human fingerprints is called dermatoglyphics. Usage of fingerprints could be traced back to the cave-man era [10] where thumb prints were left on the walls along with other carvings to identify sculptures and family members. As shown in [Figure 1-1](#), plenty of ancient carvings and written manuscripts have been discovered proving that fingerprint based identification has been in use for centuries. These traces can be found in most of the civilizations including Babylonian history [11], Egyption history [12], and Roman history [13] showing the natural tendency of biometric identification. One of the ancient puzzles in the Indian history, still not demystified by scientists is Naadi Jothidam [14], the usage of fingerprints to individually identify people through centuries and predict their future.

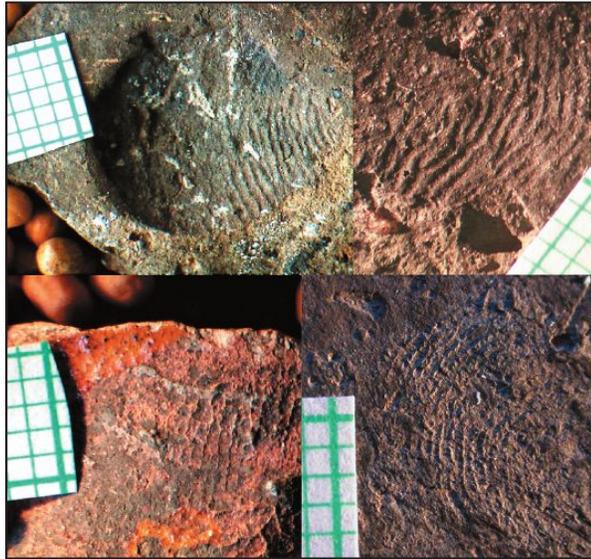


Figure 1-1: Examples of imprints common on ceramic pots from various periods (discovered by Martin Hloek) [1].

Galton's definition of fingerprint [15]

Perhaps the most beautiful and characteristic of all superficial marks are the small furrows with the intervening ridges and their pores that are disposed in a singularity complex yet even order on the under surfaces of the hands and the feet

Fingerprint has been a commercially successful biometric modality utilized for human identification. Until recently in 2015, the market for automated fingerprint identification systems and fingerprint biometric technologies account for the greatest share of the global biometrics market with a total worth of US \$8.49 Billion and is forecast to continue to be the main source of overall market revenues from 2015 to 2020 at an estimated Compound Annual Growth Rate (CAGR) of 21.0% between 2015 and 2020¹. The growing range of applications and extensive analysis has paved way for a whole new era in the field of security, and authentication and allowed researchers to explore more variants of fingerprint recognition. Civil applications such as India's Aadhaar project [16], Department of Homeland Security's US-VISIT program [17], now called as Office of Biometric Identity Management (OBIM), and the UK Border Agency [18] uses rolled or slap (dap or flat) fingerprints for authentication. Fingerprints can be captured using a variety of

¹Automated Fingerprint Identification System Market by Component (Fingerprint Input Modules, Microprocessors/Microcontrollers, Displays, Matchers), Search Type (Tenprint Search, Latent Search), Application, and Geogra-

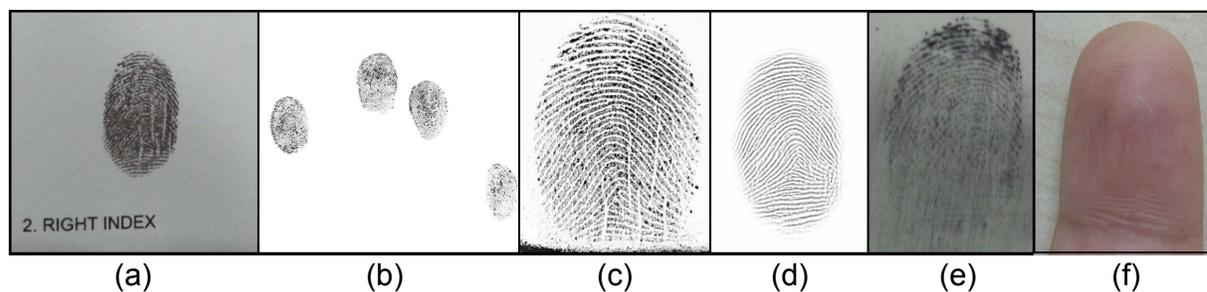


Figure 1-2: Sample images showing high intra-class variation introduced during fingerprint capture. The right index finger is captured from two different subjects using different capture methods. (a) inked fingerprint, (b)-(d) live scan fingerprints: (b) CrossMatch sensor, (c) Secugen Hamster IV sensor, (d) Lumidigm multi-spectral sensor, (e) latent fingerprint lifted using black powder dusting method, and (f) smartphone camera captured fingerphoto image.

capture devices and varying skin conditions introducing a wide range of intra-class variability in captured fingerprints. Figure 1-2 shows multiple fingerprint images from the same finger, captured during the same session using different capture mechanisms. It can be observed that fingerprint data content visually differs a lot based on the capture sensor. On the basis of capture type, fingerprints can be classified into four types: (i) fingerprints, (ii) live-scan fingerprints (including flat and rolled fingerprints), (iii) latent fingerprints, and (iv) fingerphoto images. Figure 1-2(a)-(d) show controlled capture mechanism of fingerprints captured using offline (inked) or live-scan methods while, Figure 1-2(e) show and Figure 1-2(f) show uncontrolled capture mechanism of fingerprint. Both rolled and flat fingerprints can be captured using a live-scan device (for ex, optical and capacitive sensors) or using off-line methods like inked fingerprints. The third type of fingerprint, called latent fingerprint, are unintentional reproduction of fingerprints by transfer of materials from the skin to the surface in contact. The secretions in the surface of the skin such as sweat, amino acids, proteins, and natural secretions when come in contact with the surface, a friction ridge impression of skin is deposited on the surface. These impressions depend on the nature of the skin and the nature of the surface, and are often not easily visible directly to human eyes. Latent fingerprints are extensively used in forensic applications as common evidences in crime scene applications. The fourth type, termed as fingerphoto, is a contactless imaging of finger impression using a camera. A common application of fingerphoto includes use of current day smartphone device or any other handheld electronic device, to capture a photo of the finger region. Due to the contactless nature

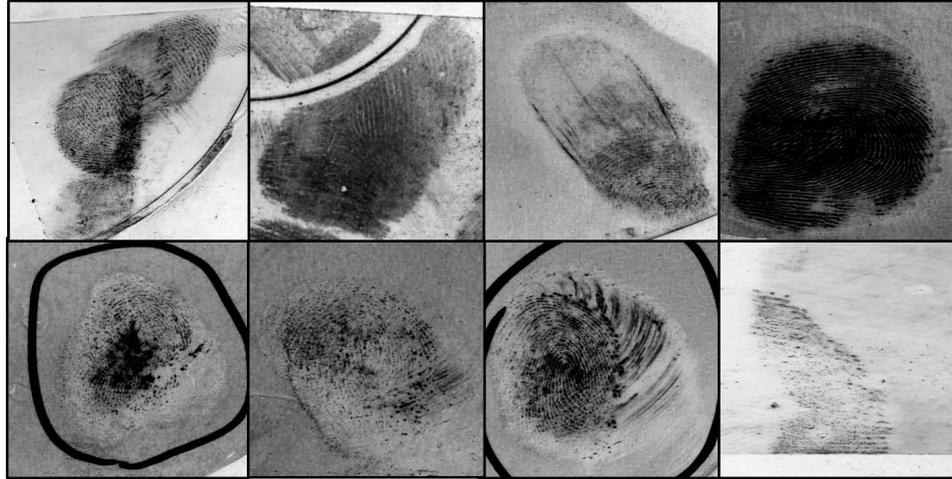


Figure 1-3: Sample latent fingerprint images from the [ELFT-EFS](#) database [2].

of the capture, the ridge valley contrast obtained in a fingerphoto image is highly different from a fingerprint image captured using a live-scan capture device.

Extensive research has been undertaken in recognition of fingerprints captured in a controlled setup such as offline and live-scan methods [19], [20], [9], [21]. In recent trends, there is a growth of applications involving uncontrolled capture of fingerprint images. For example in 2014, Safran introduced a contactless slap fingerprint capturing and recognition system called MorphoWave², to provide border control and access control. Although, the fingerprint is captured in a touchless fashion, the overall environment is semi-controlled in nature providing high quality biometric information. As we progress towards completely uncontrolled applications, such as, forensic applications (latent fingerprint recognition) and commercial applications (smartphone fingerphoto recognition), harder research challenges are introduced. The applications require matching live-scan images with latent fingerprints and fingerphoto images of significantly lower quality and reduced information content. There is an indisputable requirement for undertaking research in these growing applications and create enhanced algorithms to perform fingerprint recognition.

²<http://www.morpho.com/en/public-security/smart-borders/automated-solutions/acquisition-devices/morphowave-desktop>

1.1 Latent Fingerprint Recognition

As shown in Figure 1-3, latent fingerprints vary a lot in quality and information content depending on the nature of the skin (pressure of contact, handling of the item, presence of a transferable matrix on the skin) and type of surface. Among the five example latent fingerprints shown in Figure 1-4, only 2 exemplar fingerprints, (a) and (e), are true matches for the latent prints while the remaining three are false matches or erroneous identifications. Due to the observable challenges such as partial and noisy information, latent fingerprint matching can be uncertain and erroneous.

In practical scenarios, when the size of the background exemplar database is large, the uncertainty is further increased. Manual matching in such cases is not scalable both in terms of time and performance. Automated latent fingerprint matching system could assist the human examination in performing large number of matches especially under uncertain complex circumstances. However, a “lights-out” matching system is still under development and has received research importance in the last few years [22]. As a first step towards building such a system, it is imperative to understand the difficulties involved in automated latent comparison and provide a perspective of the state-of-art.

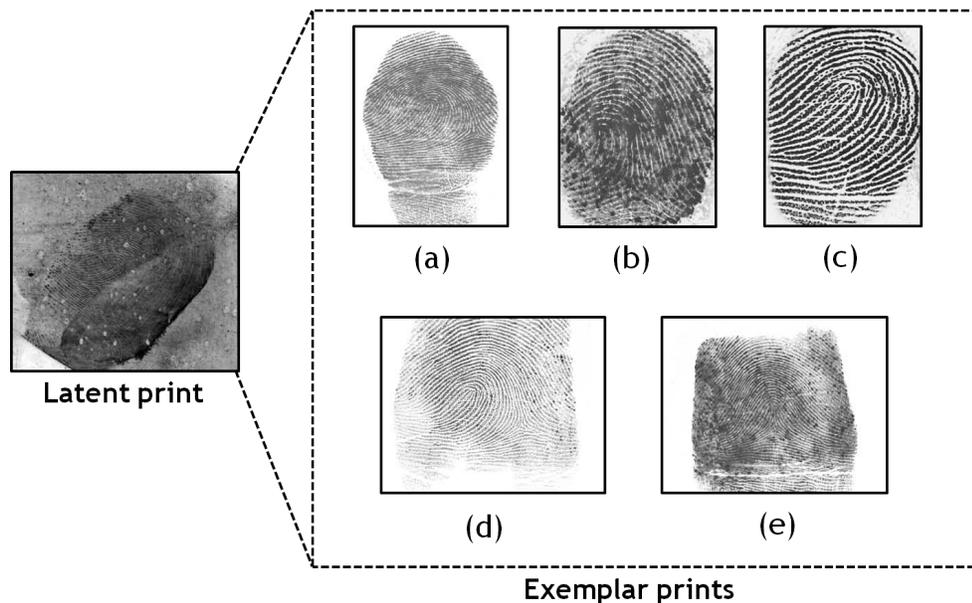


Figure 1-4: An example illustrating the challenge of latent fingerprints. A sample unknown latent fingerprint along with a pool of known exemplar prints (a)-(e) is shown. Of these five latent fingerprints, only two are mated with the exemplar fingerprint, while the remaining three are false matches.

To encourage the growth in technology and research of automated latent fingerprint matching, FBI's [Criminal Justice Information Services \(CJIS\)](#) division awards the title "The Hit of the Year" since 2007. This award is given to the best solved case by [Integrated Automated Fingerprint Identification System \(IAFIS\)](#) using latent fingerprints [23]. This award highlights the utility of latent fingerprints in crime scene investigation and the advancement in latent fingerprint matching technology. Some of the recent recipients of this award are listed below:

- The 2012 "The Hit of the Year" [24] was awarded for solving a 33-year old case of the brutal murder of Carroll Bonnet. In 1969, the collected evidences (latent fingerprints and palm-prints) were not enough to make a positive identification due to the lack of automated bio-metric technologies and the unavailability of large background fingerprint databases. However in 2009, the same evidences of latent prints were sent to FBI's [IAFIS](#) for matching and within five hours [IAFIS](#) returned a set of possible suspects. Upon manual inspection of the suspect's prints, the criminal was identified and found guilty, exactly 33 years after the crime occurred.
- The 2010 "The Hit of the Year" [25] recognized solving of the 1972 San Diego case, where a man was stabbed more than 50 times and murdered. In 2008, the case was reopened and the latent fingerprints lifted from the crime scene were matched by FBI's [IAFIS](#) system. The system returned the top 20 matches and upon further manual investigation, the latent fingerprint was correctly individualized to the murderer who then pleaded guilty. Once again the latent fingerprint along with the murderer's [DeoxyriboNucleic Acid \(DNA\)](#) served as major evidence in solving a cold murder case.

The above case studies show that latent fingerprints could be used as an informative evidence in the court of law. It is to be noted that, at many places, latent fingerprint matching is still performed in a manual or semi-automated environment. Also, the improper application of the matching methodology arising from human inconsistency sometimes leads to erroneous results. These mistakes are compiled in the Innocence project [26], [27] and some case studies are discussed below:

- Shirley Mckie fingerprint case [28] was one of the high profile cases of false accusation. Shirley Mckie, a Scottish police officer, was wrongly charged with perjury after her finger-

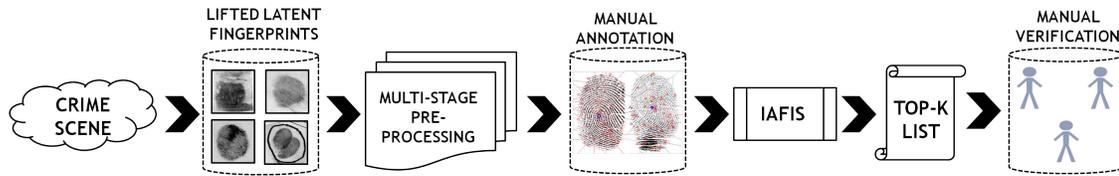


Figure 1-5: A stepwise, semi-automated procedure for analyzing latent fingerprints obtained from a crime scene. The dotted cylinders represent the human intervention in the latent fingerprint identification process.

prints were found at the murder scene of Marion Ross. David Asbury was the prime suspect as his fingerprints were found on a gift tag in Ross’s home. However, four expert examiners provided testimony for Shirley’s latent fingerprint match and Shirley was arrested. The only evidence that was held against her was the latent fingerprint and after months of imprisonment she was released without a formal apology.

- Another high profile case was related to Madrid bombings in 2004, when Brandon Mayfield, an American lawyer was wrongly arrested [29]. The latent fingerprints obtained from bomb site were matched using an FBI system and it returned a match with Brandon Mayfield. After two months of allegation and 14 days of imprisonment, the court released the lawyer declaring his innocence while FBI announced a public apology. The court of law documented that, “The incorrect arrest sprang from an erroneous match of latent fingerprint by FBI’s supercomputer system” [29].

An automated matching technology for latent fingerprints is still in nascent stages and far from being used in real time environments. With growing needs and applications of latent fingerprint matching, there are several challenges faced by the forensic and biometric research community for developing automated systems. Figure 1-5 demonstrates a stepwise procedure for analyzing latent fingerprints obtained from a crime scene. The procedure consists of capture and preprocessing (segmentation and quality enhancement), fingerprint feature extraction, and matching (also known as fingerprint comparison). Fingerprint feature extraction requires ridge quality enhancement of a given latent print and segmenting the ridge like patterns from a noisy background. Fingerprint specific level 1, level 2, and level 3 features including extended features are extracted from partially available print. Feature extraction process results in a fingerprint template which is a representation to uniquely identify a latent print. The matching process compares such templates to verify

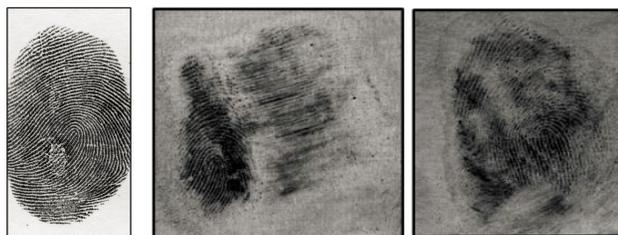
or identify the individual to whom the latent print belongs. Generally, the features of a latent fingerprint lifted from a crime scene are manually annotated by forensic experts. An IAFIS matches the annotated latent print with the background (reference) database of ten-prints and provides the list of *top-K* probable matches (typically, $K = 50, 100$). The list is then manually verified by a forensic expert to determine if individualization exists. This procedure involves manual intervention at different stages which is time consuming, laborious, and subjective to variations. To reduce manual intervention, automating the entire pipeline of latent fingerprint matching would be effective. Some fully automated crime scene investigation clippings shown in some science fiction movies and television shows are fully fictional. The development of this technology, its accuracy and speed, as depicted in these episodes are still farfetched and fictitious, though that would be the ultimate goal to achieve [30]. Some of the research challenges for building a “lights-out” automated latent fingerprint matching algorithm are shown in Figure 1-6 and described as follows:

- Poor quality of the available ridge information as the ridges would be smudgy and imperfect as shown in Figure 1-6(b). This may be because of the uneven pressure with which the person holds the object or because of loss in information while lifting the fingerprint.
- Availability of partial latent fingerprint ridge information as the entire distal phalanx bone region may not come in contact with the object and the entire fingerprint might not be deposited on the surface, as shown in Figure 1-6(a).
- Presence of background noise because the latent fingerprint could be lifted from any surface that comes in contact with the hand. Hence, the amount of distinguishable ridge information depends on the background surface characteristics such as type, material, and texture. These constitute the background noise of the latent fingerprint.
- Non-linear distortion in ridge information as the surface from which the latent fingerprint is lifted need not be always flat and also the skin surface is elastic in nature. The ridge information in the fingerprint gets distorted or warped in a non-linear manner, with respect to the shape of the surface, as shown Figure 1-6(c).

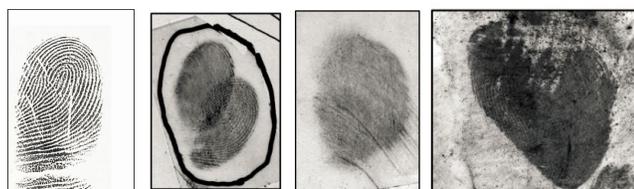
In 2010, Dror and Mnookin [31], discussed the risks and challenges in using automated technologies in forensic fingerprint matching. They conducted studies to understand the implications



(a) Partial availability of latent fingerprint information



(b) Poor quality of ridges and presence of smudges and dusting noise



(c) Presence of non-linear distortion in latent fingerprint ridges

Figure 1-6: Challenges in latent fingerprint matching compared with their corresponding full fingerprints.

and consequences of the technology in [Automated Fingerprint Identification System \(AFIS\)](#) based matching. They put forth the opinion that human cognition has not been effectively transformed into latent fingerprint matching technologies. As a consequence, the probability that an [AFIS](#) will produce incidental similarities has not been sufficiently investigated or explored. Research in overall automated latent fingerprint matching technology is still in its preliminary stages. One of the fundamental challenges is the lack of large public latent fingerprint database available for motivating research in this problem. Also, the metric primarily used to evaluate both the intermediate processes and complete matching algorithm is rank-k identification accuracy. Although improving the matching performance is the eventual aim of an automated matching system, defining some evaluation metrics to examine the different stages as such may help to devise better techniques in the future.

1.2 Smartphone Fingerphoto Recognition

In today's world, smartphones and handheld devices are omnipresent and they are identified as one of the fastest spreading technologies [32]. With such a penetrating growth, smartphones have become an inevitable part of our day-to-day life, holding all our personal data in one place. With increasing capabilities and power of mobile phones, its use in e-commerce applications such as mobile banking is also increasing. Hence, access control to these devices should be secure, flexible, and easy-to-use. Traditionally, the access control mechanisms adopted in smartphones include pins, passwords, and patterns. These lock-screen authentication mechanisms are popularly used and well explored in existing smartphones. However, these mechanisms are prone to attacks including over-the-shoulder copy attack. Therefore, it is important to search for another authentication mechanism that is accurate and less prone to presentation or copy attack. In the realm of biometrics, other modalities such as fingerprint, face, and iris have been explored [33], [34], [35] along with biometric gesture based mechanisms in touch screen smartphones [36], [37], [38]. One such example is BioID facial recognition application (<https://mobile.bioid.com/>), which is a multifactor user authentication application using face biometrics. From a usability point of view, fingerprints have been found to be more adoptable and easier than pin based access control mechanism [39].

There are two broad approaches on how finger impression can be used for providing access control to smartphones: (i) fingerprint based authentication, and (ii) fingerphoto based authentication. Fingerprint recognition is performed with the use of specially designed fingerprint sensors. An embedded sensor (external or attached within the display unit) is used to capture the fingerprint, and minutia based approaches are used for matching. Some of the existing smartphones have integrated fingerprint sensors for authentication [33]. However, adding a capacitive or resistive sensor to a smartphone further adds to the cost of the device. Fingerphoto³ based authentication, as shown in Fig. 1-7, utilizes the in-built camera to capture a photo of the finger which can then be used for authentication. With improvements in technology, every smartphone has a good resolution rear-camera that is extensively being used. Using smartphone camera for fingerphoto capture can provide a cost effective method for user authentication. Further, it can be observed that cap-

³Fingerphoto images are hand finger ridge impressions captured directly using a camera in a touchless method. Fingerprint images are ridge impressions captured using a touch based live-scan sensor.

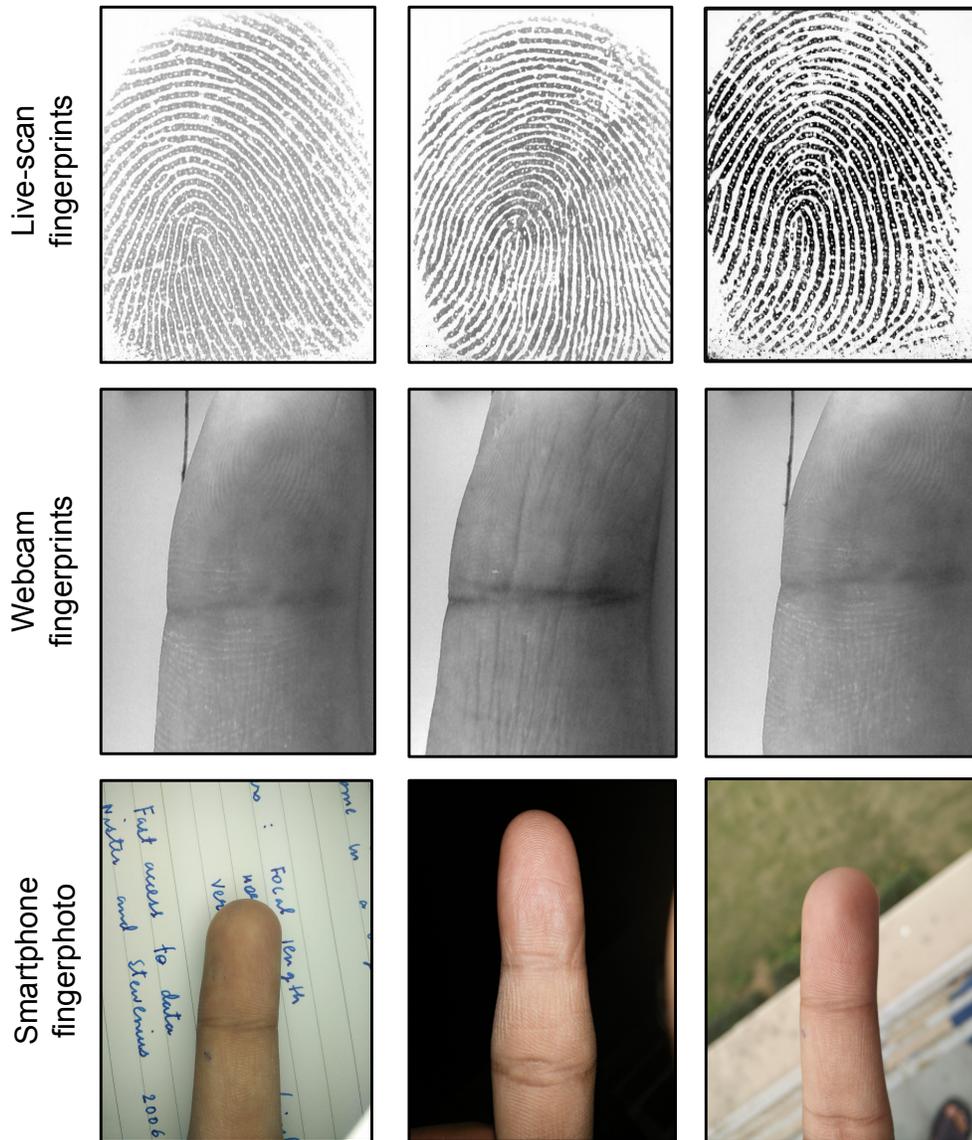


Figure 1-7: Sample images illustrating the variations in the ridge patterns and the noise levels between different types of fingerprint capture mechanisms: (a) live-scan fingerprints image captured using Secugen Hamster IV sensor, (b) contactless finger images captured using low resolution webcam under constrained environment [3], and (c) contactless fingerphoto images captured using three different smartphone cameras - Micromax Canvas Knight, Apple iPhone, and OnePlus One, respectively.

turing fingerphoto images can happen in any kind of indoor or outdoor environment and varying illumination conditions such as broad daylight or night time. The various challenges associated with smartphone camera based fingerphoto authentication can be summarized as follows:

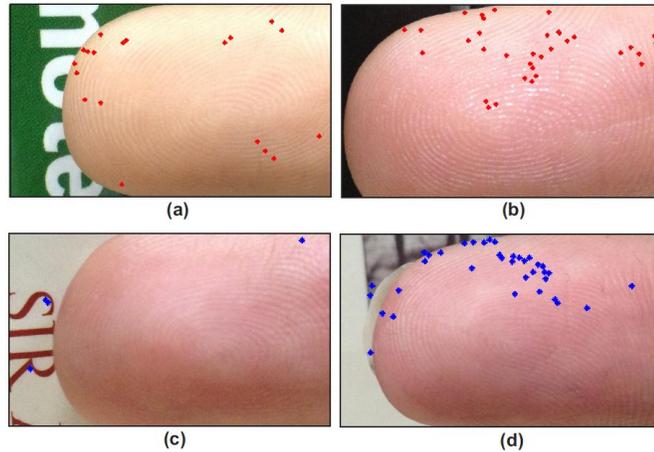


Figure 1-8: Examples showing minutiae extraction using VeriFinger SDK (one of the state-of-the-art fingerprint matching systems) on fingerphoto images captured using smart phone camera.

- **Uncontrolled background:** Real-time capture of images with an uncontrolled background is another important challenge. The distance of the nearest background object can be very close or very far from the finger, making foreground segmentation an arduous task.
- **Varying illumination:** Fingerphoto can be captured in a controlled indoor illumination or in an uncontrolled outdoor illumination settings. During outdoor capture, ambient lighting during day time and night time varies a lot which affects the quality of images. Further, the presence or absence of *flash* during capture makes the preprocessing difficult.
- **Mobile camera:** Cameras in different smartphones have varying features such as resolution, autofocus, and flash **Light Emitting Diode (LED)**, that can affect the quality of the captured image.
- **Finger position:** Challenges arise due to the orientation change of the finger during capture along with the varying distance of the finger from the camera.
- **Feature extraction:** Existing minutia extraction approaches may yield very noisy responses from fingerphoto images [40], as shown in Figure 1-8.

1.3 Research Contributions

Feature representation is an integral component of any object recognition task. A meaningful and representative feature can help in obtaining higher recognition/classification accuracies. However, there is no universal feature extraction algorithm which works best for all types of applications. Therefore, researchers have proposed several feature representation algorithms. Existing algorithms can be broadly classified into two categories: hand-crafted features and learnt features. Majority of the literature has focused on hand-crafted features such as Gabor features, local binary pattern and Scale invariant feature transform. In the last one decade, learning based representation algorithms have gained widespread attention. These algorithms utilize large amount of training data to learn discriminatory feature representations that can tolerate noise and variations in data distribution. Popular examples of such algorithms include dictionary learning and deep learning (autoencoders, deep belief networks and convolutional neural networks). Further, advancements in computing technology and [Graphics Processing Unit \(GPU\)](#) technology has instigated research in learning based representation approaches and almost every domain where abundant data is available, these approaches are providing state-of-the-art results. In this thesis, we build “lights-out” algorithms for automated matching of latent fingerprints and fingerphoto images, without human intervention. We propose novel deep learning based representation models for describing noisy and highly varying information in these two fingerprint variates.

The **first contribution** addresses one of the most important challenge of creating a benchmark dataset and benchmarking the algorithms available for matching the considered fingerprint variates. The performance of existing algorithms is significantly affected when the fingerprints are captured with diverse acquisition methods (multi-session, multi-spectral, multi-resolution, with slap, and with latent fingerprints). The primary challenge in developing a generic and robust fingerprint matching algorithm is the limited availability of large public datasets that capture such intra-class diversity. In this thesis, we create and present two datasets: (i) [MOLF](#) database of more than 19,200 fingerprint images with high intra-class variations and human annotated markup features, and (ii) [IIIT-D SmartPhone FingerPhoto Dataset version 2 \(ISPFD-v2\)](#) containing 16,800 images from 300 classes. The baseline results are established for various matching experiments on these datasets. The datasets are aimed to drive research in building robust algorithms towards solving the

problems of latent fingerprint matching and handling intra-class variations in fingerphoto capture.

The **second contribution** focuses on designing an automated latent fingerprint segmentation algorithm, which is the first stage in the latent fingerprint recognition pipeline. This research focuses on automatically segmenting latent fingerprints from any background noise by distinguishing between ridge and non-ridge patterns. There are three salient features of the proposed segmentation algorithm: (i) a machine learning paradigm for combining five different categories of features for automatic latent fingerprint segmentation, (ii) a feature selection technique using modified RELIEF formulation for analyzing the influence of multiple category features on latent fingerprint segmentation, and (iii) a novel [Spectral Image Validation and Verification \(SIVV\)](#) based metric to measure the effect of segmentation algorithm without the requirement to perform the entire matching process. The results on three publicly available datasets demonstrate the efficacy of the proposed algorithm.

In the **third contribution**, a novel minutia extraction algorithm is proposed as a major contribution towards designing a “lights-out” latent fingerprint matching algorithm. The performance of an automated latent fingerprint identification algorithm is limited due to imprecise automated feature (minutiae) extraction, specifically due to noisy ridge pattern and presence of background noise. We propose a novel descriptor based minutia detection algorithm for latent fingerprints. A novel group (or class) sparsity based $\ell_{2,1}$ regularization method is proposed to improve the features learnt using stacked autoencoders and [Restricted Boltzmann Machines \(RBM\)](#). Minutiae extraction from latent fingerprint is then posed as a binary classification problem to classify the patches as a minutia or a non-minutia patch. To the best of our knowledge this is the first algorithm available in the literature for automated minutia extraction from latent fingerprints. Experimental results show that we achieve more than 90% accuracy in minutia extraction on both, the proposed [MOLF](#) dataset and the existing [NIST SD-27](#) dataset. Using a popular latent fingerprint identification system, against a gallery of two million enrolled identities, the proposed minutia extraction algorithm provides a rank-50 matching accuracy of 69.83% improving the in-built matching performance of 69.21%.

The **fourth and last contribution** is focused towards fingerphoto matching in which a novel end-to-end fingerphoto pipeline is proposed that is invariant to the environmental factors. The ridge-valley pattern present in a fingerphoto is not as distinct as it is present in a fingerprint image,

thus making minutia extraction highly noisy. The matching pipeline consists of a segmentation technique to extract the fingerphoto region of interest from varying background, followed by enhancement to neutralize the illumination imbalance and increase the ridge valley contrast. To deal with pose variations, we then introduce a deep scattering network based representation technique whose resultant texture features are robust and invariant to environmental variations. A random decision forest based classifier is learnt as a verification system, for classifying a pair of fingerphoto images as match or non-match pair. Results and comparison with existing algorithms on the [ISPF-D-v2](#) database show the efficacy of the proposed algorithm which yields the equal error rates in the range of 2.1 - 5.2% for different experimental protocols.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Survey

Two interesting applications of fingerprints that has attracted some research focus are: (i) latent fingerprint comparison for crime-scene analysis and (ii) fingerphoto comparison for smartphone security and customization. Building an end-to-end automated “lights-out” matching system will be a huge contribution to these application domains. Researchers have identified several important challenges in automating recognition for the two fingerprint variates: (1) low information content, (2) presence of background noise and non-linear ridge distortion, (3) different methods of capturing and lifting approach including the resolution of capture, and (4) lack of public latent fingerprint databases. In this chapter, we present a comprehensive survey to summarize the growth of the literature of the two fingerprint variates from a computational and algorithms perspective. The process of automated fingerprint matching is divided into definite stages and the research gaps in each of the stages are analyzed.

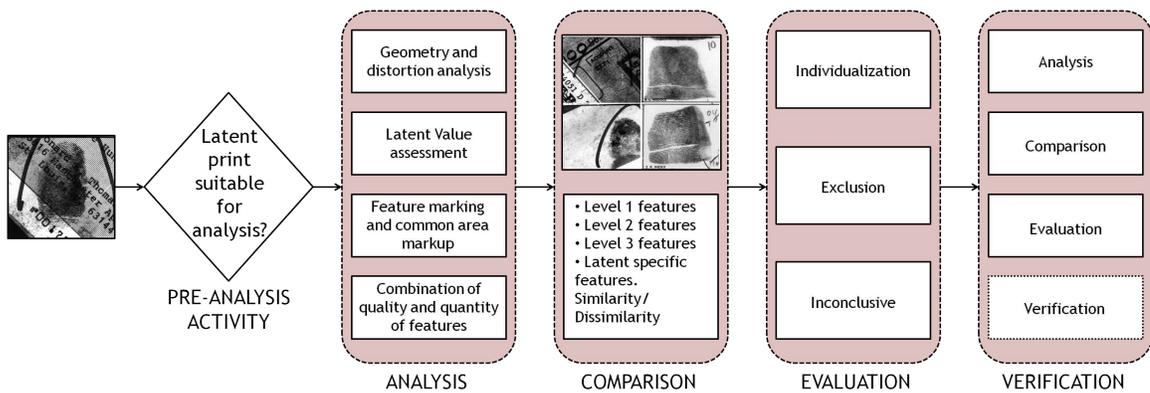


Figure 2-1: ACE-V methodology for manual matching of latent fingerprint.

2.1 ACE-V Method For Latent Fingerprint Comparison

It is important to understand how forensic experts examine and match latent fingerprints as it provides the insight for building an automated system. Human examination of latent fingerprint is performed using the [ACE-V](#) procedure [41]. [ACE-V](#) is a structured, systematic guideline for comparing friction ridge impressions. There are four sequential phases in [ACE-V](#) methodology: Analysis, Comparison, Evaluation, and Verification, as shown in [Figure 2-1](#). After every step, the knowledge gained thus far is applied in the execution of further stages. An overview of the procedure is explained below:

1. **Analysis:** An in-depth friction ridge analysis is performed on a digitally scanned latent fingerprint. The latent fingerprint is studied for different anatomical aspects, deposition pressure, distortion due to pressure, and the substrate matrix. Each fingerprint is assigned a label during this stage - [Value for Individualization \(VID\)](#), [Value for Exclusion \(VEO\)](#), and [No Value \(NV\)](#). The features of latent fingerprint (all three levels) are marked during this stage.
2. **Comparison:** The comparison is a process where visual comparative measurement is made between the latent and the exemplar fingerprints. The comparison is made in a sequential, spatial and configurative manner where marked features are compared in the order of level 1, level 2, and level 3 features. The examiner then compares the unknown print with the known prints, using the three levels of detail, noting all the similarities and differences between the prints.
3. **Evaluation:** The evaluation stage is reached if information between the unknown latent print and known print is enough to make a decision. One of the three decisions is made during the evaluation stage: individualization, exclusion, or inconclusive. Individualization occurs when a latent print is labelled with a known exemplar print, while exclusion occurs when the latent print cannot be assigned to any known exemplar labels. Inconclusive is when the examiner is not able to make a decision regarding the unknown latent prints. If needed, in cases of inconclusive decisions, re-analysis can be performed to check for supplementary information to further assist comparison.

4. **Verification:** Verification can be thought of as a form of peer review. During verification, the entire process of Analysis, Comparison, and Evaluation is verified completely by another examiner to increase the reliability of the process and to ensure repeatability and accuracy of the conclusion. Verification stage can be performed repeatedly as required.

In 2009, the [Scientific Working Group on Friction Ridge Analysis, Study and Technology \(SWGFAST\)](#) created a standard for documenting latent fingerprint matching using [ACE-V](#) method [50]. According to the standards, only the trained latent fingerprint examiners could perform latent fingerprint matching. Every single match stage had to be documented in a specific format, either during the evaluation or soon after it has been done. [ACE-V](#) methodology is generally accepted as a scientific standard for comparing latent fingerprints as it tests the hypothesis of the decision made by the comparison and verification process. In 2005, a [Committee to Define Features for Fingerprint Systems \(CDEFFS\)](#) [51] was formed as part of the [National Institute of Standards and Technology \(NIST\)](#) to define standards, quantifiable methods, and regulations for characterizing the information content of frictional ridge image. By 2011, [CDEFFS](#) proposed [Extended Feature Set \(EFS\)](#) for fingerprints and included them in the ANSI/NIST ITL-1 2011 type-9 record. The [ELFT-EFS](#) (Evaluation #1) demonstrated the performance of minutiae and other features on latent fingerprints. EFS was also presented as the basis for [Latent Inter-operability Transmission Specifications \(LITS\)](#) [52]. The evaluation results are still in its preliminary stage and an increased research focus is set towards designing new and extended features for latent fingerprint matching [22].

2.2 Study of Human Performance for Latent Fingerprint Comparison

In 2006, Wertheim et al. [42], studied the performance of human examination using four factors: the number of correct individualizations made, the number of erroneous individualizations made, the number of clerical errors made, and the assessments of the latent prints regarding the quantity and quality of information present. 92 participants with at least one year of experience were used to make 5861 individualizations. 61 of these matchings were erroneous with 98.95% matching

Table 2.1: Consequences and implications of human performance in matching latent fingerprints. It can be clearly shown that the human examiners can be inconsistent by the introduction of bias. Also, some contradictory results can be obtained from experiments performed in different setup, further adding to the inconsistency in human performance.

Research	Aim of the study	#participants	#comparisons	Results
Wertheim et al., 2006 [42]	Effect of Verification in ACE-V	16	160	<ol style="list-style-type: none"> 1. Each expert given 8 correct and 2 incorrect matches to verify. 2. None of the experts were able to remove even one actual error during verification.
Langenburg, 2009 [43]	Comparison of Analysis, Comparison, and Evaluation (ACE) and ACE-V	6	271	<ol style="list-style-type: none"> 1. ACE-V provides higher precision than ACE. 2. All nine false positives detected during verification, contradicting the observation in [42]. 3. Number of erroneous exclusions doubled during verification.
Dror et al., 2005 [44]	Analyzing the bias of the examiner	27	2484	<ol style="list-style-type: none"> 1. Examiners are manipulated with emotional stories and explicit photos. 2. Increased likelihood of making match judgments for ambiguous fingerprints.
Dror et al., 2006 [45]	Influence of the examiner's decision by a context (bias)	5	-	<ol style="list-style-type: none"> 1. Previously matched prints were reproduced to the same examiner with an additional context saying "no-match". 2. 80% of the examiners provided contradictory decisions upon the influence of context. 3. Experts are vulnerable to irrelevant and misleading contextual influences.
Dror and Charlton, 2006 [46]	Consistency of examiners when provided additional information (bias)	6	48	<ol style="list-style-type: none"> 1. When provided additional information about the case, only 33.3% of trials were consistent. 2. Major reason for inconsistency could be the active and dynamic nature of information processing by humans.
Hall and Player, 2008 [47]	Consistency of examiners with emotional context (bias)	70	-	<ol style="list-style-type: none"> 1. Subjectivity of experts on low clarity fingerprints when provided with an emotional context. 2. Context did not have any effect on the final judgement of the experts, contradicting the observation in [45, 46].
Dror et al., 2011 [48]	Influence of experts under the context of target full prints (bias)	20	200	<ol style="list-style-type: none"> 1. During analysis, a variation of about (2.6 ± 3.5) minutiae was observed. 2. A simple train tool and its feedback could be used to attune the examiner's analysis strategies.
Mackenzie et al., 2013 [49]	Availability of non-matching target prints and decisions of previous examiners	24	305	<ol style="list-style-type: none"> 1. Target prints made even low quality latent prints suitable for comparison. 2. Other experts' decision highly influenced an expert in rejected unsuitable latent prints.

accuracy. The error produced were of two types: 0.034% of error in individualization and 1.01% of clerical errors. In 2008, Dror and Rosenthal [53], conducted similar experiments to study the reliability of matches made by experts. Six experts with more than five years experience each were considered for the experiment. Each of the expert, when provided with the same fingerprints matched previously, exhibited only 89% reproducibility accuracy. To evaluate the human performance for latent fingerprint matching and to quantify the error during manual matching, Ullery et al. [54, 55] conducted two different studies in 2011 and 2012 respectively. In the first research, Ullery et al. [54] studied the accuracy and reliability of an expert's decision in latent fingerprint analysis. Three key objectives constituted the study:

- To study the frequency of error: Error is quantified in terms of both false positive rate and false negative rate, as both these false classifications are costly during a latent-exemplar match.
- To study the consensus among examiners: While performing the same latent-exemplar match, if different examiners tend to provide different results, the reliability of such a decision would be low.
- To study the factors affecting the decision of latent examiners that contribute towards variability in results.

A total of 169 latent print examiners, having a median experience of 10 years and with 83% of them certified as latent examiners, participated in the study. The database included 356 latent fingerprints from 165 distinct fingers and 484 exemplars. 744 distinct latent-exemplar image pairs were formed having 520 mated and 224 non-mated pairs. Each of the examiners were randomly assigned 100 image pairs out of the total pool of 744 pairs. It was observed from these experiments that the true negative rate was greater than the true positive rate in manual examination. 85% examiners made at least one false negative error with a false negative rate of 7.5% and a small false positive rate of 0.1%. By independently verifying the results obtained from other examiners, all the false positive matches and most of the false negative matches were removed. Also, the examiners frequently differed in deciding whether the fingerprints had enough information for reaching a conclusion or not. In a recent study [55] in 2012, the same authors studied the repeatability and

reproducibility of decisions made by latent examiners. Generally latent fingerprint examiners use their expertise rather than a quantitative standard to analyze latent fingerprints. It is very useful and interesting to study if latent examiners can repeat their own results independently (intra-examiner study quantifying repeatability) and also if an examiner's results can be reproduced by other examiners (inter-examiner study quantifying reproducibility). A total of 72 examiners were reassigned 25 image pairs after an interval of approximately seven months. The repeatability of comparison decisions was 90% for mated pairs and 85.9% for non-mated pairs. In essence, for a true positive match, an examiner can repeat his own decision only 90% of the times. However, most of the inconsistencies in examination resulted in inconclusive decisions. Also, the inter-examiner study showed that examiners were able to reproduce other's results only 81% of the time, with only 52% for "difficult" types of fingerprints. Similar conclusions were drawn by Dror et al. [48], when they conducted studies for intra and inter consistency among examiners. To remove bias, they used only latent examiners for their studies rather than forensics or psychology students. Statistically, the intra-examiner consistency provided more insights to the subjectivity of an examiner.

Multiple studies that have been performed to assess the performance of human experts in matching latent prints are summarized in Table 2.1. The results of these performance assessments act as working motivation for an automated matching system. It can be observed that human examination is inconsistent from multiple scenarios as various experiments yield contradicting results. Even experienced human examiners can sometimes introduce bias in matching when additional information such as case story, emotional photographs, and complete prints of the target is provided to them. Some conclusions derived from the studies performed on human capabilities in matching latent fingerprints are summarized below:

- Humans set hard thresholds and are very cautious about making a false positive match and end up marking false negative classification mistakes. Therefore, in manual matching very low false positive rates (0.1%) and high true negative rates can be observed. Often the accounted errors are false negative classification mistakes.
- Manual analysis of complex latent prints and comparison with a large exemplar database is challenging.

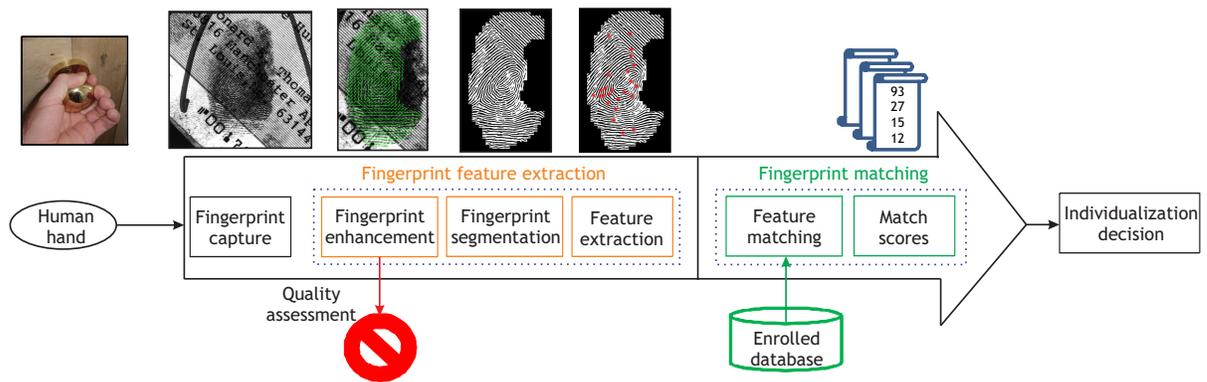


Figure 2-2: The overall schema of an automated latent fingerprint matching system.

2.3 Automated Latent Fingerprint Recognition System

The primary aim of an automated latent fingerprint recognition system is to minimize the human intervention as much as possible. It also optimizes the time required for comparison and improves the throughput for end-to-end comparison. For example, the current FBI's [IAFIS](#) system takes an average time of 1 hour, 53 minutes and 12 seconds for matching a latent fingerprint image against the enrolled gallery of 73.1 million fingerprints [56]. Therefore, an automated latent fingerprint matching system is expected to provide quicker, better and more deterministic results than manual matching. As shown in Figure 2-2, the overall process of an automated matching system can be broken into a set of sequential stages: (1) latent fingerprint preprocessing including quality assessment and segmentation, (2) feature extraction, and (3) feature comparison. The input to an automated system is a digitally scanned or camera captured latent print, that is obtained from a crime scene. Different techniques for latent fingerprint detection, lifting, and capture is an exhaustively studied topics in latent fingerprints [57].

2.3.1 Latent Fingerprint Segmentation

Segmentation involves separating foreground latent fingerprint from any kind of background noise. Latent fingerprint segmentation is a challenging task due to the lack of discrimination in estimating the relevant information and ill-posed boundary of the foreground. In the context of latent fingerprints, the definition of segmentation can be perceived in different ways. Latent fingerprint

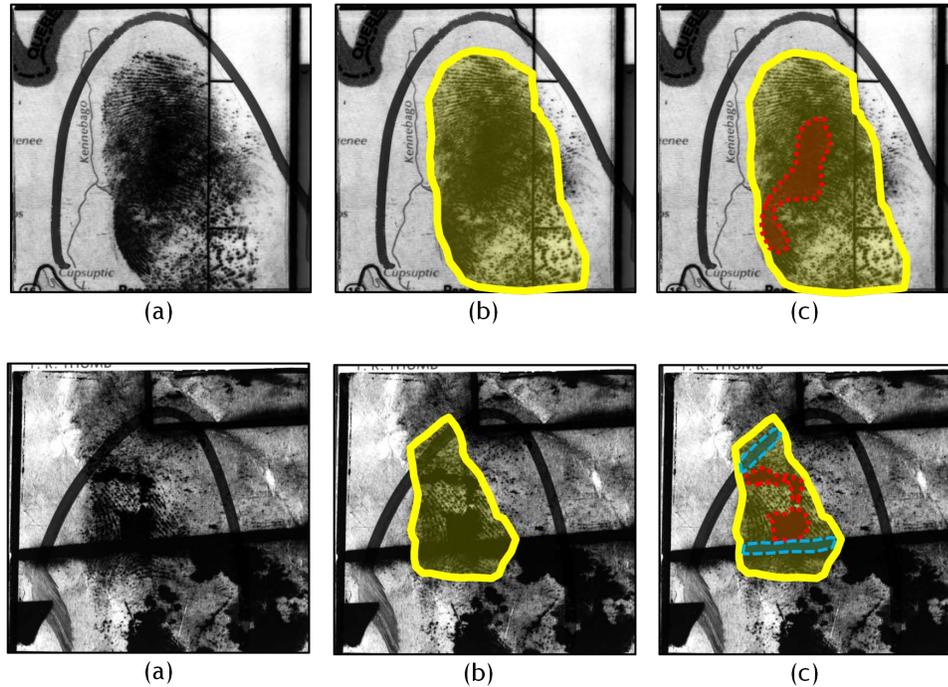


Figure 2-3: Sample latent fingerprints demonstrating different ways of segmenting a latent fingerprint image. (a) original latent fingerprint image, (b) segmentation of the outline of the entire latent fingerprint, and (c) segmentation of the outline of latent fingerprint (yellow full lines) and marking the structured noise (blue dashed lines) and smudgy region (red dotted lines) overlapping with the print.

segmentation may be defined as marking out only the outline boundary, as shown in Figure 2-3(b), or marking out the boundary including the smudges and structured noises inside the boundary, as shown in Figure 2-3(c). Since segmentation is the first step in latent fingerprint matching, the motive of segmentation should be to mark all the foreground regions accurately, while allowing as minimum background as possible.

Even though very few researchers have worked on latent fingerprint segmentation, there are some well understood and accepted challenges.

- Latent fingerprints can be lifted from a variety of surfaces including glass, wood, paper, and metal. The extensive list of surfaces from where latent fingerprints can be lifted vary significantly in texture, pattern, and color. Therefore, background modeling or prediction is a challenging task.
- Due to the variations in pressure applied while depositing and errors while lifting, the ridge information present in a lifted latent fingerprint can be of very poor quality and therefore

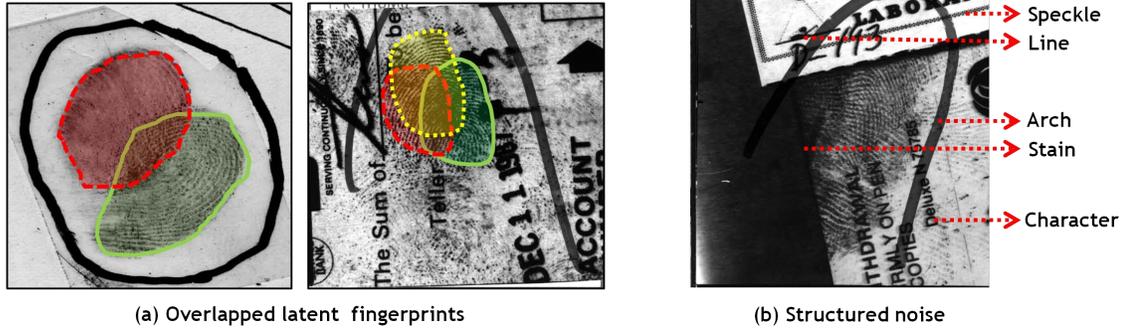


Figure 2-4: Example latent fingerprint images from [NIST SD-27](#) showcasing two specific challenges in latent fingerprint segmentation. (a) Overlapped fingerprints result in overlapped ridge information (multiple touches in the same location) making it difficult to determine the ridge flow of either of the fingerprints, and (b) the presence of structured noise in latent fingerprint background that often resemble ridge like patterns.

Table 2.2: Review of existing latent fingerprint segmentation techniques.

Paper	Methodology	Database	Results	Other metrics
Karimi & Kuo [58], 2008	Variation in ridge frequency and gradient	2 images from NIST SD-27	-NA-	-NA-
Short et al. [59], 2011	Cross-correlation strength	NIST SD-27	Equal Error Rate of 33.8%	-NA-
Choi et al. [6], 2012	Orientation and frequency tensor	NIST SD-27 and WVU DB with extended gallery of 31997 images	Rank-1 accuracy of 16.28% on NIST SD-27 and 35.19% on WVU DB	Missed Detection Rate, False Detection Rate
Zhang et al [7], 2013	Adaptive directional total variational model	NIST SD-27 with extended gallery of 27258 images	Rank-1 accuracy of less than 3%	Missed Detection Rate, False Detection Rate, Feature extraction
Cao et al. [60], 2014	A coarse and fine structured ridge dictionary	NIST SD-27 and WVU DB with extended gallery of 31997 images	Rank-1 accuracy of 61.24% on NIST SD-27 and 70.16% on WVU DB	-NA-
Yang et al. [61], 2015	Detection and orientation field based segmentation	NIST SD-27 with extended gallery of 27000 images	Rank-1 accuracy of 38%	Missed Detection Rate, False Detection Rate
Ezeobijesi and Bhanu [62], 2016	Extreme machine learning classifier using fractal dimension features	NIST SD-27 , WVU DB, IIIT-D DB	Total Empirical Error (TER) of 13.96 on NIST SD-27 , 12.60 on WVU DB, and 8.23 on IIIT-D DB	-NA-

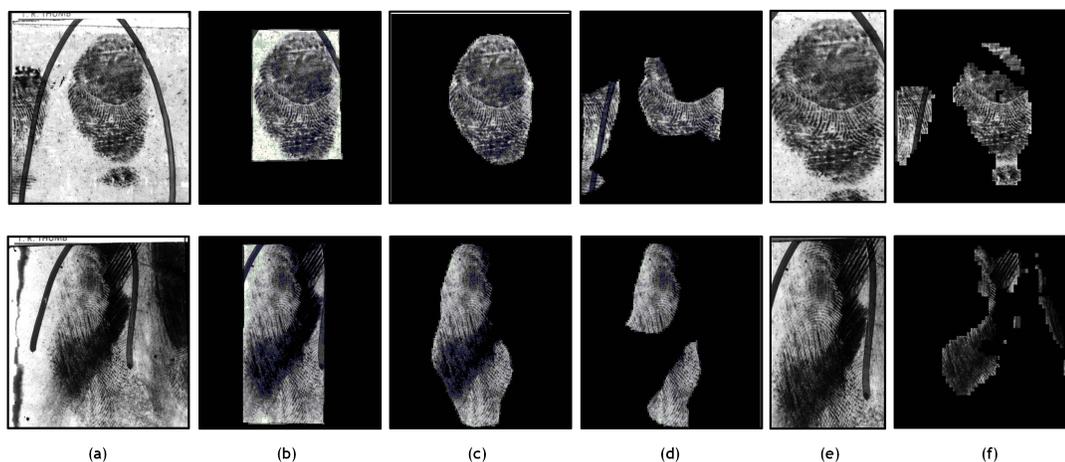


Figure 2-5: Sample latent fingerprints from the [NIST SD-27](#) fingerprint database [4] showing segmentation results. (a) Original latent fingerprint images, (b) manually segmented output with just a bounding box around the fingerprint region, (c) manually segmented output with exact boundary around the fingerprint region, (d) manually segmented output with only the useful ridge information rejecting all the smudgy and noisy (non-informative) regions, (e) segmented output from *nfseg* module of [NBIS](#) [5], and (f) segmented output from Choi et al. algorithm [6] (implemented by the authors).

assessing the quality of ridge patterns contained in a latent print is also challenging.

- As shown in [Figure 2-4\(a\)](#), two or more latent fingerprints may be overlapped during lifting, deposited on top of each other appearing as overlapped ridges when those prints are lifted. Estimating the orientation of the latent fingerprints independently and segmenting them is also a hard problem. As shown in [Figure 2-4\(b\)](#), structured noise such as arch, lines, and characters very often resemble ridge patterns and pose a challenge in differentiating between ridge and non-ridge patterns.

Some latent fingerprint segmentation approaches have been developed in literature which are summarized in [Table 2.2](#). Karimi and Kuo [58] proposed the first automated approach of latent fingerprint segmentation in 2008. They computed the orientation and frequency components at local windows to estimate the regional uniformity property of the fingerprint ridge patterns. A reliability measure is computed using inter-ridge distance for segmenting the foreground image. The results were demonstrated using two images from the [NIST SD-27](#) database [4]. In 2011, Short et al. [59] proposed a segmentation technique by preprocessing latent fingerprints and cross-correlating it with an ideal template of ridge patterns. Based on the correlation strength, the regions were

classified as foreground and background. An **EER** of 33.8% was reported on the **NIST SD-27** database. In 2012, Zhang et al. [63] identified six different patterns of structural noises that could be found in the background of a latent fingerprint - lines, arches, characters, stains, speckles, and others. The authors further proposed a preliminary approach using total variation TV-L1 model to remove the structured patterns and noise in the background. The model is made adaptive by dynamically adjusting the fidelity coefficient that separates the texture patterns of the foreground with the background. The proposed model was observed to perform efficiently for three sample images from the **NIST SD-27** latent fingerprint database. The authors later in 2012, proposed a **Directional Total Variational (DTV)** model [64] which is a variant of TV-L2 model for identifying ridge patterns. The proposed **DTV** model is suitable for decomposing textures with orientation patterns. The extracted orientation vector controls the separation extent of foreground with background. The working of the proposed model is visually demonstrated using three sample images from **NIST SD-27**. More recently, Choi et al. [6] proposed a two step segmentation process using both orientation tensor and frequency tensor (local fourier analysis). The orientation tensor was applied to eliminate structured noise in the background while the local fourier analysis detected ridge like patterns in a local window. The final segmentation output was obtained by intersection of segmented masks obtained from the individual tensors. Experimental results showed the rank-1 identification accuracy of 16.28% on the **NIST SD-27** database and 35.19% on the WVU database. It was observed that the algorithm failed to segment some low contrast latent fingerprints from the WVU database [65].

The problem of segmentation becomes even more challenging when there are more than one latent fingerprint impressions overlapping partially that need to be separated individually. In 2012, Zhao and Jain [66] proposed a model based approach for segmenting overlapping fingerprints using relaxation labelling algorithm. By mathematically modeling the fingerprint orientation field, the authors attempted to enhance the orientation of the overlapping fingerprints especially for low quality fingerprint images. Two different databases were created for experiments: an overlapping fingerprint database and a simulated latent fingerprint overlapping database. The ground truth orientation field of the overlapping fingerprints was manually marked by the experts and the results showed improvement for both the databases. This research work also pointed out the absence of a database with overlapping latent fingerprints to encourage further research in this area. Feng et

al. [67], further improvised this approach for two specific cases: (i) the mated template fingerprint for one of the overlapping fingerprint is available and (ii) both of the overlapping fingerprints are from the same finger. Specific constraints were added to the constraint based relaxation labelling algorithm to address each of these cases specifically. Experiments were performed in two publicly available database: a simulated ten-print overlapping database and a latent fingerprint overlap database. The proposed algorithm approximately showed a rank-1 identification of 85% on latent fingerprint database and 96% on simulated database. Recently, Schott et al. [68] suggested the usage of a latent fingerprint aging feature called Binary Pixel to separate overlapping latent prints. Among the overlapping fingerprints, the age estimation assessed the sequence of latent fingerprint deposition, thereby differentiating the prints. Experimental results showed a success rate of 70%, irrespective of the initial age of either of the print.

An automated latent fingerprint segmentation system is still farfetched from being confidently used in an AFIS. Figure 2-5 shows two sample latent fingerprint images along with its expected manual segmented outputs and the output from *nfseg* module of NBIS [5] and Choi et al.'s algorithm [6] (implemented by the authors). As it can be visually observed, one of the state-of-art algorithms for latent fingerprint segmentation misses out on valid foreground regions in many cases. This shows that there is a scope for further research and improvement in latent fingerprint segmentation. Also, there is no standard definition for the expected output of the segmentation stage in AFIS. As shown in Figure 2-5, the segmentation can be perceived and performed in different ways. In future, a well justified and standard way of segmenting latent fingerprints should be defined such that automated algorithms can work towards that direction.

2.3.2 Latent Fingerprint Quality Assessment and Enhancement

Given a segmented latent fingerprint, a quality assessment has to be made to check if the segmented impression has minimum information to make a valid confident match. Latent fingerprints that do not qualify for minimum information content should be discarded as **Failure To Enroll (FTE)** or **Failure To Register (FTR)** fingerprints [9] and they generally do not affect the performance accuracy of the matching system. Quality enhancement assists the feature extraction process by removing the noise and improving the clarity of a latent fingerprint image. Thus, latent fingerprint

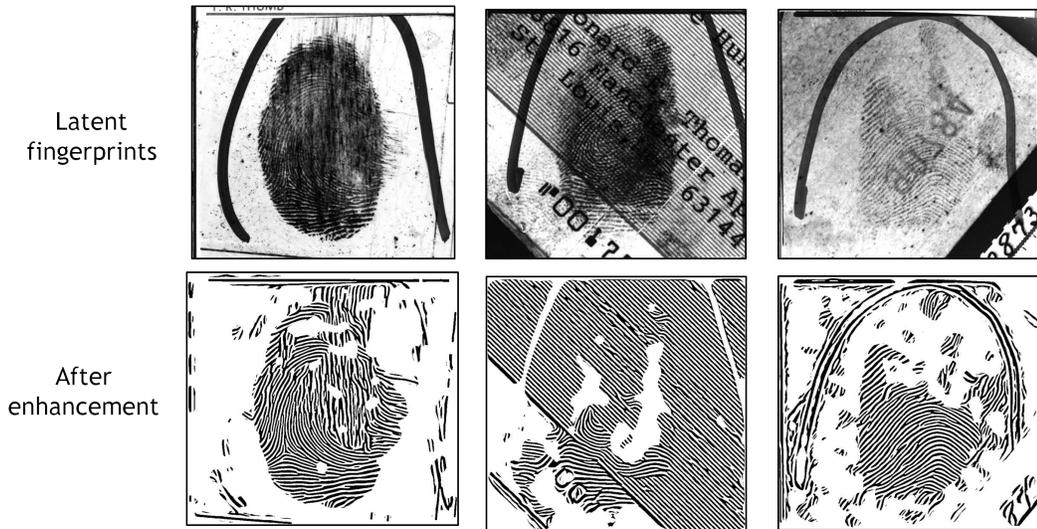


Figure 2-6: Sample latent fingerprints from [NIST SD-27](#) enhanced using VeriFinger [SDK 6.0](#).

enhancement increases the confidence of the features to be extracted. Very few researchers have worked on a quality assessment and improvement of latent fingerprints. Figure 2-6 shows a few latent fingerprints enhanced using VeriFinger [SDK 6.0](#), one of the popular commercial systems used for ten-print matching. It can be observed that the latent enhancement using VeriFinger fails because of the incorrect orientation field estimation of ridge patterns. Some of these general challenges associated with latent fingerprint quality enhancement are summarized below:

- The partial availability of fingerprint ridge patterns is a challenge for ridge quality assessment. Segmentation error affects the performance of quality enhancement.
- Structured noise that resembles ridge patterns such as brush strokes, circular markings, and characters sometimes are enhanced better than the ridge information itself. Also, the ridge information is lost and noise is enhanced when the structured noise overlaps with ridge information. Some contemplating textures in the background similar to ridge patterns are enhanced thereby distorting the actual fingerprint, as shown in Figure 2-6.
- Parameterized enhancement algorithms face challenges in fine tuning their parameters as the environment from which latent fingerprints can be lifted is not limited.

Hicklin [69] in 2007 performed the first study on latent fingerprint quality assessment by comparing the confidence of various levels of fingerprint features towards quality estimation with the

results from human experts. The confidence of matching latent fingerprints using level 1 features was much higher than using level 2 or level 3 features. In 2011, [NIST](#) provided the complete set of experimental features for [NFIQ 2.0](#) [70], which is the quality metric for latent fingerprints. Olsen et al. [71] in 2012 suggested the use of Gabor filters as a candidate quality feature along with other features for [NFIQ 2.0](#). However, they did not publish the results on latent fingerprints and hence its effectiveness in latent fingerprints is still unknown. Yoon et al. [72] provided a metric for latent fingerprint quality assessment. Following the [ACE-V](#) standard for deciding the value of latent fingerprints at analysis level, the authors performed a local ridge analysis to analyze the clarity of latent fingerprints. The ridge clarity maps, combined with the number of minutiae extracted, acted as a good matching dependent predictor of quality latent fingerprints. Using this quality measure, a two-class problem was formulated to estimate if the latent fingerprint is a [VID](#) (Value of Individualization) or not-[VID](#). On a combined database of [NIST SD-27](#) [4] and WVU database [73] with manually extracted minutiae, the authors reported a classification accuracy of 88%. In 2010, Yoon et al. [74] proposed a semi-automated method for enhancing the ridge information using the estimated orientation image. The proposed method utilizes the skeleton image extracted using VeriFinger [SDK](#) to find a coarse orientation map. The coarse orientation field regularization is performed using the “zero-pole model” with a higher order polynomial function. Region of Interest (ROI) and singular points are manually annotated for latent fingerprints and the experiments are conducted using the [NIST SD-27](#) database. The estimated orientation field monotonically increased the matching accuracy over all the quality bins of latent fingerprints. In 2011, Yoon et al. [75], proposed a more robust orientation field estimation technique for latent fingerprint enhancement. For every small non-overlapping patch of fingerprint, a set of coarse orientation fields are initially computed using the [Short-Time Fourier Transform \(STFT\)](#). A set of hypothesized orientation fields using randomized RANSAC based hypothesize-and-test paradigm are generated. Non-overlapping random orientation patches are chosen and tested for orientation consistency based on predefined thresholds. The best-fit regularized orientation field parameter is chosen to enhance the latent fingerprints. Experiments are performed using VeriFinger 4.2 [SDK](#) on latent fingerprints from [NIST SD-27](#) against a combined gallery of [NIST SD-27](#) and NIST 14 databases. The enhancement algorithm shows the rank-1 identification accuracy improvement from 12% to 26%. In 2012, Feng et al. [76], inspired from spelling correction methods employed

in natural language processing, proposed an approach that makes use of the prior knowledge of ridge structure in fingerprint enhancement. A dictionary of reference orientation patches is created using ground truth orientation field and a compatibility constraint between neighboring orientation patches is also applied. Orientation field estimation for latent fingerprint is then posed as an energy minimization problem, solved using a loopy belief network. The average estimation error of orientation (in degrees) is used as the performance metric and is found to be at least 18.44° for the proposed network. More recently, Cao et al [60] presented a coarse to fine, dictionary based ridge flow enhancement technique. A dictionary of ridge structures are learnt from high quality fingerprint images. For any given latent print, the background noise is removed by decomposing using a TV-L1 model. The low quality latent print is then reconstructed using the closest similar dictionary elements. Orientation and frequency parameters for enhancement are estimated from the reconstructed fingerprint. Experimental results of the proposed algorithm when fused with a [Commercial Off-The-Shelf \(COTS\)](#) matcher, showed the rank-1 of about 75% in [NIST SD-27](#) database and about 78% on WVU database. Cao et al [77], further proposed an automated algorithm for latent fingerprint value determination as [VID](#), [VEO](#), or [NV](#). Their feature fusion approach provided about 86% classification accuracy on the combined [NIST SD-27](#) and WVU latent fingerprint dataset.

The term quality has different meaning in biometrics and forensic science communities. In 2013, Hicklin et al. [78] distinguished the concepts of clarity and quality, though the latent print examiners tend to use them synonymously. Clarity is defined as the ability to discern the presence or absence of feature attributes while quality depends on the number of features present. Hence, high clarity regions would be of low quality, if only very few features are available. A prototype of [GUI based Latent Quality Assessment Software \(LQAS\)](#) was created to manually annotate the local clarity regions. The color coded clarity map is visually informative for manual experts and ensures rapid analysis of local regions. The study on local clarity annotation and value determination concluded that there is a strong inter-examiner consistency in clarity boundary assessment but different examiners tend to vary while assigning a clarity value to different regions. Sankaran et al. [79] automated the clarity extraction using a 2-D structure tensor and provided a three bin color map representation. They proposed local orientation fitness as a quality metric and clarity maps to better estimate the quality. Experimental results on the [NIST SD-27](#) database showed that the

quality thus extracted, better predicted the matching performance of latent finger print images.

Latent fingerprint quality assessment and enhancement is a challenging open-ended problem. Extracting orientation field from latent fingerprint requires manual input in terms of singular points and [Region of Interest \(ROI\)](#). There is a huge scope of improvement by developing improved automated techniques for singular point detection as well as segmentation. Quality assessment can either refer to image capture quality or biometric quality which a direct measure of the amount of useful information in a image [80]. In literature, the available information is measured in terms of the number of confident minutiae extracted. However, the information depends on many other factors such as the size of foreground information available, the region of the finger's surface that is deposited, and the clarity of fingerprint ridges. Extracting these features, even though would be challenging, could provide an effective robust quality measure. Also, quality assessment can be made matcher independent or matcher dependent, as different matchers can produce different results for the same input image. Selection of the appropriate metric depends on the application as well as the algorithm used. Quality can be enhanced by not only improving the confidence of the features to be extracted but also by predicting the missing features in latent fingerprints. The latter technique increases the amount of information available for matching and can be given more focus in the future. Also the performance of the quality enhancement process is evaluated by the improvement in matching performance, which in turn depends on many other factors. Hence, some metrics have to be developed to evaluate the performance of quality enhancement as such.

2.3.3 Latent Fingerprint Feature Extraction

Features are the most succinct and precise representation of any data. A fingerprint, basically assumed to be highly variable, needs a very robust feature representation to maintain the uniqueness. Broadly, fingerprint features can be classified into three categories - overall ridge flow pattern (level 1), minutia points (level 2), and extended features (level 3) such as dots, pores, and incipient ridges [9].

1. **Level 1:** The overall ridge flow pattern in a fingerprint is considered as level 1 features. The ridges often flow smoothly, in parallel, except at a few points which are distinctively marked by high curvature or sudden termination of ridges. These points of ridge flow abnormality

are called singular points. As shown in Figure 2-7(a), there are two types of abnormalities in ridge flow pattern - cores and deltas. Henry [81] defined a core point as the “north most point in the inner most ridge line”. Based on the occurrence and position of the core and delta points, fingerprints can be broadly classified into five categories: whorl, loop (left and right), arch, and tented arch. To determine the ridge pattern type and capture the singular points, fingerprint images should be captured at least at 300 PPI resolution.

2. **Level 2:** The minutia constitutes level 2 features. Minutiae are local features of a fingerprint and represent some discontinuity in the flow of ridges. The ridge flow consists of two types of discontinuities - ridge bifurcation and ridge ending, as shown in Figure 2-7(a). Ridge bifurcations are points where a single ridge splits and continues as two different ridges whereas ridge endings are sudden spontaneous ridge terminations. Other general discontinuities in ridge flow are lakes, islands, independent ridges, spurs, and crossovers. Every minutia is represented as $\langle x, y, \theta \rangle$ where (x, y) refers to the 2-D spatial location of the minutia and θ refers to the angular orientation of the ridge flow at (x, y) . To extract minutiae, the fingerprint image must be captured at a resolution of at least 500 PPI.
3. **Level 3:** Level 3 features are fine and intricate features of fingerprint ridges [82–84]. Features such as pores, dots, incipient ridges, ridge width, shape, edge contour, scars, breaks, and creases can be grouped into level 3 features, some of which are shown in Figure 2-7 (b). Although level 3 features are more distinctive in nature, not many automatic feature extraction algorithms exist due to the challenging nature of the problem and lack of higher resolution fingerprint image dataset.

In case of low information content and poor quality of ridge information such as in latent fingerprint, extracting these features is a very challenging task. It is noteworthy to observe that for latent fingerprints, even manual annotation of features can be an arduous and erroneous process. Some of the recent techniques developed to address these challenges are summarized in Table 2.3. In an attempt to perform fingerprint indexing using level 1 features, Feng and Jain [86] in 2008, proposed a background database filtering method. Filtering was performed in three cascaded stages using three different features - pattern type, singularity point similarity, and orientation field similarity. In their experimental study, 258 latent fingerprint images from NIST SD-27 were matched

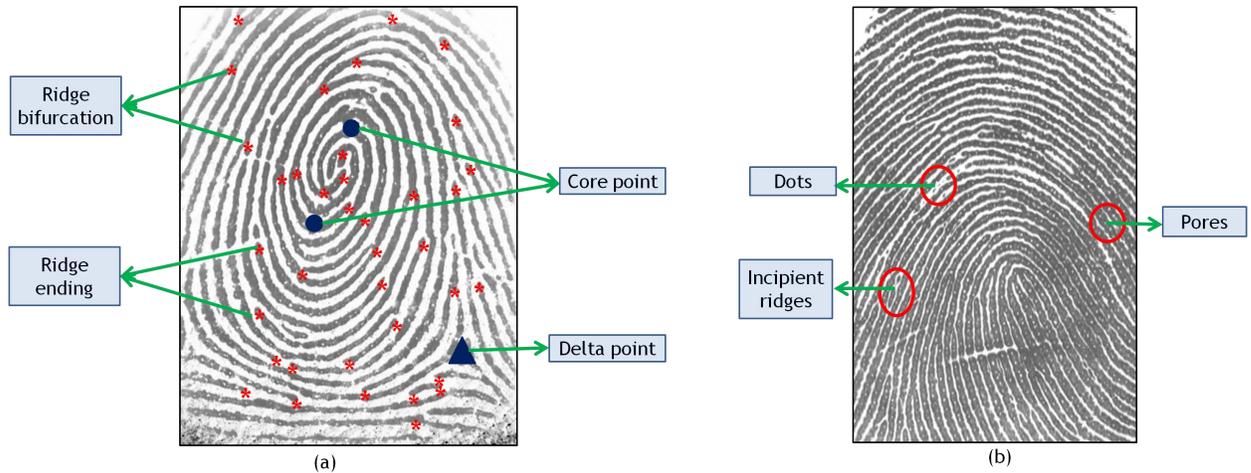


Figure 2-7: Different types of features extracted from fingerprint. (a) shows level 1 and level 2 features and (b) shows level 3 features.

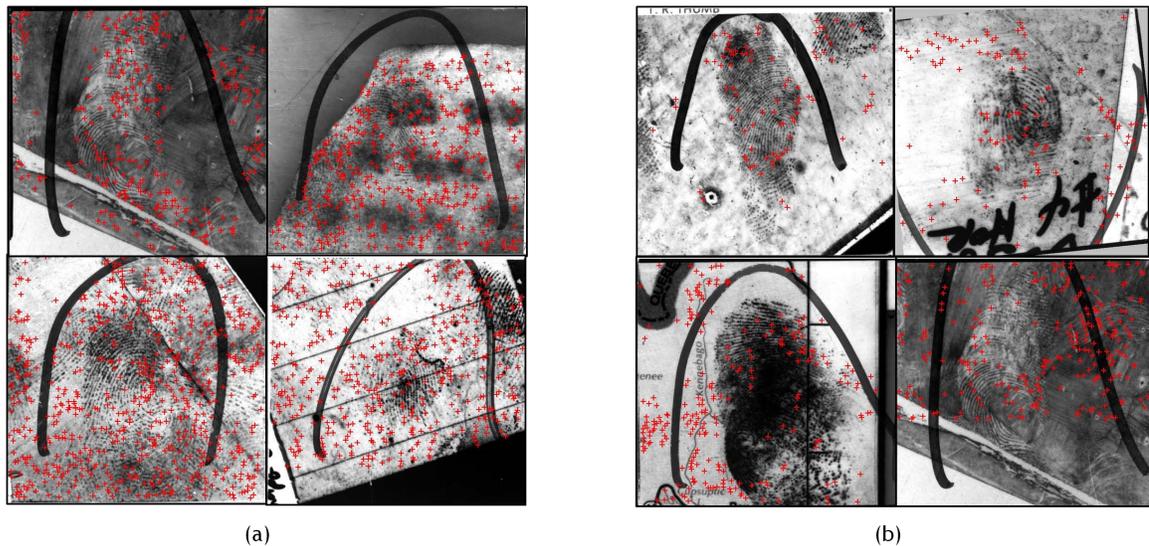


Figure 2-8: Sample latent fingerprints from NIST SD-27 showing spurious minutiae extracted by (a) NBIS and (b) VeriFinger 6.0 SDK.

against a combined database of 10,258 fingerprint images from NIST-4, NIST-14, and [NIST SD-27](#) databases. The penetration rate of 39% was reported with an accuracy of 97.3%. It was also observed that the rank-1 identification accuracy increased from 70.9% to 73.3%. To automatically predict level 1 features, Su and Srihari [87] in 2010, proposed core point detection of latent fingerprints using Gaussian process. The prior joint Gaussian distribution of singular points was learnt and regression was applied to predict the location of singular points. The results were compared with the standard poincare index method [88]. The Gaussian process models were trained using

Table 2.3: A comparison of different descriptors used in the literature to describe latent fingerprint ridge structure.

Paper	Problem addressed	Descriptor used	Remarks
Paulino et al. [65], 2013	Ridge enhancement	Minutiae Cylinder Code (MCC)	Manual annotation of minutiae
Paulino et al. [85], 2013	Descriptor indexing	Minutiae triplets, Binary MCC	Manual annotation of minutiae
Cao et al. [60], 2014	Ridge enhancement	Sparse dictionary learning	Automated segmentation of latent fingerprint
Proposed approach	Minutiae detection	Sparse coding using Stacked Denoising Sparse AutoEncoder (SDSAE)	Manual segmentation of latent fingerprint

fingerprints from the NIST-4 database and tested on the [NIST SD-27](#) database. Ground truth orientation field was obtained by simple gradient method and the ground truth core points were marked manually. The proposed method produced a core point prediction accuracy of 84.5% compared to the poincare index method having 69% accuracy.

Automatic extraction of level 2 features has been attempted on latent fingerprints with very little success. To better understand the performance of minutiae in actual scenarios, Puertas et al. [89], in 2010, compared manual minutiae extraction with automatic minutiae extraction using [COTS](#). The matching performance of latent fingerprints with plain and rolled fingerprints was also compared. A database was created having latent, plain and rolled fingerprint of 50 subjects with an extended gallery of 2.5 million ten-print cards from the Department of Spanish Guardia Civil. The automated system marked, on an average, 31.2 minutiae in the latent prints while the experts marked an average of 25.2 minutiae. Four different experimental scenarios were adopted: (1) using manually annotated minutiae, (2) with automatically extracted minutiae, (3) using top 12 manually annotated minutiae based on confidence, and (4) using top 8 manually annotated minutiae, based on confidence. The performance accuracy of latent fingerprint matching decreased in the same order specified. The authors also mentioned that the quality assessment of latent fingerprints is an open problem that needs to be addressed. In 2010, Paulino et al. [90] attempted to fuse manually marked and automatically extracted minutiae for latent fingerprint matching. Latent fingerprints were enhanced by orientation field reconstruction using the extracted minutiae. The matching per-

formance of these enhanced latent fingerprints was found comparable with the manually marked latent fingerprints. To further improve the performance of manual annotation, different levels of rank and match score fusion were performed. Experiments were performed using latent fingerprints in [NIST SD-27](#) with a combined background database of [NIST SD-27](#) and NIST-14 databases. It was observed that highest rank and boosted-max score fusion performs better than all other fusion methods. In 2011, Jain and Feng [22] provided a complete analysis of latent fingerprint matching with increased number of features and improved matching methods. The feature set extracted from fingerprints were singular points (core and delta), ridge flow map, ridge wavelength map, ridge quality map, fingerprint skeleton, minutiae points, ridge correspondence, and level 3 features (dots, incipient ridges and pores). Features were manually annotated in latent fingerprints. Both local and global matching methods were performed with and without using the additional level 3 features, to study the effect of these additional features. Extensive experiments were performed using 1000ppi latent fingerprints from NIST SD-27(A) with an extended background database of NIST SD-4, NIST SD-14, and NIST SD-27(A). The results show that the extended features were useful and may be utilized only when minutiae extraction is poor. Rank-1 identification accuracy increased from 34.9%, when only minutiae features were used, to 74% when all the features were used. In 2012, Paulino et al. [65] proposed a minutiae alignment technique for latent fingerprints using local descriptor based Hough transform. Minutiae were manually annotated for latent fingerprints while an automated fingerprint feature extractor was used to extract minutiae for background rolled fingerprints. [MCC](#) [91] was used as the local descriptor for minutiae. Minutiae correspondences were established using a simple bounding box algorithm and euclidean distance measure. Experiments were conducted by matching latent fingerprints in the [NIST SD-27](#) database against the combined gallery of [NIST SD-27](#) and NIST-14 using a normalized similarity score metric. The normalized match scores showed a rank-1 identification accuracy of 57.4% when the proposed matcher was combined with the [COTS](#) matcher. In 2008, Vatsa et al., [92] proposed a method to combine pore and ridge features with minutiae for improved verification. Nine different indexing measures were proposed to combine level 1, level 2, and level 3 features. [Redundant Discrete Wavelet Transform \(RDWT\)](#) based local quality analysis is performed. The experiments were performed using 150 high resolution latent fingerprints having level 1, level 2, and level 3 features manually annotated. Quality based likelihood ratio provided a high rank 20 identification accuracy

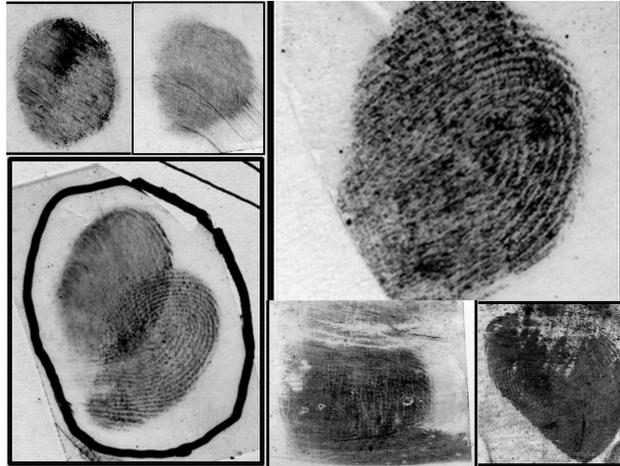


Figure 2-9: Sample images showing high intra-class variation in latent fingerprints captured from the same finger. Images obtained from the [ELFT-EFS](#) public challenge dataset [2].

of 95.35%.

The ultimate aim of latent fingerprint research is to develop a “lights-out” system that can automatically extract valid features from a given latent fingerprint. Figure 2-8 shows many spurious minutiae extracted using [NBIS](#) and VeriFinger 6.0 [SDK](#). In general, it is observed and accepted that the standard algorithms and procedures practiced for live-scan fingerprint comparison do not work on latent fingerprints effectively. The problem of latent fingerprint feature extraction can be viewed as a different problem, rather than an extension or a variation of ten-print fingerprint feature extraction. Though minutiae are the most commonly and widely accepted fingerprint features, in case of latent fingerprints, minutiae based representation may not be distinctive. Some additional properties that might be considered for latent fingerprint feature extraction are as follows:

- Detecting the size of the informative region available in a latent fingerprint can enable us to choose an appropriate technique for comparison.
- Some regions of a fingerprint surface are more informative than the others. Automatic detection of the actual fingerprint region available in the lifted print may provide a better understanding of the actual amount of information available to us.
- The availability of singular points in ridge flow in the lifted fingerprint can provide us additional information. The ridge flow and minutiae extracted around singular points provide distinctive information and are more reliable.

2.3.4 Latent Fingerprint Comparison

The aim of latent fingerprint matching process is to find a similarity or distance score between the two features of gallery and probe latent fingerprints. The matching process should attempt to increase the inter-class variations while decreasing the intra-class variations. Figure 2-9 shows multiple latent fingerprints of the same finger exhibiting extreme intra-class variations. Latent fingerprint matching becomes a complex problem as it has to provide a valid match with just the available limited and noisy features.

In an attempt to perform fingerprint indexing using level 1 features, Feng and Jain [86] in 2008 proposed a background database filtering method. Filtering was performed in three cascaded stages using three different features - pattern type, singularity point similarity, and orientation field similarity. In their experimental study, 258 latent fingerprint images from [NIST SD-27](#) were matched against a combined database of 10,258 fingerprint images from NIST-4, NIST-14, and [NIST SD-27](#) databases. The penetration rate of 39% was reported with an accuracy of 97.3%. It was also observed that the rank-1 identification accuracy increased from 70.9% to 73.3%. Jain et al. [93] proposed a preliminary automatic latent fingerprint matching algorithm in 2008. Features such as minutia, ridge flow, quality map, and orientation field were manually annotated for latent fingerprint matching. The singular points were detected automatically for latent fingerprints and were shown to work better than the poincare index method for latent fingerprints. Two different feature matching strategies were performed: (i) Local minutiae matching and (ii) Global minutiae matching. In local minutiae matching, two different descriptors were used to represent the local minutiae: orientation based descriptor and neighborhood minutiae based descriptor. In global minutia matching, a greedy approach was used, where only the top five matches of the entire minutiae set were considered. Weighted sum score fusion of orientation based and neighborhood minutiae based matching was performed. The experiments were performed using latent fingerprints from [NIST SD-27](#) against a combined gallery of rolled fingerprints from [NIST SD-27](#) and NIST 4. An increased rank-1 accuracy of 79.5% and a rank-20 accuracy of 93.4% were obtained for the proposed matching method. Feng et al. [94], in 2009 proposed a method to match latent fingerprints against the corresponding fusion of flat and rolled fingerprints. The features used were minutiae, quality map, and orientation estimation. Three levels of fusion were separately per-

formed - rank level, match score level, and feature level. Rank level fusion was performed using the highest rank method and Borda count method. Match score level fusion was performed using min, max, sum, product and boosted-max score fusion methods. In the boosted-max match score fusion method, the scores corresponding to genuine matches were boosted by a factor because the spatial transformation in genuine matches was consistent. For feature level fusion between flat and rolled fingerprints, the features were considered from rolled fingerprints in overlapping regions while in non-overlapping regions, features from corresponding image were considered. The experiments performed using the [ELFT-EFS](#) database showed that boosted-max provided the maximum rank-1 identification accuracy of 83% compared to 57.8% for flat and 70.4% for rolled fingerprints. Dvornychenko [95] performed fusion for latent fingerprint matching in three different strategies: (i) fusion of the output of two different classifiers with same feature set, (ii) fusion of the output of same classifier with two different feature sets, and (iii) fusion of the output of two different classifiers with two different feature sets by a specific combination strategy. Experimental results showed that a rank-1 performance boost of 6 – 15% is obtained when multiple features were given to the same classifier and fused. Recently in 2012, Mikaelyan and Bigun [96], established the ground truth of minutiae level correspondences for the publicly available latent fingerprint database [NIST SD-27](#). The authors performed verification tests using two different publicly available matchers, Bozorth3 [5] and k-plet [97], yielding an [EER](#) of 36% and 40% respectively. The results suggest that both the matchers have poor ability to separate genuine and impostor matches in latent versus ten-print matching experiment. However, in an identification setup, at higher ranks, k-plet provided better accuracy than bozorth3 matcher. Kargel et al. [98] in 2012, performed a comparative study of existing exemplar fingerprint matching systems for latent fingerprints. Evaluation was performed to understand the usability of the existing exemplar matching systems and exemplar quality metrics for latent fingerprints. A multi-variate latent fingerprint database, having 480 latent impressions was created. The experiments were performed on four open source fingerprint matching systems: Source-AFIS [SDK](#) [99], FVS [SDK](#) [100], [NBIS SDK](#) [5], Biometrics [SDK](#) [101], and [COTS](#) Innovatrics IDKit PC [SDK](#) [102]. The overall analysis showed that none of the existing exemplar systems used in this experiment could be used as a valid and confident matching system for latent fingerprint matching. It was also observed that the standard quality assessment metric [NFIQ](#) in [NBIS](#), was not an efficient quality measure for latent fingerprints. In

2013, Liu et al [103] proposed an automated feedback mechanism to refine the set of features that are similar between the rolled and latent fingerprints. Using this feedback mechanism the rank list is re-ordered to achieve improved performance. The experiments performed using latent fingerprints from the [NIST SD-27](#) and WVU databases with an extended gallery using NIST SD-14 show an average improvement of about 10%. Vatsa et al. [104] used the multiple latent fingerprints deposited together while holding an object (called simultaneous latent fingerprints) to increase the data available during comparison. In the proposed two step semi-automated procedure, simultaneity of latent fingerprints was first established using geometrical features and minutiae features from multiple prints are fused using likelihood ratio and 2ν -[Support Vector Machine \(SVM\)](#). On a database of 300 simultaneous latent prints against a gallery of 2250 tenprints, it was observed that the use of simultaneous prints improved the rank-1 identification accuracy by 37%. Sankaran et al. [105] created a public database for simultaneous latent fingerprint matching called [IIITD Simultaneous Latent Fingerprint \(IIITD-SLF\)](#) database having almost 360 simultaneous impressions from 60 classes. A completely automated, hierarchical multi-level fusion approach was proposed to combine the information available from multiple latent prints in the same impression. On the [IIITD-SLF](#) database, a rank-10 performance improvement of 12% was observed using the proposed method. Most of the feature extraction and matching techniques in literature have been proposed for matching level 2 (minutiae) and level 3 features from flat and rolled fingerprints. The primary challenge for matching latent fingerprints is the extraction of valid reliable features. Reliable and accurate matching techniques could be devised along with the development of feature extraction techniques. The growth in feature extraction methods would guide the growth in feature matching techniques, as well. Another challenge in latent fingerprint matching, is to transform the human cognition into automated systems to match fingerprint features [106].

Latent fingerprint matching is naturally challenging due to the limited information availability and noisy information. An automated latent fingerprint matching system would be a significant contribution towards crime scene analysis and other forensic applications. To develop such a "lights-out" system, the individual modules must be addressed thoroughly. The research in every single module is at its preliminary stage allowing a large scope of research in this field. With manual annotation of minutiae features, a maximum accuracy of about 75% can be achieved in the [NIST SD-27](#) database. Growth should occur in parallel and in all the modules of a latent

Table 2.4: Steps involved in latent fingerprint analysis, the features used in literature for the individual steps and corresponding evaluation metric. Majority of the features are extended from full fingerprint analysis literature. Along with the metrics specific to evaluation the performance for individual stages, rank- K matching performance is also used for evaluation.

Process	Features used in literature	Evaluation metrics
Segmentation	<ol style="list-style-type: none"> 1. Orientation tensor, frequency tensor [6] 2. Correlation strength [59] 3. Adaptive total variation (TV-L1) [63] 4. Directional total variation (TV-L2) [64] 	<ol style="list-style-type: none"> 1. Missed Detection Rate 2. False Detection Rate 3. Rank-K matching
Quality Assessment	<ol style="list-style-type: none"> 1. NFIQ 1.0 features, frequency domain analysis, local clarity analysis, orientation flow, radial power spectrum, ridge valley uniformity, Gabor filters, and minutiae count [70] 2. Gabor filters [71] 3. Ridge clarity map, number of minutiae [72] 	<ol style="list-style-type: none"> 1. VID and non-VID classification 2. Rank-K matching of different quality bins
Quality Enhancement	<ol style="list-style-type: none"> 1. Dictionary of orientation patches [60, 76] 2. Candidate orientation map, singular points [74, 75] 	<ol style="list-style-type: none"> 1. Average estimation error of orientation (in degrees) 2. Rank-K matching of different quality bins
Feature extraction	<ol style="list-style-type: none"> 1. Descriptor using stacked denoising sparse autoencoder [107] 	<ol style="list-style-type: none"> 1. Patch prediction accuracy 2. Rank-K matching performance
Matching	<ol style="list-style-type: none"> 1. Singular points, ridge flow map, ridge wavelength map, ridge quality map, fingerprint skeleton, minutiae points, ridge correspondence, level 3 features [22] 2. Orientation field, ridge flow, quality map, manual minutiae [93] 3. MCC descriptor for minutiae [65] 4. Manual, automated extracted minutiae [90], [89] 	<ol style="list-style-type: none"> 1. Rank-K matching

fingerprint matching system to overcome the challenges of latent fingerprint matching. The development of automated systems for latent fingerprint matching requires forensic domain experts. A lack of systematic methodology and defined procedure for manual matching of latent fingerprints impediments the growth of automated systems. The knowledge of on-field forensic experts and computational biometric researchers should be brought together to better understand practical challenges in the development of automated systems for latent fingerprint matching.

2.4 Smartphone Fingerphoto Matching

Table 2.5 summarizes these approaches in terms of the database used, challenges addressed, and the solution proposed. In 2005, Lee et al. [108] proposed a ridge segmentation algorithm by building a color model for the foreground skin region. They enhanced the ridge information by computing the ridge orientation using the gradient. On a database of 400 images, they reported a true accept rate of 75% at 0.1% false accept rate. Later in 2008, Lee et al. [109] studied the challenge of fingerphoto quality estimation using gradient information coherence in the local region, under varying poses. They created a private fingerphoto dataset with four subsets and achieved an **EER** of 3.02% over 120 fingerphoto sequences and 1200 fingerprint images. In 2012, Stein et al. [110] emphasized the need of a quality estimation algorithm for fingerphoto images and proposed an algorithm using the ridge edge density in a local region. Using a dataset of 41 subjects from two mobile devices, an **EER** of 19.1% was reported. In 2012, Li et al. [112] studied the performance of ten-print matchers such as Verifinger by Neurotechnology and **NBIS** from **NIST**. With 2100 fingerphoto images captured using three different mobile phones with varying background and illumination, an **EER** of 24.8% – 49.6% was reported for the different matchers. Based on this study, Li et al. [40] observed that minutiae extraction using existing commercial matchers such as Verifinger is extremely noisy and produces lots of spurious minutiae. They proposed a learning based quality estimation algorithm using fingerprint specific features along with **SVM** classifier. In comparison with the manually annotated ground truth labels, they achieved a Spearman correlation coefficient of 0.53. However, both dataset and ground truth annotations were not made publicly available. Stein et al. [113] presented a study using a sequence of fingerphoto images to avoid spoofing the system. The data was collected in a controlled environment and not made publicly available. In 2015, Minaee and Wang [115] proposed using scattering network features with **PCA** and **SVM** for matching touchless fingerphoto captured images captured using a camera in a controlled environment. No segmentation or enhancement approaches were considered and they obtained an **EER** of 8.1%.

Overall, the existing research has focused on fingerphoto preprocessing technique such as quality enhancement, pose correction, and foreground segmentation independently. For matching, primarily minutia based algorithms have been explored. The major limitations of existing research

can be summarized as follows:

- Existing fingerphoto recognition algorithms use minutiae based matching techniques. However, Li et al. [112] have shown that minutia extraction is highly spurious in fingerphoto images. Therefore, there is a need to explore non-minutia based fingerphoto matching algorithms as well.
- Existing research has focused on addressing individual challenges such as segmentation or feature extraction only. There is a lack of an end-to-end matching pipeline that involves preprocessing, feature extraction, and matching to address multiple challenges.
- There is no publicly available dataset and protocol to promote benchmarking in the important problem of smartphone based fingerphoto matching.

Table 2.5: A literature survey of existing algorithms for processing and matching fingerphoto images captured using mobile phones.

Research	Database	Challenges					Algorithm	Results
		Illumination	Background	Resolution	Position	Video		
Lee et al., 2005 [108]	840 fingerphoto from 168 fingers						Segmentation using color model, ridge gradient extraction	Separability: 1.754, Genuine Accept Rate (GAR): 0.75@0.001 False Accept Rate (FAR)
Lee et al., 2008 [109]	120 fingerphoto sequence from 120 fingers, 1200 fingerprints from 60 fingers				✓		Pose, quality estimation using gradient coherence and symmetry	Rejection rate: 5.67%, EER: 3.02%
Stein et al., 2012 [110]	492 fingerphoto from 82 fingers				✓		Quality estimation using edge density	EER: 19.1%
Derawi et al., 2012 [111]	1320 fingerphoto from 220 fingers						Fingerphoto matching using VeriFinger SDK	EER: 4.5%
Li et al., 2012 [112]	2100 fingerphoto from 100 fingers	✓	✓				Fingerphoto matching using VeriFinger SDK and NBIS	EER: 24.8% - 49.6%
Li et al., 2013 [40]	2100 fingerphoto from 100 fingers	✓	✓				Quality estimation using 12 features and SVM	Spearman correlation coefficient of 0.53
Stein et al., 2013 [113]	990 fingerphoto from 74 fingers, 66 finger videos from 22 fingers					✓	Reflection based spoofing detection	EER in the range: 1.2% - 3%
Tiwari and Gupta [114]	156 fingerphoto from 50 fingers				✓		Matching using scale invariant features	EER: 3.33%
Minaee and Wang [115]	1480 fingerprint images				✓		Matching using deep ScatNet, Principal Component Analysis (PCA) , and SVM	EER: 8.1%

Chapter 3

Database Collection

3.1 Introduction

After decades of research, fingerprint recognition has become one of the most reliable and commonly used biometric modality. In 2012, the market for automated fingerprint identification systems and fingerprint technologies contributed the greatest share of the global biometrics market and is forecasted to continue to be the main source of overall market revenues [116]. This can be ascertained by the growing number of deployed applications over the last decade using fingerprint biometric. Some of the notably large scale applications are:

- The [Office of Biometric Identity Management \(OBIM\)](#), previously called the US-VISIT program [117], provides biometric identification services by collecting fingerprints and other biometric modalities from all the visitors applying for U.S. visa. A fingerprint database of over 90 million identities is currently accessed by various federal and state government agencies.
- Aadhaar [118], the brand name of [Unique Identification of Authority of India \(UIDAI\)](#), is one of the largest biometrics projects providing civil and commercial applications for Indian residents. It uses a combination of fingerprint and iris biometrics for de-duplication and authentication.
- FBI [IAFIS](#) [119] is U.S. national fingerprint and criminal history system. It houses one of

the largest fingerprint database, recording more than 70 million suspects, along with more than 34 million civil prints.

Evolution of fingerprint authentication has resulted in a broad spectrum of applications including personal authentication, e-commerce, security, and forensic applications. This widespread usage has also led to emergence of different challenges in fingerprint recognition. Some of these challenges are:

- **Interoperability across multiple fingerprint sensors:** Wide range of intra-class variations can occur based on the method or the sensor by which fingerprint is captured [138]. It can be observed that fingerprints captured from the same finger during the same session, visually differ with variations in capture process or sensor. The report by National Research Council [139] also discusses this important challenge and suggests that the availability of a large database with fingerprint impressions from multiple fingerprint devices can help in improving the performance of algorithms (Recommendation 12).
- **Matching latent prints to slap or rolled fingerprints:** Forensic experts in law enforcement agencies lift latent fingerprints from crime scenes and match them with enrolled databases containing slap or rolled fingerprints. Since the information content and quality of latent fingerprints is significantly different from slap and rolled, there is significant research required to improve the performance of current systems [140].
- **Matching fingerprint images of different resolutions and spectrums:** Fingerprint capture technology was primarily driven by optical and capacitive sensors. However, with growing usage of fingerprint in e-commerce applications and advent of smart mobile phones, matching fingerprints across different resolutions is also gaining importance. Further, there are fingerprint sensors such as Lumidigm Venus, that utilize information from multiple spectrums for fingerprint capture. Matching such images with the ones obtained from optical or capacitive sensors requires additional research.

Similar to other data driven research areas, advancements in fingerprint recognition, especially in the academic community, are dependent on the availability of large databases. Some of the

Table 3.1: Comparison of publicly available fingerprint databases in terms of capture methodology, size, and the types of research challenges that can be addressed.

Capture	Database	Classes	Images	Research Challenges						Characteristics
				session	sensor	spectral	resolution	Slap	Rolled	
Inked	NIST SD-30 [120]	360	1008	✓		✓	✓		✓	Card database. Scanned at 500, 1000 PPI .
	NIST SD-29 [121]	2160	3024	✓				✓	✓	Card database. Scanned at 500 PPI .
	NIST SD-4 [122]	2000	4000	✓					✓	Card database. Five L1 class annotation.
	NIST SD-10 [123]	5520	5520						✓	Card database of rare L1 patterns.
	NIST SD-14 v2 [124]	27000	54000	✓					✓	Card database. Wavelet Scalar Quantization (WSQ) compression.
Live-scan	UCSD WWF [125]	300	300	✓		✓				Wet and Wrinkled fingerprint matching.
	ATVS-FFp [126]	68×2	1632×2	✓	✓					Fake fingerprint matching. Captured using 3 sensors.
	FVC 2000 [127]	110×4	880×4	✓	✓	✓				Low-cost optical, Low-cost capacitive, optical, and synthetic fingerprint subsets.
	FVC 2002 [128]	110×4	880×4	✓	✓	✓				Optical, capacitive, and synthetic fingerprint subsets.
	FVC 2004 [129]	120×4	1440×4	✓	✓	✓				Optical, thermal sweep, and synthetic fingerprint subsets.
	FVC 2006 [130]	150×4	1800×4	✓	✓	✓	✓			Electric field, optical, thermal sweep, and synthetic fingerprint subsets.
	WVU multi-modal [73]	272	7219	✓						CrossMatch, Precise Biometrics, SecuGen sensor at 500 PPI .
	CASIA v5.0 [131]	4000	20000	✓						URU4000 fingerprint sensor.
	MCYT bi-modal [132]	1000	24000	✓	✓					Digital Persona UareU, Precise Biometrics SC-100 sensors.
Camera	NIST SD-24 [133]	100	100 (video)	✓			✓			10 seconds of MPEG-2 Compressed digital video of live-scan fingerprint data.
	HKPU low-resolution [134]	306	3080	✓			✓			Captured directly using a web camera.
	PolyU HRF [135]	148	3170	✓			✓			Captured directly using a high-resolution camera.
Latent	Tsinghua OLF [136]	12	100						✓	Overlapped latent fingerprint segmentation and matching.
	NIST SD-27A [4]	258	258				✓		✓	Latent, 500 PPI and 1000 PPI exemplars. Manual annotation of features for latent prints.
	IIIT-D SLF [105]	180	420	✓				✓	✓	Simultaneous latent, 500 PPI slap. Manual annotation of features for latent prints.
	IIIT-D Latent Fingerprint [137]	150	1241	✓	✓		✓		✓	Latent-to-latent with 500 PPI slap. Latent prints directly captured using a high-resolution camera.
	IIIT-D MOLF Database (proposed)	1000	19200	✓	✓	✓		✓	✓	Dap, slap, latent and simultaneous latent fingerprints. Manual annotation of features for latent prints.

large publicly available fingerprint databases include card ink-print databases, live scan fingerprint databases, multi-sensor fingerprint databases, multi-resolution fingerprint databases, latent with mated full fingerprints databases, and other special databases. A comparative analysis of all the existing public fingerprint databases is provided in Table 3.1 which also provides a listing of the types of research challenges that can be addressed using each database. Existing databases primarily have two limitations:

- they generally contain image variations corresponding to a few challenges only, and
- some of the challenges such as latent fingerprint recognition and cross spectrum matching only have small databases associated with them.

Some of these challenges are being researched upon using non-public databases and therefore it becomes challenging to understand the progression in state-of-the-art in fingerprints and reproduce some of the results. It is our assertion that the availability of a large fingerprint database containing images with variations such as multi-sensor, multi-spectral, and latent vs. optical images can significantly instigate the research in academic community and can help visualize the improvement in the literature. Therefore, we have created a new fingerprint database, termed as Multi-sensor Optical and Latent Fingerprint database¹. The MOLF database contains 19,200 multi-sensor, multi-spectral, dap and slap fingerprint images of 100 subjects obtained from three different sensors along with mated latent and simultaneous latent fingerprints. The latent and simultaneous latent fingerprints have manually annotated features as well. The motivation in forming a new diverse fingerprint database is to have a large latent and simultaneous latent fingerprint collection with the live-scan prints collected using multiple types of sensors. It provides a scope for development, evaluation and performance assessment of fingerprint matching algorithms based on a single variate matching as well as cross-variate matching in various applications. The next section presents the details of the database.

¹Sharing/ downloading information at: <http://iab-rubric.org/resources/molff.html>

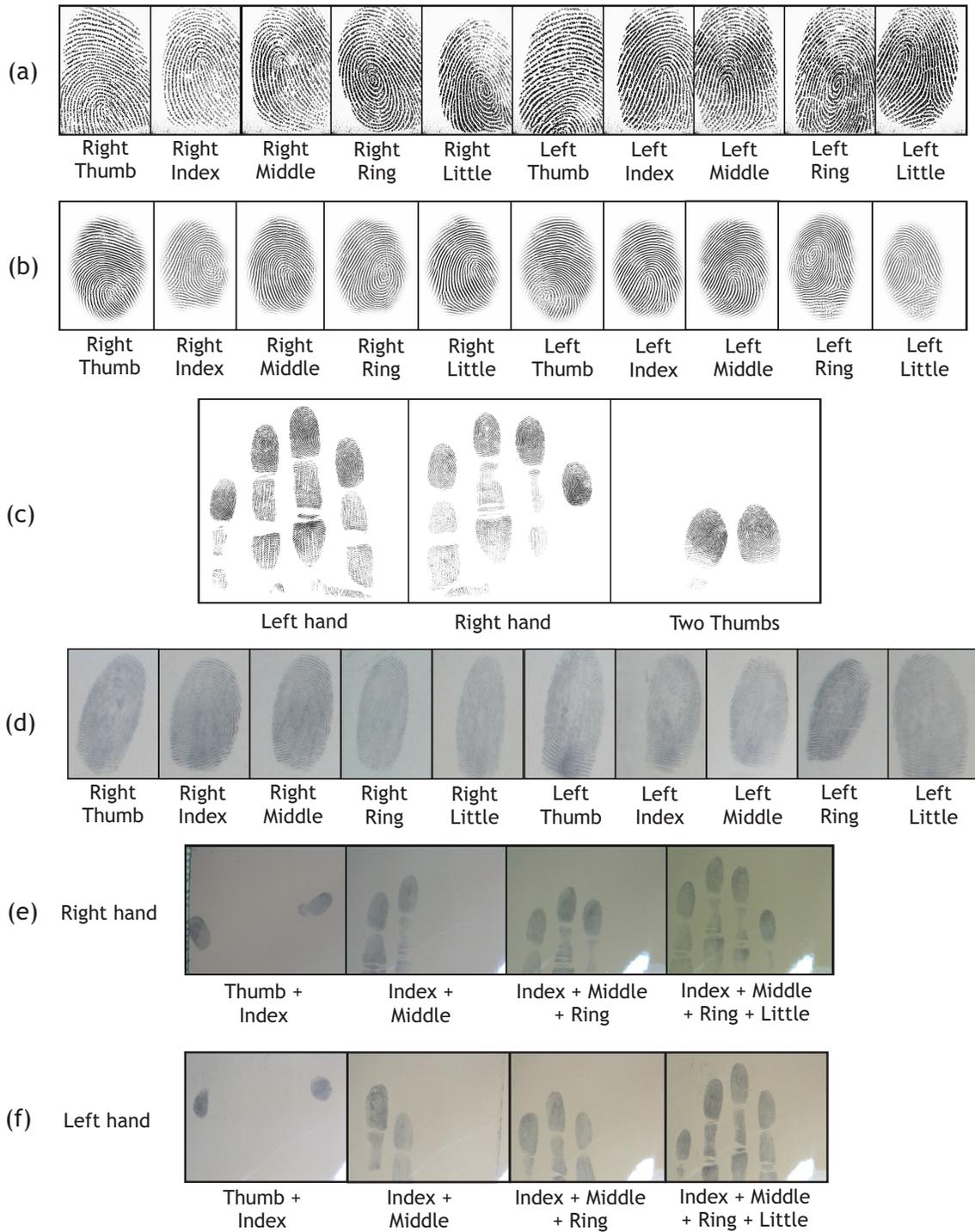


Figure 3-1: Sample fingerprints captured of a subject representing capture variations in the MOLF database: (a) 500PPI fingerprint set from Secugen live scan sensor, (b) multi-spectral fingerprint set from Lumidigm live scan sensor, (c) slap fingerprint set from CrossMatch L-Scan Patrol live scan sensor, (d) latent fingerprint set, (e) simultaneous latent fingerprint set of subject's right hand, and (f) simultaneous latent fingerprint set of subject's left hand.

Table 3.2: Different subsets of the MOLF database along with fingerprint type, capture protocol, and its properties.

Subset	Type	#Images	Image Size	Capture protocol	Comment
<i>DB1</i>	Multi-spectral live-scan	4000	352 × 544	100 users × 10 fingers × 2 sessions × 2 instances	Lumidigm Venus
<i>DB2</i>	Live-scan dap	4000	258 × 336	100 users × 10 fingers × 2 sessions × 2 instances	Secugen Hamster IV
<i>DB3</i>	Live-scan slap	1200	1600 × 1500	100 users × 3 slap prints × 2 sessions × 2 instances	CrossMatch L-Scan Patrol
<i>DB3_A</i>	Live-scan dap	4000	variable	100 users × 10 fingers × 2 sessions × 2 instances	Cropped prints from DB3
<i>DB4</i>	Latent	4400	variable	100 users × 2 hands × 2 sessions × 11 instances	Cropped from simultaneous prints
<i>DB5</i>	Simultaneous latent	1600	1924 × 1232	100 users × 2 hands × 2 sessions × 4 instances	Annotated ROI, core points and minutiae

3.2 Multi-sensor Optical and Latent Fingerprint Database

The MOLF database contains large number of fingerprint images with variations in terms of sensor, resolution, capture spectrum, with slap, latent, and simultaneous latent fingerprint images. Therefore, it provides the opportunity to develop and evaluate algorithms for preprocessing, feature extraction, and matching in different scenarios including latent print matching. As shown in Table 3.1, the database contains 19,200 fingerprint samples from all 10 fingers of 100 individuals (total 1000 classes, treating each finger as a class) captured in two different sessions with an average time difference of 15 days. The database is captured in an indoor environment with controlled lighting. During each session, each individual provides the following information:

1. two independent instances of all 10 fingerprints captured using Lumidigm Venus sensor,
2. two independent instances of all 10 fingerprints captured using Secugen Hamster IV sensor,
3. two independent instances of slap fingerprints (4 + 4 + 2) captured using CrossMatch L-Scan Patrol sensor, and

4. four independent simultaneous latent impressions (2 + 2 + 3 + 4 latent fingerprints) of left and right hand fingers, separately.

A sample fingerprint instance captured from all the sensors is shown in Figure 3-1. Depending on the type of problems that can be addressed, the database is partitioned into six subsets: *DB1* contains the flat dap (all 10) fingerprints collected using Lumidigm Venus sensor and *DB2* contains the same fingerprints collected using Secugen Hamster IV sensor. *DB3* contains the slap fingerprints (4 + 4 + 2 configuration) collected using CrossMatch L-Scan patrol sensor while *DB3_A* contains the dap fingerprints cropped from *DB3* using *nfseg* tool [5]. *DB4* contains the latent fingerprints and *DB5* contains the simultaneous latent fingerprints. Latent fingerprints are obtained by manually cropping the simultaneous latent prints. Table 3.2 provides a summary of different subsets of the database and the details are presented further.

3.2.1 Fingerprint Data Collected with Optical Sensors

The MOLF database has fingerprints taken using three different optical sensors: (i) Lumidigm Venus IP65 Shell, (ii) Secugen Hamster-IV, and (iii) CrossMatch L-Scan Patrol. These three sensors are in compliance with FBI Image Quality Specifications. The resolution of images captured from Lumidigm, Secugen, and CrossMatch sensors are 500ppi each while the image sizes are 352×544 , 258×336 , and 1600×1500 respectively.

For 100 individuals, each of the 10 fingerprint is captured in two sessions and in each session, two independent instances are captured. During the first session, the whole process of collection is explained to all the volunteers (individuals) providing the fingerprints and is also assisted by cleaning their fingers before capture. During the second session, the volunteers are allowed to act upon their own, without forced cleaning. The capture is neither controlled by the expert nor any constraints are applied on the finger's condition. The key motive behind this procedure is to mimic the practical situation of an intentionally registered gallery fingerprint (session I) and an unconstrained probe fingerprint (session II). Thus, for each sensor, there are 4000 images (*DB1*, *DB2*, *DB3_A*) with 1000 fingerprint classes.

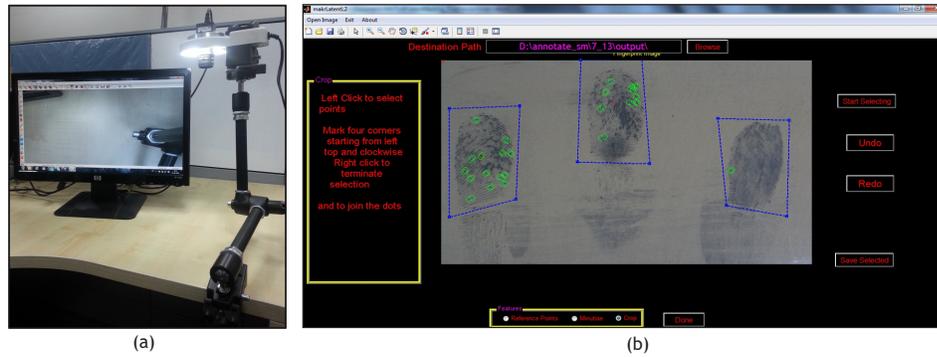


Figure 3-2: (a) The latent fingerprint capture setup utilizing a $1/2''$ CMOS sensor with a 8mm focal length CCTV lens mounted on a Manfrotto magic arm that yields an image of size 3840×2748 and (b) a sample screen shot of the GUI based software tool developed for fingerprint feature annotation.

3.2.2 Latent Fingerprint Collection and Annotation

The latent and simultaneous latent fingerprints are captured with black powder dusting process. The usual method of lifting dusted fingerprints, using forensic tapes, introduces non-linear distortion in the fingerprint ridge information. Therefore, instead of lifting the dusted fingerprints using tapes, a camera setup is created to directly capture the simultaneous latent fingerprint. The camera setup is an improvised version of the setup created during the capture of the IITD-SLF database [105]. The camera setup consists of a USB programmable UEye camera that has a capture size of 3840×2748 pixels. It has a $1/2''$ CMOS sensor and captures at a maximum rate of 3 frames per second. A manual C-Mount CCTV lens having a focal length of 8mm is mounted on the camera which provides finer focus for capturing the latent fingerprint. An illumination ring is attached around the camera to enhance the capture quality. This whole camera setup is mounted on a flexible Manfrotto magic arm - an elbow arm, clamped to the camera on one end using a Manfrotto super clamp and another end is clamped to the table or to any support near the presence of dusted fingerprints. Figure 3-2(a) shows the camera setup used for capturing latent fingerprints.

The volunteers deposit their simultaneous latent fingerprints on a ceramic tile. Though the data collection happens in a closed environment, the participants are completely unconstrained, introducing a large amount of variation and challenges in the deposited latent print. Two different slabs of the same tile are used to capture the left and right hands of the user during a single session. Four different impressions of both hands of the user are captured during each session as follows:

1. thumb and index finger,
2. index and middle finger,
3. index, middle and ring finger,
4. index, middle, ring and little finger.

Fingerprints are then directly captured using the self constructed camera apparatus. Thus, 16 instances of simultaneous latent are captured from each individual in two different sessions. A total of 1,600 simultaneous latent impressions are captured constituting *DB5*. The simultaneous latent fingerprints are manually cropped to get the individual latent fingerprints, thus forming *DB4*. As shown in Table 3.2, there are a total of 4400 latent fingerprints from 100 subjects with 1000 classes. *DB4* contains two latent print instances of every thumb and little finger, four instances of ring finger, six instances of middle finger, and eight instances of index finger.

3.2.3 Latent Fingerprint Annotation

Automatic feature extraction from latent and simultaneous latent fingerprints is an important research challenge [22]. One of the major goals of FBI's [Next Generation Identification \(NGI\)](#) system is to develop a "lights-out" (fully automatic) fingerprint matching algorithm. To evaluate such automated algorithms, a large latent fingerprint database, with manually annotated feature points, is essential. Therefore, to facilitate the evaluation of such systems, manually marked ground truth feature points are provided for latent and simultaneous latent fingerprints in *DB4* and *DB5*, respectively. For every simultaneous latent impression from *DB5*, three different features are marked: (i) [ROI](#) boundary around every finger impression, (ii) singular points - core and delta (only those found within the available impression) on each finger, and (iii) minutiae on all fingers. Two different annotators² independently marked the features, each annotating equal number of images from *DB5*. The annotators marked these features at the rate of 2 – 3 subjects per day and in 22 days the annotation task was completed. The annotators worked for about 8 hours a day with regular breaks to avoid stress. Using the manually marked [ROI](#), individual fingerprints are cropped

²The annotators are not certified latent experts. However, the annotations are made publicly available and can be iteratively improved.

Table 3.3: The nomenclature followed for the five subsets of the MOLF database.

Database	Image Nomenclature
<i>DB1</i>	<i>subjectID_captureID_fingerID</i>
<i>DB2</i>	<i>subjectID_captureID_fingerID</i>
<i>DB3</i>	<i>subjectID_captureID_handID</i>
<i>DB3_A</i>	<i>subjectID_captureID_fingerID</i>
<i>DB4</i>	<i>subjectID_handCode_instanceID_fingerID</i>
<i>DB5</i>	<i>subjectID_handID_instanceID</i>

from the simultaneous impressions and provided as latent fingerprints in *DB4*. The corresponding features for individual latent prints are also separated and provided along with *DB4*.

To enable simultaneous latent fingerprint annotation and to ease the process, we also developed a manual annotation tool in Matlab. A screenshot of the tool is shown in Figure 3-2(b). The GUI based tool allows the annotator to mark the singular (reference) points, minutiae, and ROI. Along with the database and manually annotated feature points, the tool for manual annotation will also be made available to the research community. As the manually annotated features are provided publicly, their accuracy could be improved by further verification from other experts.

3.2.4 Availability of Database

All the fingerprints are available in compressed **WSQ** format and uncompressed BMP format. Table 3.3 shows the naming convention of images in different subsets of the MOLF database. *subjectID* defines the subject number (1-100) while *captureID* defines the capture session instance number (1-4) where 1 and 2 belong to the first session, while 3 and 4 belong to the second session. *fingerID* defines the captured finger number (1-10) with 1-5 from right thumb to right little finger and 6-10 from left thumb to left little finger. *handID* defines the slap fingerprint capture ID where 1 denotes the right four fingers, 2 denotes the left four finger, and 3 denotes the two thumbs. *handCode* defines which hand the simultaneous latent is captured from (L,R), and *instanceID* is the particular instance of capture of the impression where 1-4 belongs to first session and 5-8 belongs to second session. The total size of the database in **WSQ** format is 600 MB and in uncompressed **Bitmap Image (BMP)** format is 18.2 GB. The database is made available for research purpose via: <http://iab-rubric.org/resources/molff.html>

3.3 Research Applications of the Database

MOLF database provides an opportunity to study multiple challenging problems related to fingerprint recognition. Major applications and new research challenges that can be addressed using the database are discussed as follows:

- **Inter-sensor fingerprint matching:** *DB1*, *DB2*, and *DB3_A* contain images captured from three different live-scan fingerprint sensors. By having one of the subsets as gallery and any other as probe, the performance of a fingerprint matcher can be evaluated for sensor interoperability. This also represents a practical scenario where the gallery and probe images are not captured using the same sensor.
- **Latent fingerprint feature extraction and matching:** Forensic applications require matching latent fingerprint with live-scan fingerprints [22]. Extracting reliable features from latent fingerprints is a challenging task. Given the ground truth minutiae annotations, the performance of a minutiae extraction algorithm can be evaluated with good confidence. Also, with an exemplar gallery set (any one of *DB1*, *DB2*, or *DB3_A*) and latent probe set (*DB4*), the performance of a latent fingerprint matching system can be analyzed.
- **Latent to latent fingerprint matching:** The *DB4* subset can be used for evaluating the performance of a latent to latent fingerprint matcher for crime scene linking applications [137]. Since the latent prints in *DB4* consist of multiple instances of the same finger, both gallery and probe can be formed using latent prints in *DB4*.
- **Simultaneous latent fingerprint matching:** The *DB5* subset can be used for matching simultaneous latent fingerprints [104, 105]. Simultaneous latent prints in *DB5* can be matched with live-scan dap prints in *DB1*, *DB2*, or *DB3_A*, and slap fingerprints in *DB3* to evaluate the performance of the matcher.
- **Simultaneous latent fingerprint segmentation:** As the manual segmentation results for simultaneous latent fingerprints in *DB5* are provided, the ground truth can be used to assess the proficiency of automatic segmentation algorithms.

3.4 Experimental Evaluation for Baseline Results

To establish the baseline performance on the MOLF database, several experiments are performed. These experiments are designed to demonstrate the challenges associated with the proposed database and to highlight its usage. The baseline results for livenesscan fingerprint experiments are computed using two fingerprint matching algorithms: **NBIS** [5] and VeriFinger [141]. **NBIS** is an open source minutiae based matching algorithm developed by **NIST** whereas VeriFinger is a low cost proprietary software by Neurotechnology.

Latent fingerprint matching is one of the open research problems that the community is attempting to address. It is important to note that there is no standard latent fingerprint matching **SDK** or commercial system available in public domain, using which baseline can be established. In literature, we have observed that local **MCC** [142, 143] description for manually marked minutiae provides state-of-the-art results [85]. Therefore, **MCC** descriptors are utilized for establishing baseline results on the latent fingerprint dataset.

First, a **NFIQ** [144] based analysis is performed to understand the quality distributions of different subsets of the databases. Thereafter, three different sets of experiments are performed to establish the baseline in different application scenarios. All the experiments are performed in identification mode and the results are reported in terms of **CMC** curve. The first experiment evaluates the performance on optical scanners and the last two experiments pertain to latent prints.

3.4.1 Quality Analysis

Quality of all the fingerprints captured are analysed using **NFIQ** [144]. It is an open source minutiae-based quality extraction algorithm that provides one of the quality values $\{1, 2, 3, 4, 5\}$, with 1 representing the best quality and 5 the worst. **NFIQ** quality distribution of *DB1*, *DB2*, *DB3_A*, and *DB4* are shown in Figure 3-3. In live-scan fingerprints, it can be observed that the images from *DB1* (Lumidigm) have the best quality images highlighting the robustness of multi-spectral images. Lumidigm Venus sensor captures the fingerprint in multiple spectrums and while fusing them, it enhances the image quality. Also, CrossMatch L-Scan Patrol has an in-built quality control mechanism and captures only those fingerprints that pass the quality threshold. However, no such quality constraint is imposed on Secugen Hamster IV scanner, thus some of the finger-

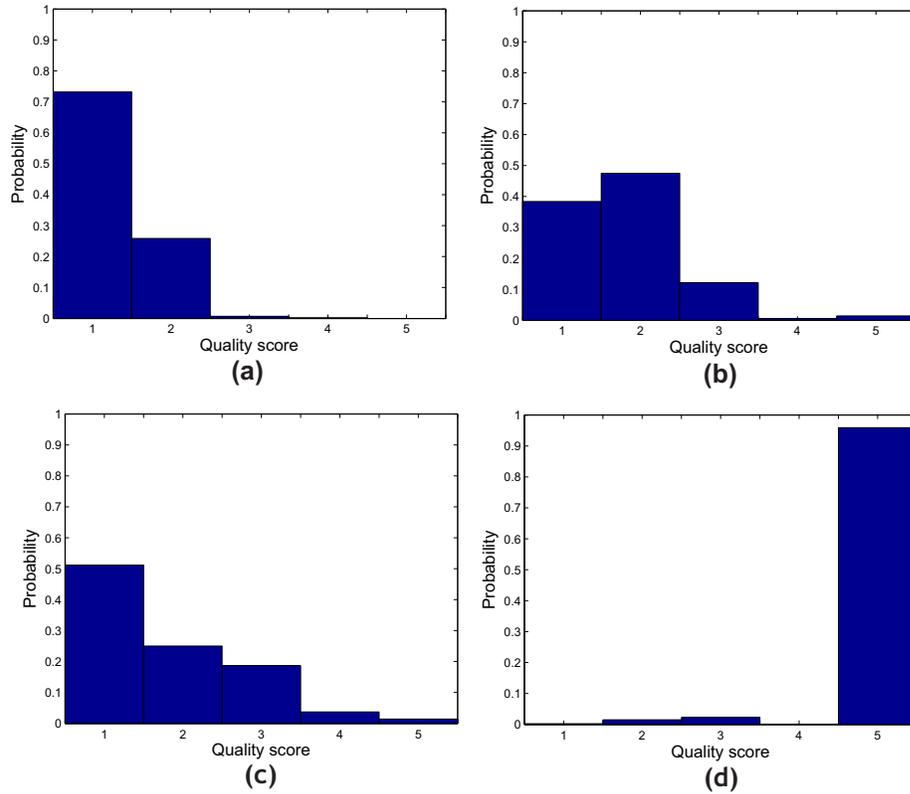


Figure 3-3: NFIQ quality score distribution of (a) *DB1* (Lumidigm) images, (b) *DB2* (Secugen) images, (c) *DB3_A* (CrossMatch) images, and (d) *DB4* (latent) images. In NFIQ measure, 1 denotes the best quality score while 5 denotes the worst.

prints in *DB2* have relatively lower quality scores. As expected, latent fingerprints in *DB4* are poor quality fingerprints with almost 96% of them having a quality score of 5. However, NFIQ is not designed to evaluate the quality of latent fingerprints and a standard (open source) latent fingerprint specific assessment algorithm is still a research challenge. Similarly, there is no exclusive quality measure for simultaneous latent fingerprints (*DB5*) as well. Therefore, this is a high impact research challenge which could be addressed using this database.

3.4.2 Sensor Interoperability Analysis

This experiment (termed as *Experiment I*) is performed to establish the baseline accuracy with fingerprints captured in different sessions using different sensor. In all three subsets, the first two instances captured during the first session are taken as gallery and the fingerprints captured during the second session are used as probe. Thus, the gallery and probe both contain 2000 images

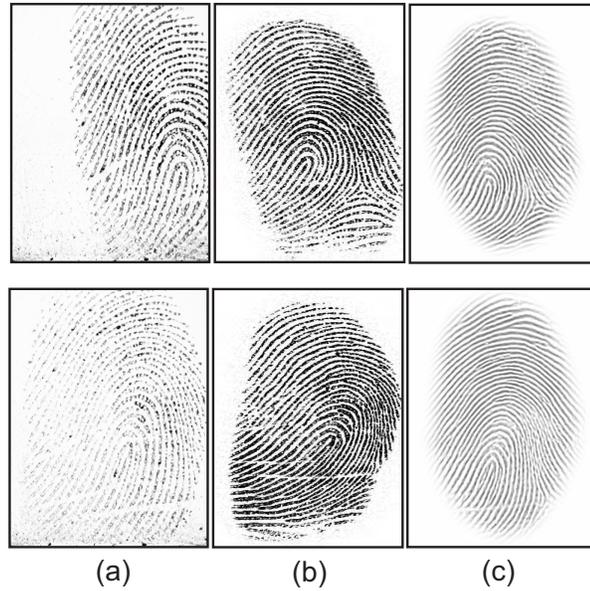


Figure 3-4: Sample images showing quality variations across the three sensors (a) Secugen Hamster IV, (b) CrossMatch L-Scan Patrol, and (c) Lumidigm Venus. It can be observed that some of the images captured using Secugen Hamster IV has a poor capture quality because of its unconstrained capture mode.

pertaining to 1000 (100×10) classes. Datasets *DB1*, *DB2*, and *DB3_A* are used. Since *DB3* contains slap fingerprints, it is not used for this experiment. [NBIS](#) [5] and VeriFinger [141] are then used for feature extraction and matching. Both identification and verification experiments are performed and the results are reported in Table 3.4. The corresponding [CMC](#) curves are shown Figure 3-5, Figure 3-6 and the [Receiver Operating Characteristic \(ROC\)](#) curves in Figure 3-7, Figure 3-8. The major observations made are as follows:

- In experiment I, VeriFinger is observed to yield higher accuracies compared to [NBIS](#) on all three subsets of the database. VeriFinger provides same-sensor rank-1 matching accuracy in the range of 96-98% whereas [NBIS](#) is at least 7% lower in performance.
- From experiment I, it can be observed that matching performance is high when the gallery and probe fingerprints are captured using the same sensor. However, when the gallery and probe fingerprints are captured using different sensors, performance is reduced significantly for both [NBIS](#) and VeriFinger. This highlights that cross-sensor fingerprint matching, especially when one sensor is a multi-spectral sensor, is a research challenge.

Table 3.4: Rank-1 identification accuracy and equal error rate (for verification) pertaining to experiment I (sensor interoperability analysis).

Experiment	Algorithm	Gallery	Probe	Accuracy (%)	EER (%)
I: Live-scan fingerprints	NBIS	DB1 (Lumidigm)	DB1 (Lumidigm)	84.90	8.57
			DB2 (Secugen)	42.50	10.11
			DB3_A (CrossMatch)	43.50	49.67
		DB2 (Secugen)	DB1 (Lumidigm)	44.75	10.05
			DB2 (Secugen)	91.70	7.85
			DB3_A (CrossMatch)	44.70	49.77
		DB3_A (Cross-Match)	DB1 (Lumidigm)	42.45	46.74
			DB2 (Secugen)	43.95	46.67
			DB3_A (CrossMatch)	84.90	08.88
	Verifinger	DB1 (Lumidigm)	DB1 (Lumidigm)	96.75	3.16
			DB2 (Secugen)	47.40	6.46
			DB3_A (CrossMatch)	46.90	6.42
		DB2 (Secugen)	DB1 (Lumidigm)	47.35	6.47
			DB2 (Secugen)	98.10	3.20
			DB3_A (CrossMatch)	46.20	3.94
DB3_A (Cross-Match)		DB1 (Lumidigm)	47.80	6.42	
		DB2 (Secugen)	43.25	3.94	
		DB3_A (CrossMatch)	97.05	3.51	

- Verification experiments performed using NBIS show clear impact of cross-sensor matching, having about 40% more errors than same-sensor matching. However, VeriFinger reduces the effect of cross-sensor matching to great extent showing a difference of only about 3%. Nonetheless, in large scale applications such as India’s Aadhaar project, 3% is a very significant error and might have a greater impact.

3.4.3 Latent Fingerprint Matching

This experiment is performed to establish the baseline accuracy of latent fingerprint matching. There are two different experiments performed on latent fingerprint matching: (i) latent print matching using manually annotated minutiae (termed as *Experiment II*), and (ii) latent print matching using automatically extracted minutiae (termed as *Experiment III*). In experiment II, 4400 latent images in *DB4* are used as probe and they are matched against three different galleries of *DB1*, *DB2*, and *DB3_A*. The results are computed with two different approaches (a) MCC descriptor and (b) *Bozorth3* (an open source matcher) available as a part of NBIS. The results are reported in

Table 3.5: Rank-50 identification accuracy (%) of experiment II (latent matching with manually marked minutiae) and experiment III (latent matching with automatically extracted minutiae).

No.	Experiment	Algorithm	Gallery	Probe	Accuracy
II (a)	Latent fingerprints (manually annotated minutiae)	MCC	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	7.84
			<i>DB2</i> (Secugen)		7.28
			<i>DB3_A</i> (CrossMatch)		5.88
II (b)	Latent fingerprints (manually annotated minutiae)	<i>Bozorth3</i>	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	31.86
			<i>DB2</i> (Secugen)		31.49
			<i>DB3_A</i> (CrossMatch)		33.38
III (a)	Latent fingerprints (automatically extracted minutiae - without FTP)	NBIS	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	6.06
			<i>DB2</i> (Secugen)		9.09
			<i>DB3_A</i> (CrossMatch)		10.60
		VeriFinger	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	6.80
			<i>DB2</i> (Secugen)		6.37
			<i>DB3_A</i> (CrossMatch)		6.51
III (b)	Latent fingerprints (automatically extracted minutiae - with FTP)	NBIS	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	53.03
			<i>DB2</i> (Secugen)		42.42
			<i>DB3_A</i> (CrossMatch)		46.97
		VeriFinger	<i>DB1</i> (Lumidigm)	<i>DB4</i> (La- tent)	55.60
			<i>DB2</i> (Secugen)		49.27
			<i>DB3_A</i> (CrossMatch)		56.09

Table 3.5 and the CMC curves are shown in Figure 3-9. In experiment III, both *mindtct* (NBIS) and VeriFinger are used for feature extraction and matching. Latent fingerprints in *DB4* are matched with live-scan fingerprints in *DB1*, *DB2* and *DB3_A*, individually. The gallery-probe splits used are same as in experiment II. Two sets of experiments are performed: (a) using all probe images in *DB4*, and (b) after removing the Failure To Process (FTP) latent prints from *DB4*. The results of all latent print matching using automatically extracted minutiae are reported in experiment III (a). During automatic minutiae extraction in set III (a) experiments, the minutiae extractor (*mindtct* or VeriFinger) failed to extract even one minutia from several latent probes. In experiment III (b), these images are excluded from the probe set and considered as FTP error [9]. The identification results are reported in Figure 3-9 and Table 3.5. The following key observations can be made:

- Experiment II (a) exhibits that state-of-the-art MCC descriptor provides very low rank-50 identification accuracy of about 5 – 7%, showcasing the challenging nature of latent fingerprints in this database.
- Experiment II (b) shows that with manually annotated minutiae, rank-50 matching accuracy

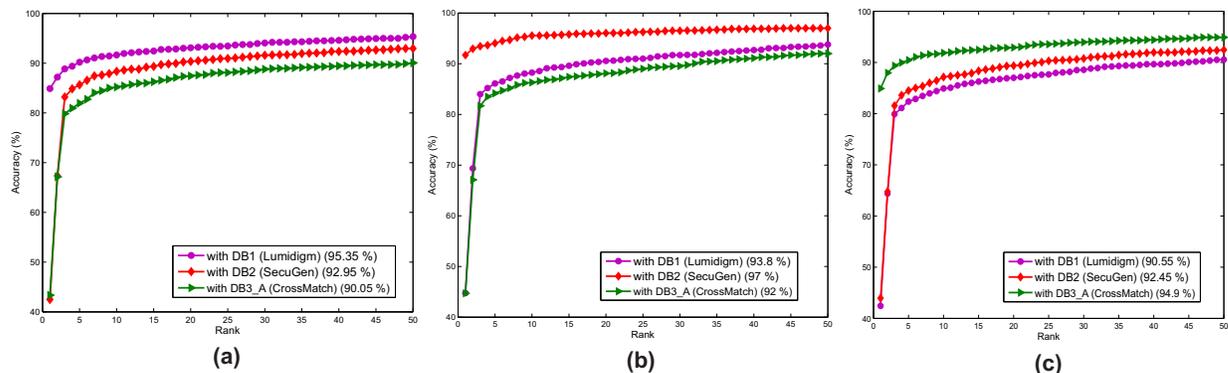


Figure 3-5: CMC curves using NBIS for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability.

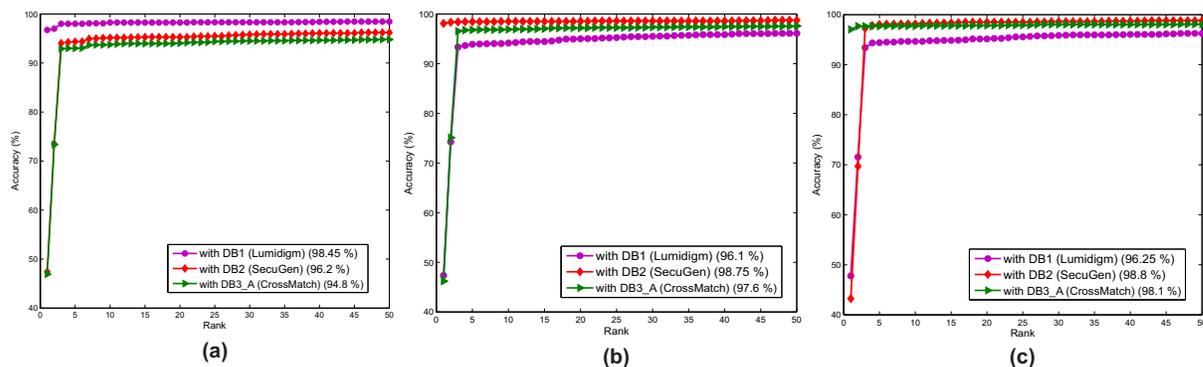


Figure 3-6: CMC curves using VeriFinger for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability

of latent fingerprints is in the range of 31-34%. This indicates that even after manual annotation of minutiae, latent fingerprint matching has a scope for designing robust algorithms for minutiae matching in partial prints.

- For experiment III with *DB4* subset, *mindtct* extracts an average of four minutiae per latent print, while VeriFinger extracted almost 42 minutiae per latent print. On the other hand, an average of 11 minutiae per latent print are marked during manual annotation. This indicates that *mindtct* produces too few minutiae while VeriFinger extracts too many spurious minutiae for latent fingerprint.
- Experiment III (a) shows the results of matching latent and live-scan prints using an auto-

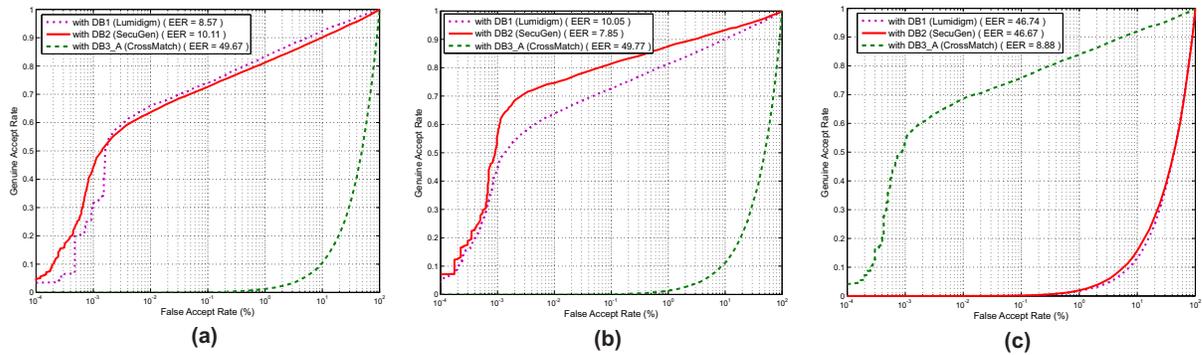


Figure 3-7: ROC curves using NBIS for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability.

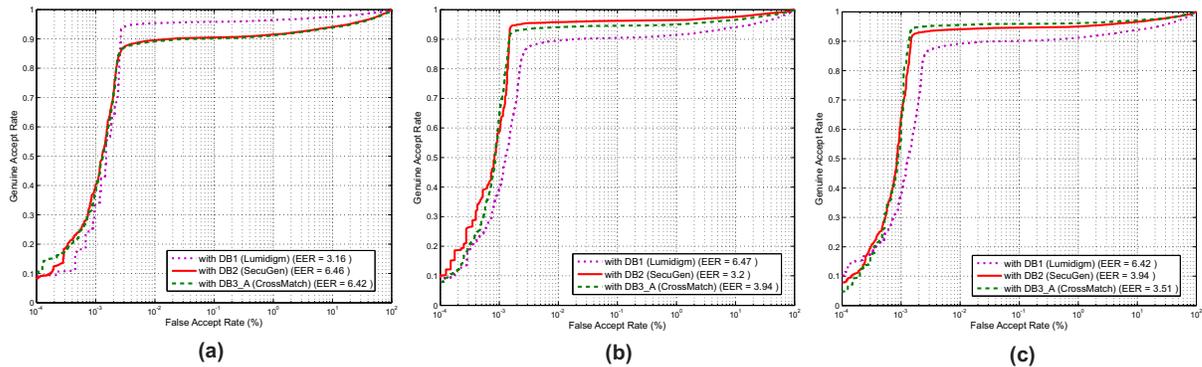


Figure 3-8: ROC curves using VeriFinger for experiment I. (a) *DB1* (Lumidigm) as gallery, (b) *DB2* (Secugen) as gallery, and (c) *DB3_A* (CrossMatch) as gallery. For all three cases, probe is also varied to study the effect of interoperability

mated feature extractor and matcher. The results obtained are in the range of 6 – 11%, which shows that automated feature extraction requires a significant amount of research. Similar to experiment II, the best matching performance is obtained for NBIS matcher while using *DB3_A* (CrossMatch) as gallery.

- After removing the FTP latent prints from *DB4*, the performance improves and the accuracy of experiment III (b) is found to be in the range of 42-56%. It is interesting to note that NBIS shows a very high FTP rate of almost 78% while the FTP rate for VeriFinger is approximately 17%. However, we would like to emphasize that VeriFinger and NBIS are not meant for matching latent fingerprints.

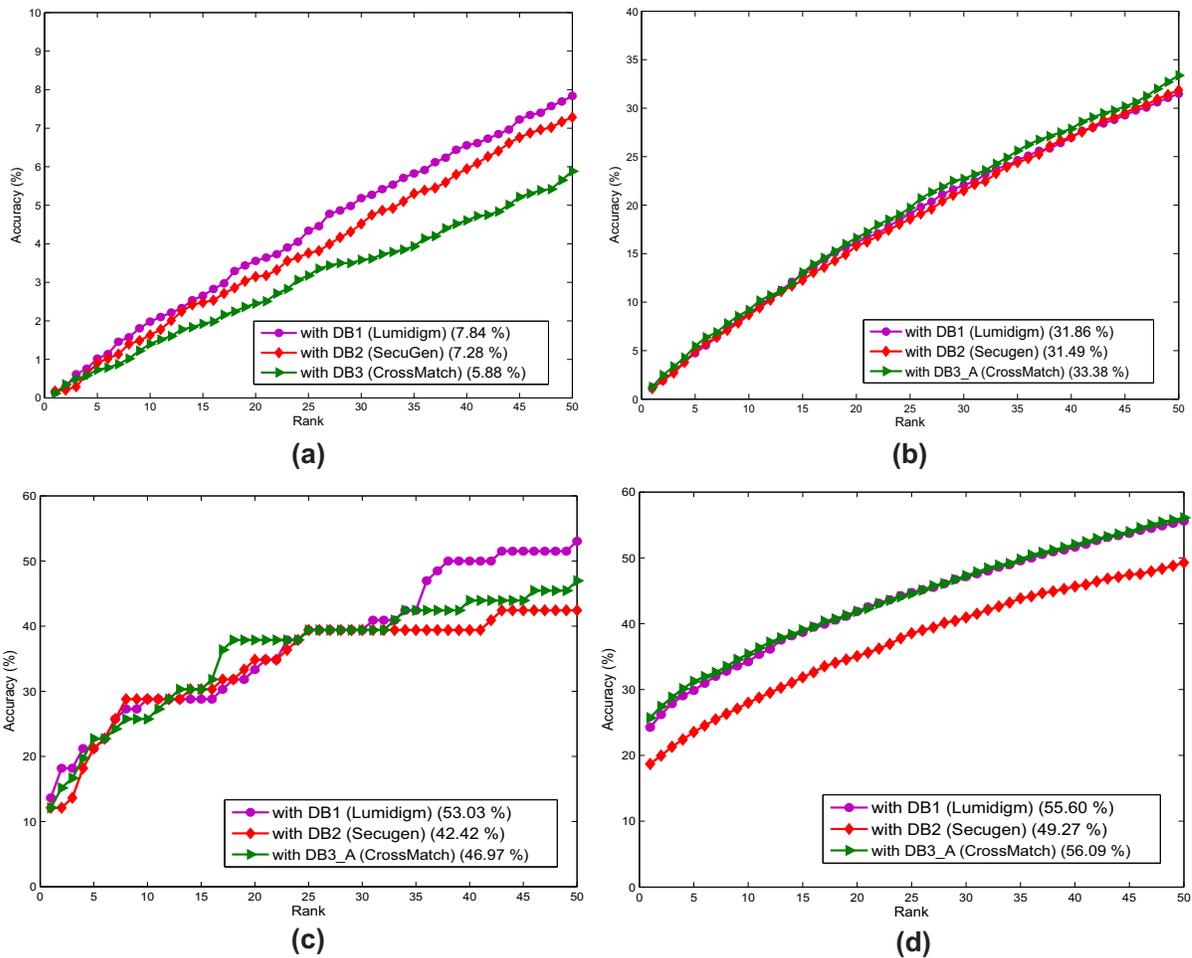


Figure 3-9: CMC curves for experiments II(a), II(b) and III(b). The results are computed with (a) manually marked minutiae matched using Minutiae Cylinder Code, (b) manually marked features matched using BOZORTH3, (c) NBIS, and (d) VeriFinger.

- *nfseg* in NBIS is used to crop slap fingerprints captured using CrossMatch sensor. A segmentation accuracy of 98.4% is obtained for segmenting 1200 slap fingerprints into 4000 individual fingerprints, failing to segment 64 fingerprint images. These images are further manually cropped for our experiments. However, NFSEG fails to perform segmentation in simultaneous prints, segmenting only 134 latent prints from a total of 4400 prints (with $\approx 3\%$ accuracy).

Since there is no automatic algorithm for establishing simultaneity and/or automatic simultaneous latent fingerprint matching, baseline results are not computed for *DB5*.

3.5 IIITD Smartphone Fingerphoto Database v2

In our preliminary research [145], we proposed an initial matching pipeline for fingerphoto images. We created [IIIT-D SmartPhone FingerPhoto Dataset version-1 \(ISPF-D-v1\)](#)³, focusing on background and environmental illumination variations as two challenges for fingerphoto matching. In our experiments, we observed that fingerphoto images captured in the outdoor environment with natural background provided the best verification performance. From the lessons learnt using the previous experiments conducted, we decided to create an extended and improved fingerphoto database. The main motivation for creating the new database are as follows:

1. To study multiple variations in fingerphoto recognition such as inter-sensor matching including varying camera resolution, background variations, and illumination.
2. To study the effect of range of variations caused by varying the surrounding illumination. The indoor illumination is highly controlled using an extra light source, and the outdoor illumination is highly varied by capturing fingerphoto images both in day light and night time (using flash).
3. To create a larger fingerphoto database having at least 300 classes.

Incorporating the above understandings, we present the [ISPF-D-v2](#) database consisting of more than 16000 images obtained from 300 unique fingers. The fingerphoto images are taken using two smartphones: OnePlus One (OPO)⁴ and MicroMax Canvas Knight⁵. Indoor fingerphoto images are captured in both constrained and uncontrolled environments. A special phone holding stand is fixed onto the side of the desk onto which the smartphone is mounted, as shown in [Figure 3-10](#). While the phone is mounted to the stand, a finger is placed on the desk below it, thus ensuring that the distance between the finger and the camera is fixed. An external illumination ring is placed around the mounting portion of the stand to be able control the illumination of the image capture in indoor environment. In outdoor conditions, images are captured without flash during day light and with flash during night. Auto-focus is always kept ON. Based on the challenges considered, the

³Available at <http://iab-rubric.org/resources/spfd.html>

⁴<https://oneplus.net/>

⁵<http://www.micromaxinfo.com/canvasknight/>

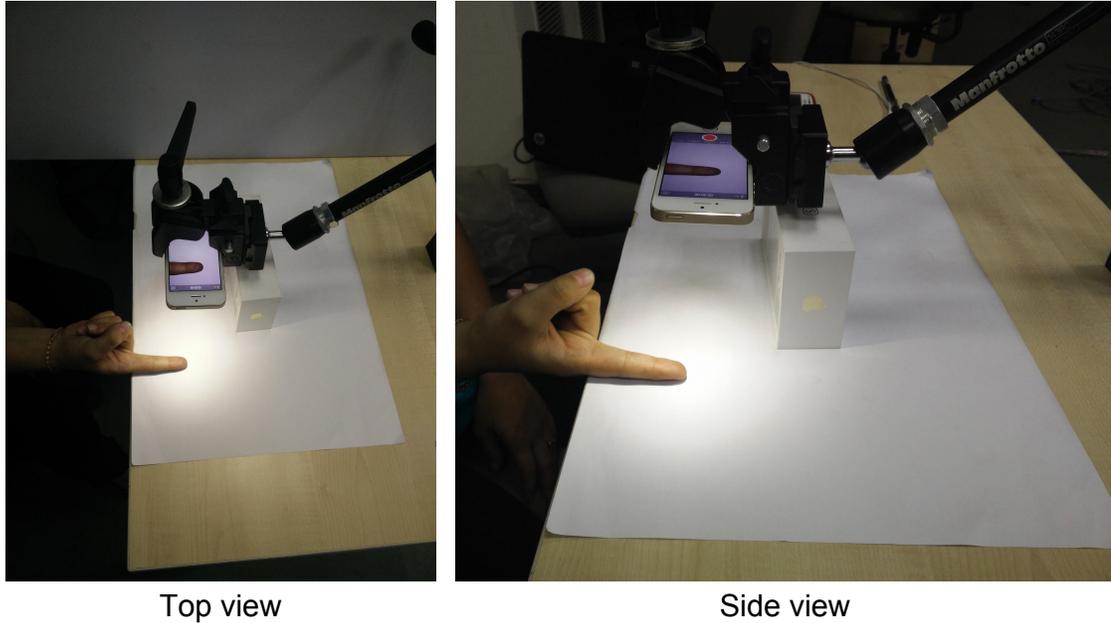


Figure 3-10: The environment and the mount setup used to collect fingerphoto images in the proposed [ISPF-D-v2](#).

Table 3.6: A summary of the multiple subsets and their variations in the [ISPF-D-v2](#).

Set	Challenge	Variations			Classes	Images
		Illumination	Background	Resolution		
Set I	White Indoor	Controlled	White	13MP	300	2400
	White Outdoor	Uncontrolled	White	13MP	300	2400
Set II	Natural Indoor	Controlled	Natural	13MP	300	2400
	Natural Outdoor	Uncontrolled	Natural	13MP	300	2400
Set III	Resolution	Controlled	White	5MP	300	2400
	Resolution	Controlled	White	8MP	300	2400
	Resolution	Controlled	White	16MP	300	2400

database is divided into three subsets and the summary of all three subsets is provided in Table 3.6.

The three subsets are:

- Set I - white background:** Fingerphoto images are captured in both indoor (controlled illumination) and outdoor (with uncontrolled lighting) environment with white background, as shown in Figure 3-11(a) and Figure 3-11(b). The two subsets, White Indoor (*WI*) and White Outdoor (*WO*) show the effect of varying illumination with a constant uniform white background. The images are taken using OnePlus One phone at 13MP resolution. Each



Figure 3-11: Sample images showing various challenges addressed in IIITD SmartPhone Fingerphoto Database v2. (a)-(b) illumination variation with white background, (c)-(d) background variation, and (e) camera resolution variation. Multiple samples showing the intra-class variations and noise present in the database.

subset has 8 images each of right index, right middle, left index, and left middle fingers of 76 subjects. This results in $76 \text{ subjects} \times 4 \text{ fingers} \times 2 \text{ lighting variations} \times 2 \text{ sessions} \times 4 \text{ instances} = 4864 \text{ images}$ for Set I.

- **Set II - natural background:** Fingerphoto images are captured in both indoor and outdoor environment, allowing any natural background to be present, as shown in Figure 3-11(c) and Figure 3-11(d). In real world applications, the natural background available in an indoor environment is very different as the background objects are much closer to the fingerphoto as compared to the outdoor background. Thus, the Natural Indoor (*NI*) subset shows the effect of background variation under controlled illumination while the subset Natural Outdoor (*NO*), shows the effect of background and illumination variations occurring together. The images are captured using OnePlus One phone at 13MP resolution. Similar to Set I, Set II also has *4864 images*.
- **Set III - resolution:** This set consists of fingerphoto images captured in three different resolutions with uniform controlled illumination and white background, as shown in Figure 3-11(e). Two different smartphones, OnePlus One and MicroMax Canvas Knight, are used to capture the images at three different resolutions 5MP, 8MP, and 16MP. Flash LED are turned off while the auto-focus is kept ON. All the images are captured in an indoor lab environment, with uniform lighting and a blank white paper as the background. Under these settings, four instances of the index finger and middle finger of the right and left hand of 76 subjects are captured at all three resolutions. This results in $76 \text{ subjects} \times 4 \text{ fingers} \times 3 \text{ resolutions} \times 2 \text{ sessions} \times 4 \text{ instances} = 7296 \text{ images}$.

Figure 3-11 shows sample fingerphoto images from the proposed database. The database will be made publicly available for academic research at the following link: <http://iab-rubric.org/resources/spfd2.html>.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Adaptive Latent Fingerprint Segmentation

4.1 Introduction

This research work focuses on automating the task of latent fingerprint segmentation. As shown in Figure 4-1, some of the factors involved in making latent fingerprint segmentation a difficult problem are that fingerprints may be of poor ridge clarity [146] or certain latent fingerprints may not have a clear boundary due to smudges and background noise. In this chapter, we propose a feature selection and learning based classification approach for segmenting fingerprint foreground from background. As shown in Figure 4-2, the interleaving ridge-valley patterns and the background are much clearer and distinct in live-scan fingerprints than in latent fingerprints. Therefore,

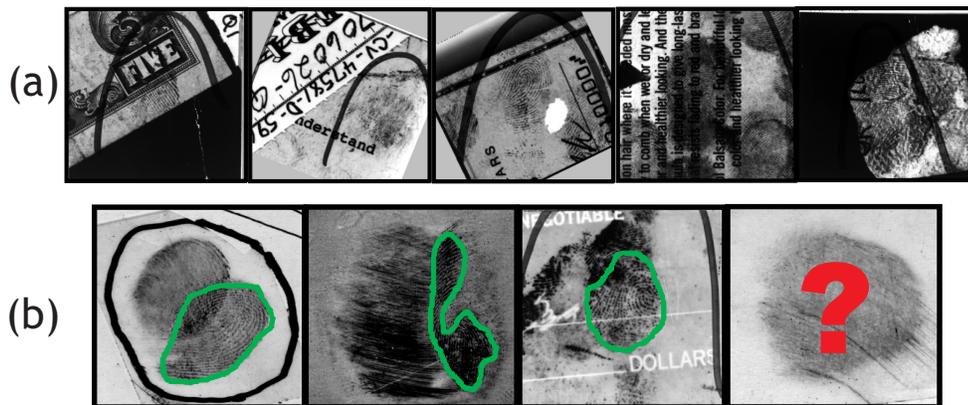


Figure 4-1: (a) Sample latent fingerprint images from the [NIST SD-27](#) database [4] demonstrating the effect of background information on ridge information and (b) latent fingerprint samples illustrating the problem of segmentation.

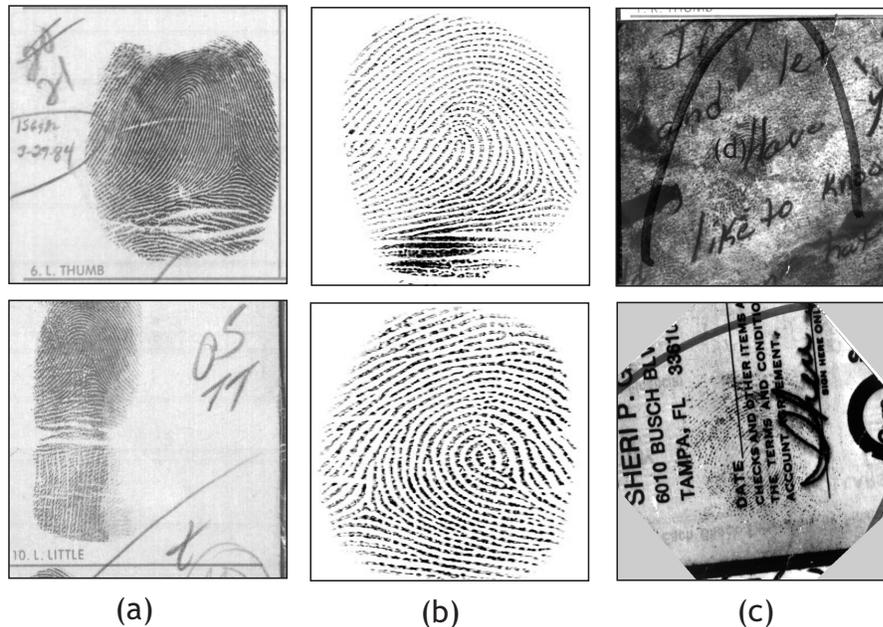


Figure 4-2: Sample latent fingerprints illustrating the distinct nature of foreground ridge patterns and background. (a) inked fingerprints, (b) live-scan fingerprints, and (c) latent fingerprints.

any single feature or a category of features is unlikely to yield proper segmentation results across all kinds of latent fingerprints lifted from different surfaces. Also, to accommodate the variations in the ridge patterns and to make generalized conclusions, the segmentation algorithm needs to select useful features and to learn the difference between background and foreground regions from these features. Inspired from these observations, the proposed approach extracts a composite set of features to represent latent fingerprint ridge patterns, performs feature selection, and classification for improved accuracy. The key research contributions of this research are:

- Latent fingerprint segmentation is modeled as a learning based two-class classification problem with *foreground* and *background* being the two classes. To the best of our knowledge, no classification based segmentation approach has been proposed for latent fingerprints. Though in live-scan fingerprints, there are classification based segmentation approaches [147], [9], [148], [149], the nature of latent fingerprints (with poor ridge features, varying background, overlapping foreground-background information) cause difficulty in applying existing algorithms on these images. The problem is further exacerbated with the availability of small sample size latent fingerprints databases where classes have high intra-class variations. Traditional classifiers such as SVM and Neural Network on such databases may not

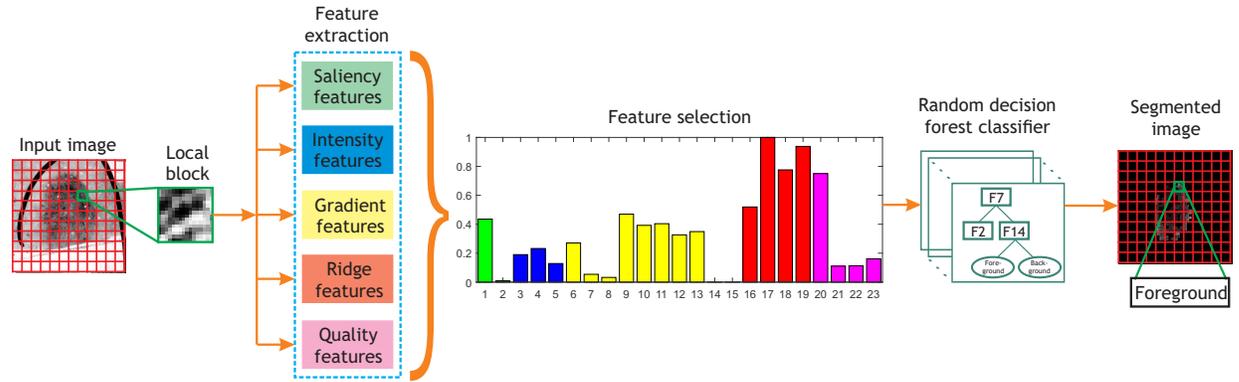


Figure 4-3: Illustrating the steps involved in the proposed **RDF** based latent fingerprint segmentation algorithm. A composite set of features is extracted from every local block and a random decision forest based binary classifier is used to classify the foreground regions from background.

yield good classification results. Therefore, **RDF** classifier, which is an ensemble of multiple decision trees, is utilized to classify the extracted features into the two classes.

- We propose image saliency as a key feature for latent fingerprint **ROI** detection. To the best of our knowledge, saliency has not been used for fingerprint segmentation in literature. It is observed that salient regions of an image contain the foreground ridge information. However, for some instances, the background has very distinct, salient objects other than fingerprints. Therefore, along with saliency based features we combine image intensity based features and fingerprint specific features (gradient based, ridge based, and quality based) to obtain a more robust representation of the fingerprint ridge patterns. We grouped many of the existing features into five categories to perform a more system study on the foreground representation.
- A modified RELIEF formulation is proposed to perform feature selection and study optimal features for fingerprint segmentation which are finally used for classification.
- The performance of the proposed algorithm is evaluated using two different databases: (1) **NIST SD-27** database [4] and (2) **IIITD-CLF** database, which is a combination of IIIT-D latent fingerprint database [137] and **IIITD-SLF** [105]. The segmentation performance is evaluated in terms of multiple metrics: (1) **SIVV** based metric for evaluating the effect of segmentation, (2) segmentation accuracy (SA) which captures the amount of useful information retained after segmentation, and (3) matching accuracy (MA) which captures the

contribution of segmentation process in improving the latent fingerprint matching performance.

4.2 Proposed Segmentation Algorithm

Latent fingerprint segmentation is formulated as a binary classification problem where every local region is classified as either foreground or background. As illustrated in Figure 4-3, the proposed segmentation algorithm consists of local block tessellation, feature extraction, and feature selection followed by RDF based binary classification.

4.2.1 Feature Extraction

Determining whether a local block contains fingerprint patterns requires extracting patterns that are very specific to fingerprints (e.g. ridge patterns). A significant amount of research has undergone in describing ridge patterns or segmenting inked and live-scan fingerprints [147], [9], [148], [149]. The features used for distinguishing fingerprint foreground and background can be classified into five categories namely:

1. **Saliency-based features:** General saliency based features can be used to define the most salient regions in a latent fingerprint image.
2. **Image intensity-based features:** Features such as intensity mean, variance, and ridge cluster degree can be grouped into image intensity based features.
3. **Gradient-based features:** Features such as ridge orientation, variance along and normal to the ridge orientation flow, and symmetric orientation response can be categorized as gradient based features.
4. **Ridge-based features:** Features such as inter-ridge distance, ridge frequency, and ridge angular bandwidth belong to ridge based features.
5. **Quality-based features:** Features such as ridge energy and ridge continuity map measure the quality of local ridge blocks.

Let I be the input latent fingerprint image and $I(i, j)$ denote the intensity at pixel coordinates (i, j) . The image is tessellated into local blocks of size $w \times w$ and five categories of features (mentioned above) are extracted from every local block.

Saliency-based features

In an image, a salient region is defined as the region which is noticed first by a human eye [150]. In general, salient regions are the most informative regions in an image and in the case of fingerprint, it is generally the foreground that we notice first. Therefore, saliency-based features are applied to segment foreground from background. Here, two such features are used, orientation and intensity of the intermediate neighborhood, that help in generating the saliency map of a fingerprint image. As studied by Harol et al. [151], a saliency map gives higher values in the most salient regions i.e. the fingerprint region along with some prominent background regions. Thus the intermediate features, intensity and orientation ($f_1 - f_2$), should give a similar higher response in the foreground as shown in Figure 4-4. The saliency features are computed as follows:

- (i) **Salient Intensity** (f_1): This feature is related to the saliency of a pixel, which is computed as the dissimilarity of the given pixel with respect to its $w \times w$ neighborhood (in terms of image intensity). The dissimilarity measure is weighted by a Gaussian function,

$$f_1 = \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \left| \log \frac{I(x, y)}{I(x+i, y+j)} \right| \cdot \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right) \quad (4.1)$$

where, (x, y) is the center pixel of the local block and σ is a free variable which is assigned the value 0.5. Higher value of dissimilarity in the local block represents that the region is more salient.

- (ii) **Salient Orientation** (f_2): This feature is computed by the summation of the Gabor filter responses along two orientations: 0 degrees and 90 degrees. The orientation feature is calculated as follows:

$$f_2 = \text{abs}(F \bullet G(0)) + \text{abs}(F \bullet G(90)) \quad (4.2)$$

where, F is the Fourier transform output of the local block. $G(0)$ and $G(90)$ are the Gabor

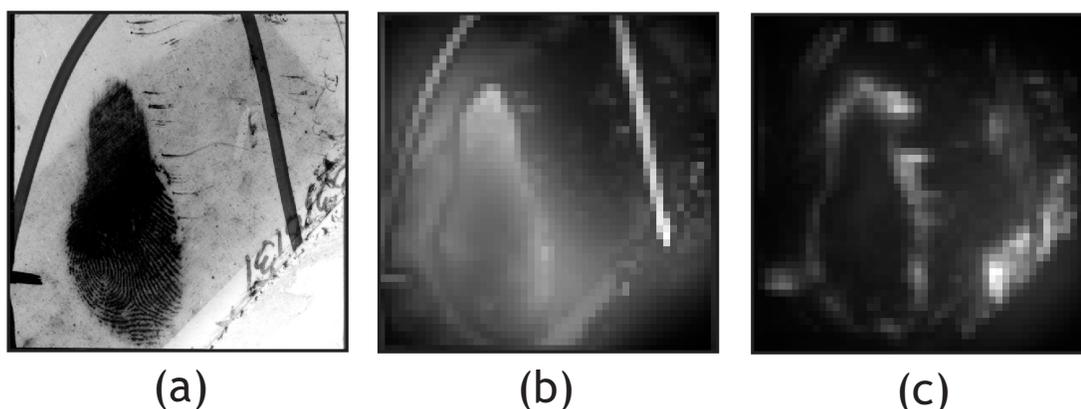


Figure 4-4: (a) A sample image from the [NIST SD-27](#) database with corresponding saliency-based (b) intensity feature (f_1), and (c) orientation feature (f_2).

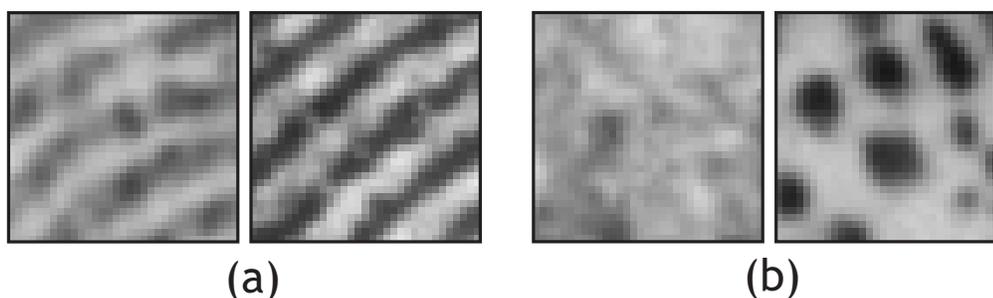


Figure 4-5: Sample local blocks from the [NIST SD-27](#) database with (a) foreground ridge blocks and (b) noisy background blocks. Varying image intensity patterns can be observed between the foreground and the background blocks.

filters along zero degrees and 90 degrees, respectively and $\langle \bullet \rangle$ denotes 2D convolution.

Though saliency extraction algorithm [151] is designed for natural images, we observe that the approach provides useful saliency maps in fingerprints as well, that can be used for segmentation. It is to be noted that saliency of all the pixels are computed and then block-wise features are extracted.

Image intensity-based features

In a latent fingerprint image, the variation in intensity values is usually definite along the ridges and valleys when compared to noisy background regions. Further, as shown in [Figure 4-5](#), the

properties of image intensity in a local region of the foreground is different from the image intensity of the background. These properties are extracted using three different intensity-based features.

- (i) **Difference of means** (f_3) computes the difference between the local intensity mean and the global intensity mean. As a result of varying intensities in the background and foreground, the global intensity mean would be closer to average grayscale value. For a local foreground fingerprint region, due to the interleaved ridge-valley structures, the mean intensity value would be closer to average grayscale value of the image than compared to a background region. Therefore, ideally the difference of means should be lower in the foreground as compared to background.

$$f_3 = \left(\frac{1}{w^2} \sum_{i=1}^w \sum_{j=1}^w I(i, j) \right) - I_{mean} \quad (4.3)$$

where I_{mean} is the mean intensity of the complete image.

- (ii) **Variance** (f_4) calculates the intensity variance in a $w \times w$ image block. Since the variance in an interleaved ridge-valley structure would be higher, high variance is expected in a fingerprint region as compared to background.

$$f_4 = \frac{1}{w^2} \sum_{i=1}^w \sum_{j=1}^w \left(I(i, j) - \frac{1}{w^2} \sum_{i=1}^w \sum_{j=1}^w I(i, j) \right)^2 \quad (4.4)$$

- (iii) **Ridge cluster value** (f_5) indicates the clustering between the ridge pixels. This feature combines the properties of both mean and variance to capture the ridge-valley structure in a fingerprint foreground region. It can be calculated as follows [152]:

$$f_5 = \sum_{i=1}^w \sum_{j=1}^w v_1(i, j) \times v_2(i, j) \quad (4.5)$$

where

$$v_1(i, j) = \begin{cases} 1 & \text{if } I(i, j) < I_{mean} \\ 0 & \text{otherwise,} \end{cases} \quad (4.6)$$

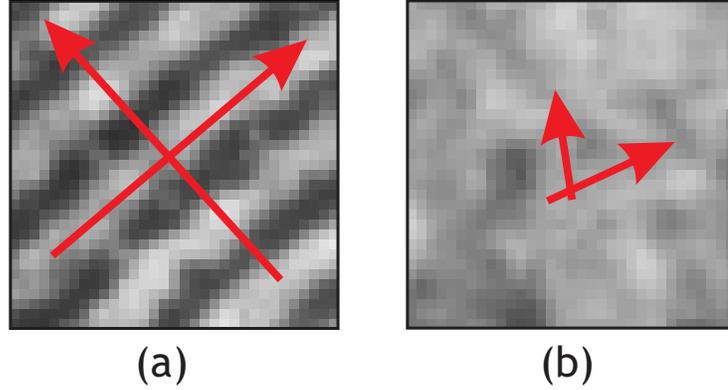


Figure 4-6: Sample local square blocks from the [NIST SD-27](#) database with (a) dominant orthogonal orientation observed in a foreground ridge block, (b) no dominant direction found in a noisy background block.

$$v_2(i, j) = \begin{cases} 1 & \text{if } s(i, j) < \left(\frac{n^2}{2} + 1\right) \\ 0 & \text{otherwise.} \end{cases} \quad (4.7)$$

$$s(i, j) = \sum_{x=i-\frac{n}{2}}^{x=i+\frac{n}{2}} \sum_{y=j-\frac{n}{2}}^{y=j+\frac{n}{2}} v_1(x, y) \quad (4.8)$$

Here, I_{mean} is the global mean intensity and the number of pixels s , in the neighborhood $n \times n$ that have intensity lower than I_{mean} (typically valley regions) is measured. s tends to be larger in uniform background regions than in ridge-valley like regions. Feature f_5 measures the number of pixels in a local block $w \times w$ whose s value is below a specific threshold. Thus f_5 maybe a robust measure even in regions with broken or noisy ridges.

Gradient-based features

The gradient of an image is used to capture the directional change in pixel intensities along a direction. This change in directional flow will be more regular in a fingerprint region as compared to background where the noise gives a non-directional change in the flow [153]. Gradient is also being used to estimate the orientation of ridges in local regions as shown in Figure 4-6. For a latent fingerprint image I , let $[I_x, I_y]$ be the gradient along x and y directions, respectively. The orientation at location (i, j) is calculated as:

$$O(i, j) = \begin{cases} \pi/4 & c_1 = 0, c_2 < 0 \\ 3\pi/4 & c_1 = 0, c_2 \geq 0 \\ \theta^I(i, j) + \pi/2 & c_1 > 0 \\ \theta^I(i, j) & c_1 < 0, c_2 \leq 0 \\ \theta^I(i, j) + \pi & c_1 < 0, c_2 > 0 \end{cases} \quad (4.9)$$

$$\theta^I(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{c_2}{c_1} \right) \quad (4.10)$$

where, c_1 and c_2 are defined as follows:

$$c_1 = \sum_{i=1}^w \sum_{j=1}^w (I_x^2(i, j) - I_y^2(i, j)) \quad (4.11)$$

$$c_2 = \sum_{i=1}^w \sum_{j=1}^w 2 \cdot I_x(i, j) \cdot I_y(i, j) \quad (4.12)$$

The gradient properties are formulated using six different features that are explained below.

- (i) **Ridge orientation** (f_6) is computed by smoothing the orientation over the block using a Gaussian smoothing kernel [154].

$$f_6 = \frac{1}{w^2} \sum_{i=1}^w \sum_{j=1}^w O'(i, j) \quad (4.13)$$

$$O'(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\sin(2O(i, j)) * G(i, j)}{\cos(2O(i, j)) * G(i, j)} \right) \quad (4.14)$$

where, G is the Gaussian smoothing kernel of size 3×3 and $O(i, j)$ is defined in Equation 4.9.

- (ii) **Sum of squared gradient** (f_7) represents the sum of squares of the gradient values of a local block. The interleaving ridge-valley pattern provides a change in flow that is higher as compared to the noisy background.

$$f_7 = \sqrt{c_1^2 + c_2^2} \quad (4.15)$$

(iii) **Sum of norm of squared gradient vector** (f_8) is computed as:

$$f_8 = \sum_{i=1}^w \sum_{j=1}^w \sqrt{\begin{matrix} (I_x^2(i, j) - I_y^2(i, j))^2 + \\ (2 \cdot I_x(i, j) \cdot I_y(i, j))^2 \end{matrix}} \quad (4.16)$$

(iv) **Variance of projected axis parallel to orientation** (f_9) is calculated by computing the ridge variation in the direction parallel to the estimated local block orientation. A projection window of size $B \times H$, which is smaller than the block size and whose center overlaps with center of the local block (k, l) is used to compute f_9 :

$$f_9 = \sum_{l=-B/2}^{B/2} \left(Pv[l] - \sum_{k=-B/2}^{B/2} Pv[k]/B \right)^2 \quad (4.17)$$

where,

$$Pv[k] = \frac{1}{H} \sum_{h=-H/2}^{H/2} I(i - h \cdot \sin(O(i, j)) + k \cdot \cos(O(i, j)), \quad (4.18)$$

$$j + h \cdot \cos(O(i, j)) + k \cdot \sin(O(i, j)))$$

(v) **Variance of projected axis orthogonal to orientation** (f_{10}) is calculated by computing ridge variation in the direction normal to the estimated local block orientation. Similar to f_9 , a projection window of size $B \times H$ perpendicular to the estimated orientation of ridges is considered and the features are computed as follows:

$$f_{10} = \sum_{l=-B/2}^{B/2} \left(Ph[l] - \sum_{k=-B/2}^{B/2} Ph[k]/B \right)^2 \quad (4.19)$$

where,

$$Ph[k] = \frac{1}{H} \sum_{h=-H/2}^{H/2} I(i + h \cdot \cos(O(i, j)) + k \cdot \sin(O(i, j)), \quad (4.20)$$

$$j + h \cdot \sin(O(i, j)) - k \cdot \cos(O(i, j)))$$

As suggested by Zhu et al. [149], a projection window of size 12×9 has been used for calculating both f_9 and f_{10} . In foreground ridge-like regions, f_9 exhibits very low variance whereas f_{10} exhibits high variance. In background regions, f_9 and f_{10} remain almost constant without much variation.

- (vi) **Mean of symmetry and texture patterns ($f_{11} - f_{15}$):** Let the complex representation of an image be denoted as $z = I_x + iI_y$. As shown by Choi et al. [6], the n^{th} order symmetric decomposition of the orientation response of an image block can be computed using:

$$[f_{11}, f_{12}, f_{13}, f_{14}, f_{15}] = \{s_0, s_1, s_{-1}, s_2, s_{-2}\} \quad (4.21)$$

$$s_n = \frac{\langle z, h_n \rangle}{\langle \text{abs}(z), \text{abs}(h_n) \rangle} \quad (4.22)$$

$$h_n = \begin{cases} (x + iy)^n \bullet G & \text{if } n \geq 0 \\ (x + iy)^{|n|} \bullet G & \text{otherwise} \end{cases} \quad (4.23)$$

where, G is the Gaussian filter with $\sigma = 8$ and $\langle \bullet \rangle$ denotes 2D convolution. The orientation response of an image block is decomposed into five symmetric orders providing features $f_{11} - f_{15}$ for $n = 0, \pm 1, \pm 2$ respectively. The peak response for s_0 is obtained in foreground ridge-like regions whereas $s_1, s_{-1}, s_2,$ and s_{-2} give peak response in the background regions [155].

Ridge-based features

As shown in Figure 4-7, a latent fingerprint may contain many ridge like noisy patterns belonging to other fingers in the background. To differentiate the actual fingerprint from such noisy patterns, the properties of ridges are extracted to effectively test the presence of ridge patterns [156]. The

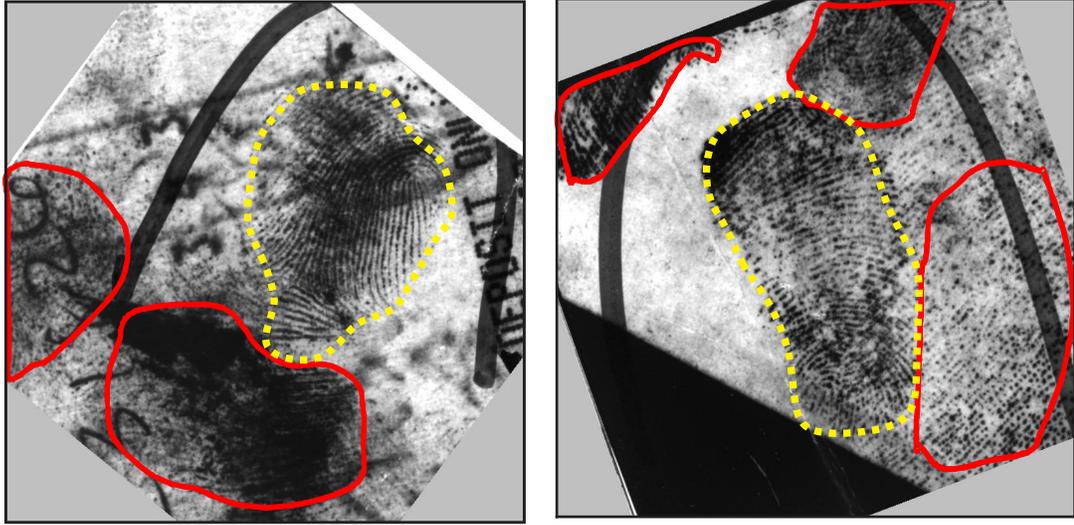


Figure 4-7: Sample images from the [NIST SD-27](#) database showing noise in latent fingerprint images. The yellow dotted lines are the actual fingerprint regions while the red full lines are ridge like noisy regions in the background.

four different features that have been utilized in this research to encode ridge information are:

- (i) **Average inter-ridge distance (f_{16}):** Ridge peaks in the local block are computed using the gradient approach [9]. The mean of absolute difference between any two consecutive peaks is denoted as f_{16} and is computed as follows:

$$f_{16} = \frac{\sum_{k=1}^n a_k}{n - 1} \quad (4.24)$$

where, n is the number of peaks in the ridges and a_k is the distance between two consecutive peak values. As the number of ridges is higher in a fingerprint region, the inter-ridge distance here would be less as compared to background.

- (ii) **Variance of peak heights in ridges (f_{17}):** It estimates the variations in ridge pressure that can be observed in a local block of a latent fingerprint. It can be computed as follows:

$$f_{17} = \frac{\sum_{k=1}^n (PR_k - PR_{mean})}{n - 1} \quad (4.25)$$

where, n is the number of peaks in the ridges, PR_k is the value of the peak ridge height for

the k^{th} ridge, and PR_{mean} is the mean of the peak ridge heights across all the blocks. A higher response is expected in a fingerprint region as compared to a non-fingerprint region.

- (iii) **Ridge frequency** (f_{18}) is calculated by applying Fourier transform to every local block, commonly known as **STFT** [154]. The magnitude spectrum of frequency response is multiplied with a set of directional filters with varying frequencies. The frequency of the filter at which the maximum response is obtained, is considered to be the ridge frequency of the local block.

$$f_{18} = argmax_l \left(\sum_{u=1}^w \sum_{v=1}^w |F(u, v)| * W_l(u, v) \right) \quad (4.26)$$

where, $F(u, v)$ is the Fourier transform output of the local image block and $W_l(u, v)$ is the l^{th} directional filter. Since a ridge-valley structure can be interpreted as a sinusoidal structure, the frequency response is higher in a structured fingerprint region as compared to a noisy and unstructured background region.

- (iv) **Angular bandwidth** (f_{19}): Similar to f_{18} , **STFT** is applied and peak response is calculated for every block. The bandwidth of directional filter along the local estimated orientation that provides the peak response is the angular bandwidth of ridges for the local block.

Quality-based features

Assessing the quality information in a local fingerprint region is very useful for segmentation. The quality of ridge patterns acts as a measure of confidence of the features extracted in the local region. Foreground regions should have a higher quality information compared to the noisy background regions. The quality features are extracted as follows:

- (i) **Ridge energy** (f_{20}): The **STFT** response of a local fingerprint block is subjected to a band-pass filter allowing only the specified ridge frequencies to pass [157]. The ridge energy is computed as follows:

$$f_{20} = \frac{1}{w^2} \left(\sum_{u=1}^w \sum_{v=1}^w (|F(u, v)| \bullet W_{l'}(u, v))^2 \right) \quad (4.27)$$

where, $W_{l'}(u, v)$ is the l' directional filter giving the highest response. The ridge energy

provides the “ridgeness” of the local region and is expected to be higher in a fingerprint region.

- (ii) **Ridge energy after clustering** (f_{21}): The Fourier response of a local block is initially clustered into two regions using k -means clustering and smoothed using a Gaussian filter [158]. Then, similar to f_{20} , energy in a local block is calculated.
- (iii) **Ridge continuity map** (f_{22}): Every local block is modeled with two 2D sine waves, \mathcal{S}_1 and \mathcal{S}_2 , corresponding to the top two local amplitude maxima of ridge intensity [159]. An indicator function is created to check if the waves in consecutive blocks (in a 8-neighborhood condition) are continuous.

$$I_c(\mathcal{S}_1, \mathcal{S}_2) = \begin{cases} 1 & \text{if } \mathcal{S}_1, \mathcal{S}_2 \text{ are continuous} \\ 0 & \text{otherwise} \end{cases} \quad (4.28)$$

The ridge continuity map is then calculated as

$$f_{22} = \sum_{\mathcal{S}'_i \in N} \max\{I_c(\mathcal{S}_1, \mathcal{S}'_1), I_c(\mathcal{S}_1, \mathcal{S}'_2)\} \quad (4.29)$$

where $1 \leq N \leq 8$ and \mathcal{S}'_i are the blocks belonging to the neighborhood N . This ridge continuity measurement gives a higher response in a good quality fingerprint region.

- (iv) **Ridge clarity map** (f_{23}): Ridge clarity map can be calculated by multiplying the peak magnitude value of every local block, a_1 , with the corresponding ridge continuity map value. The response of ridge clarity map is high in a good quality fingerprint region but is robust against background patterns that look similar to ridge-valley patterns.

$$f_{23} = a_1 \cdot f_{22} \quad (4.30)$$

Thus, a composite set of 23 features $\{f_1, f_2, \dots, f_{23}\}$ is utilized for differentiating the foreground ridge patterns from a (noisy) background. A summary of all the category-wise features is provided in Table 4.1. It is our hypothesis that image saliency potentially detects the latent print region in the image along with few other salient regions. Thus, when saliency features are combined with

Table 4.1: Summary of features used to represent the foreground ridge features.

Saliency	Image intensity	Gradient	Ridge	Quality
f_1 Intensity f_2 Orientation	f_3 Difference of mean f_4 Variance f_5 Ridge cluster value	f_6 Ridge orientation f_7 Sum of squared gradient f_8 Sum of norm of squared gradient f_9 Variance of projected axis parallel to orientation f_{10} Variance of projected axis orthogonal to orientation f_{11} Mean of symmetry and f_{15} texture patterns	f_{16} Average inter-ridge distance f_{17} Variance of peak heights in ridges f_{18} Ridge frequency f_{19} Angular bandwidth	f_{20} Ridge energy f_{21} Ridge energy after clustering f_{22} Ridge continuity map f_{23} Ridge clarity map

fingerprint specific features, the false positive regions could be minimized resulting in only the required ROI.

4.2.2 Feature Selection

The proposed algorithm utilizes an aggregation of 23 features. However, not all of them are equally distinctive and can differentiate between foreground and background efficiently. Therefore, in the proposed algorithm, we perform feature selection to select highly discriminative features so that the classification algorithm provides improved (and meaningful) output. The effectiveness of the extracted features is evaluated individually for segmentation. Choosing a subset of relevant features for better performing the task at hand is a challenging research problem [160], [161]. In a binary classification setting, RELIEF [162] is a noise-tolerant, linear time feature selection algorithm that gives good results in the presence of higher training instances. The main advantage of RELIEF feature selection is its simplicity and it does not depend on any heuristics or assumptions. Let W be the weight vector calculating the relevance of each feature i . The standard RELIEF feature selection method is given as follows:

$$W_i = W_i - (X_i - NH_i)^2 + (X_i - NM_i)^2 \quad (4.31)$$

where, X_i refers to the i^{th} training instance, NH_i is the “near-hit” instance of i denoting the nearest neighbor of X_i that belong to the same class of X_i , while NM_i is the “near-miss” instance

denoting the nearest neighbor of X_i belonging to its opposite class. Here, the nearest neighbor is calculated using Euclidean distance measure. It can be understood that the relevance of the weight value reduces if the near-hit of a particular point is at farther distance compared to its near-miss neighbour.

As studied by Robnik-Šikonja and Kononenko [163], RELIEF formulation can be optimized and modified by doing the following:

1. Use ℓ_1 -norm to find the neighbours of X_i instead of using Euclidean distance
2. Calculate the absolute difference between the points, instead of squared difference
3. Choose k -nearest neighbours of an instance X_i instead of the single nearest neighbour.

The modified formulation of RELIEF feature selection used in this experiment is as follows:

$$W_i = W_i - \sum_{p=1}^k |X_i - NH_i^{(p)}| + \sum_{q=1}^k |X_i - NM_i^{(q)}| \quad (4.32)$$

where, $|\cdot|$ represents the absolute difference between the features, NH is the near-hit vector denoting the k -nearest neighbours of X that belong to the same class of X , while NM is the near-miss vector denoting the k -nearest neighbours of X belonging to its opposite class. In our experiments, $k = 20$ is empirically observed to be optimal. A threshold is empirically applied on the weight vector and all the features contributing more than this threshold are considered in the optimal set of features for segmenting latent fingerprints.

4.2.3 Classification using RDF

A non-linear classification algorithm should potentially produce a sophisticated classification boundary between $\{background, foreground\}$ using the extracted feature. In this approach, every local block in a latent fingerprint is classified into foreground and background using Random Decision Forest [164]. **RDF** is a non-linear ensemble classifier consisting of multiple decision trees. It has been shown in literature that **RDF** yields good classification results for high dimensional data [165], [166]. **NIST** uses **RDF** as the classifier in their well-received latent print quality assessment algorithm **NFIQ-2** [71]. The repetitive random sub-sampling strategy employed by **RDF**

helps in providing robust and quicker results for overlapping features. Let N be the total number of data points, M be the number of predictor variables (features), and C be the total number of classes in a given data. A forest containing T trees is trained as follows:

1. For a ratio r ($0.5 < r \leq 1$), several bootstrap aggregates, each of size $r.N$, are created with replacement from the data.
2. Every decision tree, t , in the forest is trained with a single bootstrap of the data, thus creating an ensemble of classifiers.
3. At every node in the decision tree, a random feature sample, m (typically $m = \sqrt{M}$) is used to take the split decision based on an objective function.
4. Class labels c ($c \in \{1, \dots, C\}$) are assigned to leaf nodes depending on the label associated with the corresponding training sample. Collision resolution techniques can be used if a particular leaf node receives multiple class labels through multiple paths.

An input test sample is classified using the trained classifier as follows:

1. The candidate set of features extracted from local blocks of a latent fingerprint are provided as input to the [RDF](#).
2. Every individual decision tree, t , predicts a class label, ot_i , through repeated sub-sampling of features at every node.
3. The final predicted class label, p , is obtained from the ensemble of classifiers using a majority voting technique.

$$p = \underset{c_j \in C}{\operatorname{argmax}} \left[\frac{\sum_{i=1}^T (ot_i == c_j)}{T} \right] \text{ for } j = \{1, 2, \dots, C\}. \quad (4.33)$$

In the RDF implementation, for classifying $C = 2$ classes, $T = 1000$ independent decision trees are created with a bootstrap ratio of $r = 0.66$. At every node in a decision tree, $m = 5$ features are randomly sampled from $M = 23$ features.

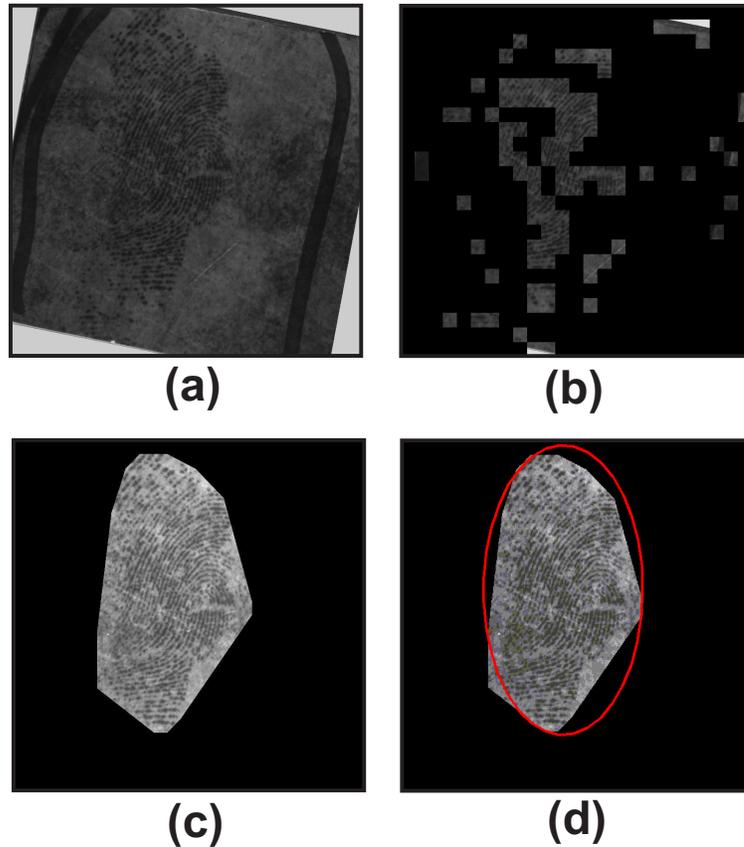


Figure 4-8: An example showing the post-processing performed on the classification output to arrive at the final segmentation output. (a) original input latent fingerprint, (b) classifier output of segmentation, (c) final segmentation boundary obtained after post-processing, and (d) an elliptical window fitted over the segmented region.

4.2.4 Representing Segmented Latent Prints

The output of a segmentation algorithm can take multiple forms. It is important to define a standard representation scheme for segmentation of latent fingerprints. In this research, we represent segmentation as a single n -degree polygonal boundary of the region of interest. The size of the output image is kept same as the input, with all the background information blacked out. Even if the background contains other partial fingerprints (ridge patterns), they are still treated as background and thus removed. To achieve such a standard representation, as shown in Figure 4-8, the following post processing steps are performed for the masks obtained from the classifier output:

- The predicted classifier output contains blocks predicted as foreground or background, as shown in Figure 4-8 (b). Two iterations of morphological erosion are applied using a square

structuring element of size same as the local block size. This helps in removing the false positives, that is, the background blocks that are classified as foreground.

- The largest connected component region in the image is then found using the standard run-length encoding technique [167]. Only the largest connected region is retained while the remaining blocks are regarded as background.
- A convex hull is fitted over the largest connected region, which gives the polygonal mask for the segmented region. The entire region within this boundary is filled as foreground to obtain the processed segmented output of the latent fingerprint, as shown in Figure 4-8 (c).

The processed segmented latent fingerprint acts as a better input for an [AFIS](#) system. It is to be noted that in special applications where the background ridge-like regions are required as well, the post processing stage can be skipped and the output of the classifier can be taken as such.

4.3 Evaluation Metrics for Latent Fingerprint Segmentation

The efficacy of a fingerprint segmentation algorithm is generally evaluated using matching accuracy. However, matching performance does not completely capture the performance of the segmentation algorithm as it includes the performance of other stages such as feature extraction as well. In cases where the ground truth segmentation is available, we propose to use segmentation accuracy (SA) to determine the effectiveness of a segmentation algorithm. Segmentation accuracy is represented in terms of the foreground segmentation accuracy (FSA) and background segmentation accuracy (BSA).

As ground truth of segmentation is not always available in practical situations, there is a need for a metric to evaluate the segmentation algorithm without ground truth. A recent [NIST](#) report [146] and the work by Guan et al. [168] have proposed a new metric for evaluating the effect of preprocessing on latent fingerprints based on [SIVV](#) [169]. SIVV based True-Positive Rate (SIVV-TPR) metric finds the number of correct peaks detected in the 1-D normalized polar transform of the power spectrum of the latent print. A peak does not randomly occur in the frequency spectrum. As proposed in [168], a peak constraint metric is included to search for the peaks in a specific bandwidth in the frequency spectrum, to minimize the detection of false positive peaks.

Thus, this metric could potentially evaluate the performance of the preprocessing stage, without performing the entire matching procedure. The major limitations of this metric are:

- **Shape of ROI:** A rectangular ROI around the latent print region has to be manually chosen by an expert. A tighter boundary estimate, extracted automatically, will provide a better segmentation of latent fingerprints.
- **Shape of Blackman window:** A circular Blackman window filter is applied prior to spectral analysis. Typically, a latent fingerprint is elliptical in shape and thus a circular filter will result in loss of information.

To address these limitations, we have proposed the following modifications to the SIVV metric:

(i) a polygonal ROI is used instead of a rectangular ROI to represent the segmented latent print, and (ii) an elliptical Blackman window is used for filtering instead of a circular filter. Thus, the metrics used to measure the performance of latent fingerprint segmentation are:

1. SIVV based True-Positive Rate (SIVV-TPR) metric [146] is defined as:

$$SIVV - TPR = \frac{\text{Number of correct peaks detected}}{\text{Total number of peaks detected}} \quad (4.34)$$

2. Segmentation accuracy is the ability of a classifier to correctly classify image blocks into foreground and background. It is defined as:

$$SA = \frac{CCB}{TB} \quad (4.35)$$

where, CCB is the number of correctly classified blocks and TB is the total number of blocks. FSA can be calculated as,

$$FSA = \frac{CCFB}{TFB} \quad (4.36)$$

where, $CCFB$ is the number of correctly classified foreground blocks and TFB is the total number of foreground blocks in the ground truth image. Similarly, background segmentation accuracies can be calculated as,

$$BSA = \frac{CCBB}{TBB} \quad (4.37)$$

where, $CCBB$ is the number of correctly classified background blocks and TBB is the total number of background blocks in the ground truth images.

3. Matching accuracy is the fraction of latent images that are correctly identified at a given rank k , when matched with the corresponding gallery of live-scan images.

4.4 Experimental Results

The performance of the proposed segmentation algorithm is evaluated on three publicly available latent fingerprint databases. The algorithm is also analyzed to determine the optimal set of features that would best discriminate the ridge regions from the remaining background. The databases, evaluation metrics, and experimental protocol are described below along with the results.

4.4.1 Datasets

The results are shown on an inked fingerprint database and three publicly available latent fingerprint databases:

- NIST SD-4 database [122] is an inked fingerprint database consisting of 2000 rolled fingerprints pairs having very high quality ridge information with very minimum background variation.
- [NIST SD-27](#) database [4] consists of 258 latent fingerprint images grouped into three quality labels: Good, Bad, and Ugly. It has mated rolled fingerprints for every latent print and also contains manually annotated minutiae for latent fingerprints.
- IIIT-D Latent Fingerprint database [137] has 744 latent impressions from 11 subjects (all 10 fingers) with mated live-scan fingerprints.
- [IIITD-SLF](#) database [105] has 1080 latent impressions from 30 subjects (all 10 fingers) with mated exemplar prints.

Since both the IIITD databases have been collected under similar environments, they are combined to form the [IIITD-CLF](#) database. After combining, it consists of 1824 latent images from 41 subjects with multiple impressions of each finger. In these two sets of latent fingerprint databases,

Table 4.2: Experimental protocol for the NIST SD-4, NIST SD-27, and IIITD-CLF databases. Inked fingerprints from 2000 classes of the NIST SD-4 database are added to extend the gallery.

Database	Training	Testing	Gallery	Image Size	Block Size
NIST SD-4	1000	1000	-	832×768	32×32
NIST SD-27	129	129	258 + 2000	800×768	32×32
IIIT-D CLF	544	1280	820 + 2000	256×400	16×16

NIST SD-27 has real forensic fingerprints with high variation in quality whereas IIITD-CLF database has large number of fingerprints collected in simulated lab environments.

4.4.2 Experimental Protocol

The experimental protocol is shown in Table 4.2. Since the NIST SD-27 database contains only 258 samples, 50% training and 50% testing protocol is followed. For the IIITD-CLF database, a more challenging protocol of using one-third images for training and the remaining for testing is adopted. NIST SD-4 also uses a 50-50% train-test protocol. Due to the variations in image resolution in the databases, inked prints from NIST SD-4 and latent prints from the NIST SD-27 are divided into blocks of size 32×32 while the images from the IIITD-CLF database are divided into 16×16 blocks. The ground truth for segmentation is manually annotated for all the latent prints from both the databases. A n -point contour is marked tangential to the foreground ridge region thus obtaining a binary mask¹. The binary mask is then tessellated into square blocks and ground truth label for each block is assigned. In 2012, Ulery et al. [54] have suggested that since latent examiners use their subjectivity and experience in latent fingerprint analysis, the results tend to vary among human experts and may not be always reproducible. However, it is our assumption that manually annotated segmentation results should not vary significantly across examiners and therefore the performance analysis of the proposed algorithm also should not vary much with the variations in ground truth. To remove any training bias, three times random cross validation is performed on both the databases. The segmentation experiments are performed under the following scenarios:

1. Results of all the features ($f_1 - f_{23}$) are shown using the proposed RDF based classification

¹The manually annotated segmented binary masks will be made publicly available for researchers through the following link: <http://iab-rubric.org/resources.html>

algorithm.

2. Results of only the saliency features ($f_1 - f_2$) are shown using the proposed **RDF** based classification scheme.
3. Results of the selected optimal features (varying size for each dataset) are shown using the proposed **RDF** based classification scheme.
4. The effectiveness of **RDF** is demonstrated by comparing the performance of all the features ($f_1 - f_{23}$) with two other classifiers in literature - neural network and **SVM**. The neural network consists of a three hidden layer architecture with $\{20, 10, 5\}$ nodes each and a single output node with sigmoid activation function. **SVM** (libSVM implementation in MATLAB [170]) with a **Radial Basis Function (RBF)** kernel function ($c = 8, g = 2$) is found optimal.

4.4.3 Importance of Saliency

Feature selection is performed separately on NIST SD-4, **NIST SD-27**, and **IIITD-CLF** databases and the features contributing to better classification are tabulated in Table 4.3. The following important observations can be made based on the feature analysis:

- In all three databases, saliency (f_1) is one of the key features contributing towards segmentation. This aspect is relatively unexplored in the literature of latent fingerprints. If saliency features could be combined with fingerprint based features, a good representation of latent print foreground region can be obtained, as visually demonstrated in Figure 4-9.

Table 4.3: RELIEF algorithm based feature analysis on NIST SD-4, **NIST SD-27**, and **IIITD-CLF** databases. The most and least contributing features for segmentation on each database are also obtained.

Database	Best 3 features	Worst 3 features	Optimal features
NIST SD-4	$\{f_1, f_{14}, f_{15}\}$	$\{f_{22}, f_2, f_{21}\}$	$\{f_1, f_{14}, f_{15}, f_{12}, f_{13}\}_5$
NIST SD-27	$\{f_1, f_6, f_{19}\}$	$\{f_4, f_2, f_{10}\}$	$\{f_1, f_6, f_{19}, f_5, f_{18}, f_{22}\}_6$
IIIT-D CLF	$\{f_4, f_5, f_1\}$	$\{f_3, f_{19}, f_6\}$	$\{f_4, f_5, f_1, f_9, f_{10}, f_8, f_{17}, f_{16}, f_7, f_{23}, f_{22}, f_{20}, f_{14}, f_{12}, f_{15}, f_{13}, f_{21}, f_{18}, f_2, f_{11}\}_{20}$

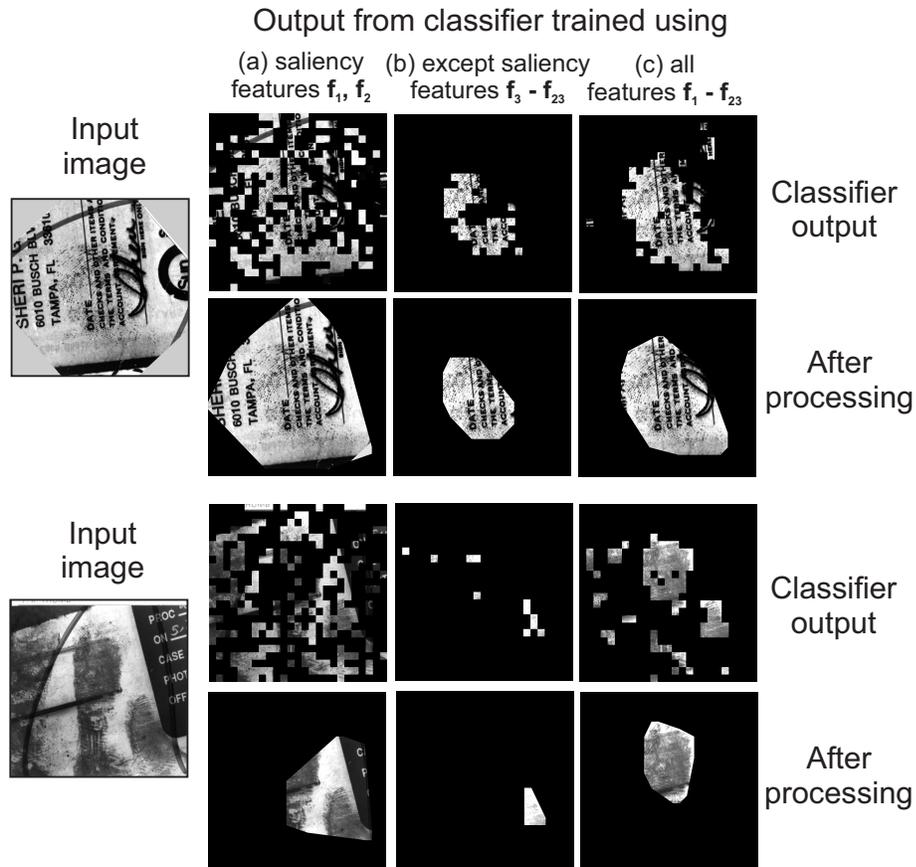


Figure 4-9: Illustrating the segmentation result of the proposed algorithm using two sample images from NIST SD-27 by using (a) only saliency features (f_1, f_2), (b) all other features except saliency ($f_3 - f_{23}$), and (c) all features (f_1, f_{23}).

- Apart from saliency, other features contributing towards ridge representation (best 3 features) are data dependent. This, as expected, explains the variation in ridge clarity in the databases used in this experiment.
- In inked fingerprints (for example images from the NIST SD-4 database), the texture pattern related features are more appropriate for segmentation while ridge continuity and energy based quality features contribute the least.
- In the most popular latent fingerprint database [NIST SD-27](#), it is observed that saliency features and fingerprint specific features contribute more towards segmentation while in the [IIITD-CLF](#) database, saliency features and image intensity based features contribute more for segmentation. Thus, across databases, saliency based features are found to primarily

contribute toward segmentation along with other features that are database specific.

- For a single latent fingerprint from the [NIST SD-27](#) database, the average feature extraction time on a Windows-7 desktop system with Intel i7 processor, 2.86GHz CPU, and 8GB RAM is 15.95ms while for the [IIITD-CLF](#) database it is 10.13ms. The computation time for extracting only the optimal features on the same desktop system is 8.07ms for [NIST SD-27](#) database and 8.38ms for [IIITD-CLF](#) database.

4.4.4 Segmentation Performance

The objective of segmentation is to extract all the foreground regions while discarding the noisy background regions. The results of the proposed segmentation algorithm are shown in [Table 4.4](#) and [Table 4.5](#). As segmentation is the first step in the feature extraction and matching pipeline, the ideal situation is to have high foreground segmentation accuracy (FSA), with not very low background segmentation accuracy (BSA), suggesting that the informative foreground region is minimally lost while allowing some background (noisy) regions. The segmentation accuracy along with FSA and BSA, when compared with the ground truth manual segmentation can be considered as a good estimate of the performance of a segmentation algorithm. When the ground truth of segmentation is not available, SIVV-TPR acts as a robust “as-is” metric to evaluate the performance of latent segmentation without the need for performing matching. Key observations from the segmentation results are as follows:

- [Table 4.4](#) shows high segmentation accuracy of about 96% on inked fingerprints from NIST SD-4 dataset. Also, the accuracy with using only the saliency features is about 78% while using the optimal features is as high as 92%. These results validate the goodness of the proposed algorithm and also the selection of optimal features.
- We analyze the performance of individual category of features on the [NIST SD-27](#) database and observe that saliency features provide the highest foreground segmentation accuracy of 73.4% whereas the FSA of other four features is less than 30%.
- From [Table 4.4](#), it is observed that for latent prints in [NIST SD-27](#) database, the proposed RDF based algorithm (with all features) yields the best FSA of 83.41%. Also, in the [IIITD-](#)

Table 4.4: Segmentation accuracy (SA), FSA and BSA (in %) of the proposed and existing segmentation algorithms on the NIST SD-4, NIST SD-27, and IIIT-D combined latent fingerprint databases.

Database	Metric	All Feat. + SVM	All Feat. + NN	All Feat. + RDF	Saliency Feat. + RDF	Optimal Feat. + RDF
NIST SD-4	SA	91.84±0.2	95.10±0.7	96.11±0.3	78.25±0.2	91.90±0.0
	FSA	92.54±0.3	98.94±0.6	95.76±0.2	79.90±0.7	92.49±0.0
	BSA	91.37±0.2	92.15±0.6	96.35±0.2	77.14±0.5	91.50±0.0
NIST SD-27	SA	66.24±0.4	76.64±0.2	73.76±0.2	61.00±0.1	66.35±0.1
	FSA	78.45±0.1	77.68±0.1	83.41±0.1	73.04±0.1	85.11±0.1
	BSA	63.18±0.5	76.39±0.2	71.34±0.2	57.98±0.1	61.63±0.2
IIIT-D CLF	SA	89.33±0.6	93.47±0.2	93.57±0.2	60.34±0.6	93.47±0.1
	FSA	91.59±0.7	93.01±0.1	93.23±0.2	56.26±0.5	93.01±0.1
	BSA	87.41±0.4	93.84±0.1	93.84±0.3	63.73±0.7	93.84±0.3

Table 4.5: The SIVV-TPR improvement on the three databases, before and after segmentation.

Algorithm	NIST SD-4	NIST SD-27	IIIT-D CLF
Unsegmented	0.9195 ± 0.007	0.3468 ± 0.04	0.4378 ± 0.066
Ground truth	0.9463 ± 0.005	0.5366 ± 0.033	0.5330 ± 0.014
All features + SVM	0.9267 ± 0.006	0.4726 ± 0.069	0.5092 ± 0.063
All features + Neural Network	0.9295 ± 0.009	0.4738 ± 0.047	0.5157 ± 0.045
All features + RDF	0.9325 ± 0.005	0.5168 ± 0.059	0.5512 ± 0.029
Saliency features + RDF	0.9330 ± 0.008	0.4757 ± 0.036	0.5354 ± 0.027
Optimal features + RDF	0.9410 ± 0.006	0.5274 ± 0.034	0.555 ± 0.024

CLF database it is observed that the proposed algorithm yields significantly higher segmentation accuracy of 93.23%. This, in general, highlights the successful adaptive nature of the proposed segmentation algorithm for different qualities of prints.

- In both the NIST SD-27 and IIITD-CLF databases, using only optimal features yield similar segmentation performance as the complete feature set. This shows that the optimal feature set is a comprehensive representation having foreground/background distinguishing capability comparable to the entire feature set. It is interesting to note that depending on the database characteristics, optimal feature sets are different for each database but the salient features are present for all three databases. From the *No free lunch theorem*, it is well understood that the same set of features may not yield best performance across all databases. However, from the implementation perspective, there are two things to note: use of all the features yield the best accuracy and requires 15 ms per test image whereas optimal features

yield similar accuracy in 7 ms. Therefore, if the training database is available for feature selection, then depending on the database characteristics, the optimal feature set can be selected and used. If the training database is not available, then all 23 features can be used for classification.

- From the results of [NIST SD-27](#), it can be observed that using only saliency features (f_1, f_2) provides a FSA of about 73%, while addition of fingerprint specific features improves the FSA to about 83 – 85%. Similar improvements can be observed in [IIITD-CLF](#) and NIST-4 databases, as well. This observation is also visually demonstrated using sample images from [NIST SD-27](#) in Figure 4-9.
- The FSA of all the algorithms is comparatively higher for the [IIITD-CLF](#) database than the [NIST SD-27](#) database. This can be attributed to the fact that the NIST database has real world images with significant amount of background information, whereas the [IIITD-CLF](#) database is prepared in simulated lab environment with very little background noise such as text and lines.
- From Table 4.5, it can be clearly observed in all three databases, that ground truth segmented images show an improved SIVV-TPR rate compared to unsegmented images. This validates the fidelity of the proposed SIVV-TPR metric and also highlights the necessity of segmentation in latent prints.
- The SIVV-TPR metric shows that in all the datasets, using optimal features with [RDF](#) classification and using all features with [RDF](#) classification provides segmentation that is comparable to the ground truth segmentation. Thus, automatic segmentation of latent prints using the proposed algorithm provides as good segmentation as ground truth, for both kind of fingerprints (inked or different qualities of latent prints).

4.4.5 Comparison with Existing Feature Selection Algorithms

The modified RELIEF algorithm is compared with some popular feature selection algorithms [171] available in literature: (i) [Max-Relevance Min-Redundancy \(MRMR\)](#) algorithm, (ii) [Joint Mutual Information \(JMI\)](#) algorithm, (iii) [Double Input Symmetrical Relevance \(DISR\)](#) algorithm, and

Table 4.6: Segmentation accuracy (SA), foreground segmentation accuracy (FSA), and background segmentation accuracy (BSA) (in %) of the proposed and existing feature selection algorithms on the NIST SD-4, [NIST SD-27](#), and [IIITD-CLF](#) databases.

Database	Metric	MRMR	JMI	DISR	RELIEF	Modified RELIEF
NIST SD-4	FSA	89.18 \pm 0.0	89.13 \pm 0.0	89.16 \pm 0.0	88.93 \pm 0.0	92.49 \pm 0.0
	BSA	82.27 \pm 0.0	82.12 \pm 0.0	82.37 \pm 0.0	82.51 \pm 0.0	91.50 \pm 0.0
NIST SD-27	FSA	39.45 \pm 0.0	45.33 \pm 0.0	45.33 \pm 0.0	43.87 \pm 0.0	85.11 \pm 0.1
	BSA	95.19 \pm 0.0	93.76 \pm 0.0	93.76 \pm 0.0	94.71 \pm 0.0	61.63 \pm 0.2
IIIT-D CLF	FSA	96.98 \pm 0.0	96.98 \pm 0.0	96.97 \pm 0.0	96.28 \pm 0.0	93.01 \pm 0.1
	BSA	85.27 \pm 0.0	85.27 \pm 0.0	84.37 \pm 0.0	83.14 \pm 0.0	93.84 \pm 0.3

(iv) RELIEF algorithm. These algorithms are individually used to select the optimal features for each of the databases. Next, an [RDF](#) classifier is trained using the optimal features selected using various feature selection algorithms.

The performance of the feature selection algorithms is compared in terms of foreground and background segmentation accuracies and is shown in Table 4.6. Of all the feature selection algorithms compared in Table 4.6, it can be observed that the proposed modified RELIEF algorithm provides a good trade-off between the overall segmentation accuracy and foreground segmentation accuracy. Further analysis into the optimal features selected by different algorithms reveal that saliency features are assigned significantly high weight in modified RELIEF. In all other algorithms, either saliency does not occur in the list of optimal features or is assigned lower weight. The advantage of the modified RELIEF feature selection algorithm is that it works better in binary classification setting with continuous features. Hence, it can be deduced that modified RELIEF algorithm is the most suitable feature selection algorithm.

4.4.6 Comparison with Existing Latent Print Segmentation Algorithm

The proposed segmentation algorithm is compared with existing segmentation algorithm proposed by Zhang et al. [7]. In order to compare the results and follow the experimental protocol discussed in Section 4.2, we obtained the binary masks for all the images in [NIST SD-27](#) database from the authors of [7] and split it according to three cross validation sets. Figure 4-10 shows the comparison of the two algorithms in terms of segmentation accuracy on the [NIST SD-27](#) dataset.

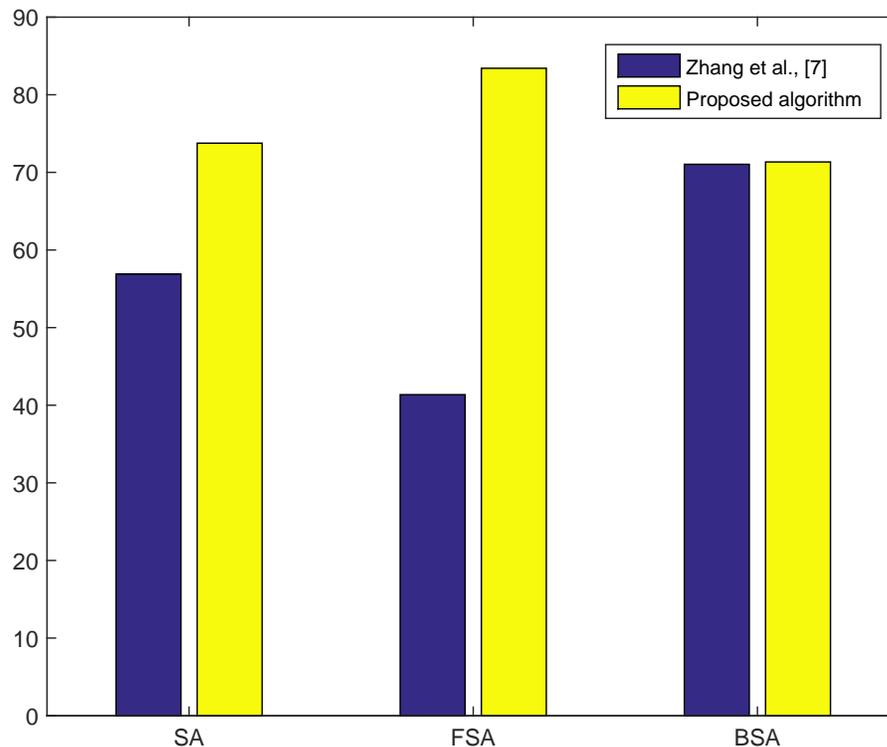


Figure 4-10: Comparing the performance of the proposed segmentation algorithm with the algorithm recently proposed by Zhang et al. [7], on the [NIST SD-27](#) dataset.

As compared to Zhang et al. [7], the proposed algorithm gives similar background segmentation accuracy but it yields an improvement of about 18% in the overall segmentation accuracy due to high foreground accuracy.

4.4.7 Matching Performance

The final objective of segmenting latent fingerprints from the background is to improve the matching performance. Therefore, the performance of the proposed segmentation algorithm is also evaluated in terms of fingerprint matching accuracy after segmentation on both the latent fingerprint databases². For the [NIST SD-27](#) database, manually annotated minutiae (available along with the database) are used while minutiae for the [IITD-CLF](#) database are automatically extracted using VeriFinger [SDK](#). For segmented images, only the subset minutiae lying within the segmented mask are considered for matching.

²The proposed algorithm is compared with the manual ground truth of segmentation.

Matching latent fingerprints is a challenging research problem and there is no standard open source latent fingerprint matching [SDK](#) or commercial system freely (or low cost) available in public domain. It is observed from the literature that local [MCC](#) [142, 143] description for the minutiae provides state-of-the-art results [85] on latent print matching. Therefore, latent fingerprint matching results are shown using the [MCC](#) descriptors. For both the databases, images of 2000 subjects from the NIST SD-4 database [122] are appended to extend the gallery and three times random split based cross validation is performed. The performance is reported in terms of the rank-50 identification accuracy and the results are reported in [Table 4.7](#) and [Figures 4-11](#) and [4-12](#). Some key results obtained are as follows:

- Using ground truth segmentation, on the [NIST SD-27](#), rank-50 identification accuracy of 83% is observed, which is significantly greater than the accuracy obtained with the unsegmented images (56%). It can be observed that both the variants of the proposed algorithm: using all features and using only optimal features, performs comparable to the ground truth segmentation with almost 80% rank-50 matching accuracy. On the NIST SD-27 database, the matching performance is not reduced much by using only the optimal set of features. Wilcoxon's rank-sum test between the results obtained from all the features and the optimal features, accepts the null hypothesis at 5% significance, claiming there is not much statistical difference between the results obtained.
- On the [IIITD-CLF](#) database, a similar trend can be observed, where the matching performance of the proposed segmented images is almost as good as the ground truth segmented images. The performance is low due to the poor feature extraction by VeriFinger, which is fine-tuned for processing tenprints.
- The number of minutiae preserved by each algorithm after segmentation is provided in [Table 4.8](#). It can be observed that, in both the databases, the percentage of minutiae preserved by the segmentation algorithm is proportional to its corresponding matching performance. A higher number of minutiae is found in the [IIITD-CLF](#) database as they are automatically extracted using a ten-print matcher, while in [NIST SD-27](#) manually annotated minutiae are used.

Table 4.7: Rank-50 identification accuracy (in %) of the proposed segmentation algorithms using MCC descriptor. Manually marked minutiae are used for [NIST SD-27](#) while VeriFinger SDK is used to extract minutiae from [IIITD-CLF](#) database.

Algorithm	NIST SD-27	IIIT-D CLF
Unsegmented	55.9 ± 2.7	26.9 ± 2.7
Ground truth	83.1 ± 7.3	34.2 ± 3.6
All features + RDF	80.0 ± 6.3	33.4 ± 3.0
Saliency features + RDF	66.9 ± 6.5	26.6 ± 0.5
Optimal features + RDF	78.7 ± 6.4	29.6 ± 3.9

Table 4.8: Average number of minutiae extracted in the fingerprint images after segmentation.

Algorithm	NIST SD-27	IIIT-D CLF
Ground truth	19.2	31.9
All features + RDF	19.2	33.3
Saliency features + RDF	15.5	31.1
Optimal features + RDF	18.7	33.3

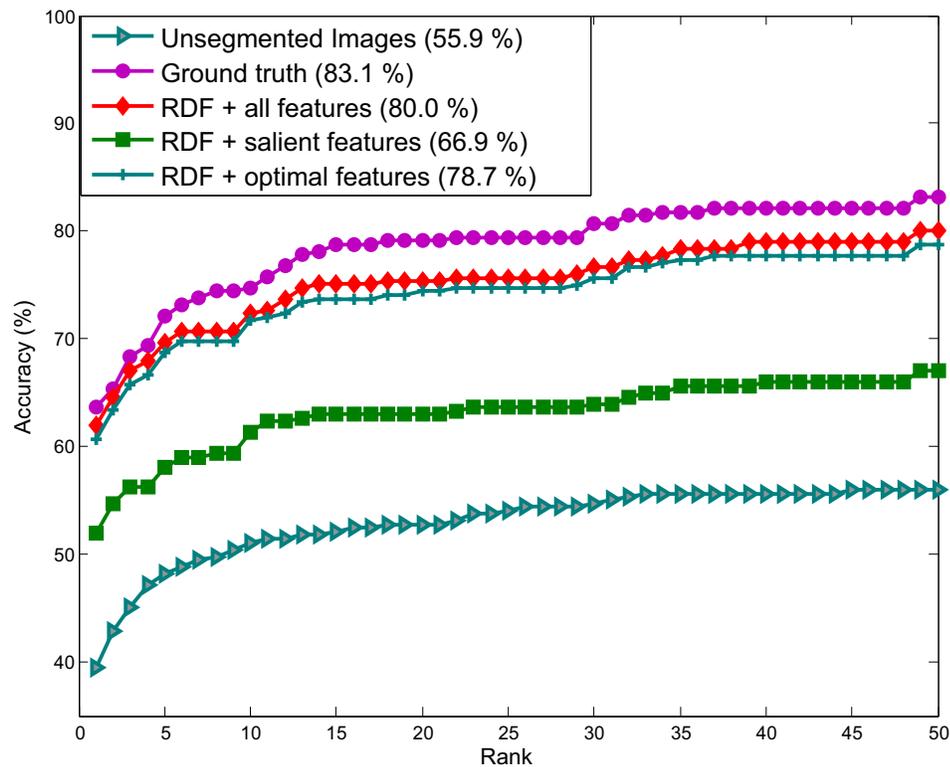


Figure 4-11: CMC curves showing the average matching performance of unsegmented and segmented images on the [NIST SD-27](#) database.

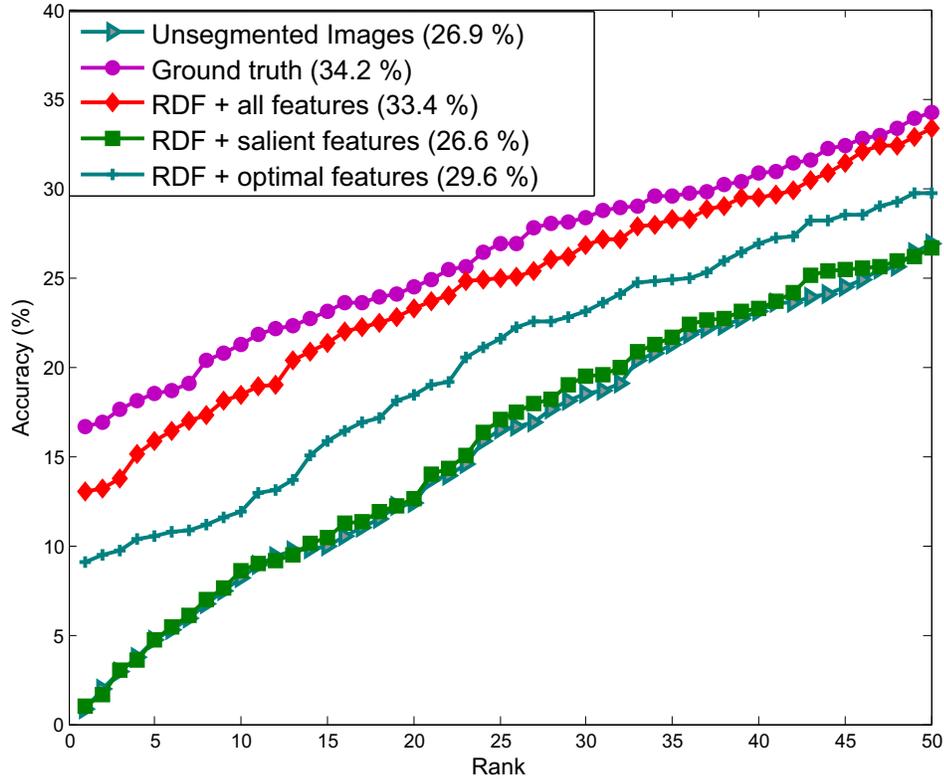


Figure 4-12: CMC curves showing the average matching performance of unsegmented and segmented images on the [IIITD-CLF](#) database.

4.4.8 Performance Evaluation using Latent Fingerprint Identification System with Very Large Gallery

We also compute the effectiveness of the proposed segmentation algorithm using a popular latent fingerprint identification system used by law enforcement agencies³. The system has over 2 million pre-enrolled identities in the database and is modular in nature. Experiments are performed on the [IIITD-CLF](#) database. NIST databases are not used as they may have been used to train the system a priori. First, the gallery images for [IIITD-CLF](#) are enrolled (after the experiments, these enrollments are deleted from the system) and then 1280 probe images are used for evaluation. Two sets of experiments are performed: (1) using the default setting of the latent fingerprint system which uses inbuilt segmentation algorithm and (ii) when segmented outputs obtained from the proposed algorithm are given as input for matching. Rank-50 accuracies obtained for both the experiments are 71.4% and 72.3% respectively. This shows that the proposed segmentation

³The license agreement does not allow us to name the commercial system in any kind of comparison.

algorithms improves the identification performance of a latent fingerprint system on a very large gallery database.

4.5 Summary

Latent fingerprints collected as a forensic evidence are affected by background noise, limited content, and varying quality. As the first step in the recognition pipeline, latent fingerprint segmentation plays an important role. In this research, we proposed a novel latent print segmentation algorithm that extracts saliency, image, gradient, ridge, and quality features from local patches of the image. These features determine the characteristics of both foreground ridge and background noise. An optimal set of features are selected using modified RELIEF based feature selection algorithm and a Random Decision Forest classifier is used to learn *foreground* and *background* regions. Further, a n -degree polynomial representation of the segmented region is found to be the most optimal representation of the segmented results. The performance of the proposed algorithm is evaluated on the basis of three metrics: SIVV-TPR, segmentation accuracy (along with FSA and BSA), and rank- k identification accuracy. The results show that the proposed segmentation algorithm yields high segmentation performance on the NIST SD-4 inked print database and [NIST SD-27](#) and [IIITD-CLF](#) latent databases, showing that the algorithm is able to segment the regions of interest from the background. Using the automatically segmented images, we have observed improved matching performance, which further supports the effectiveness of the segmentation algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Latent Fingerprint Minutiae Extraction

5.1 Introduction

The primary step for realizing a “lights-out” [IAFIS](#) system with minimum or no human interference, is to perform automated feature (minutiae) extraction. Despite the huge amount of research being done in latent fingerprint matching, feature extraction is still a challenge because of the following reasons and as shown in [Figure 5-1](#),

- Smudges and strokes introduced by chemical reagents or brush adds to the noise and information loss during latent fingerprint lifting.
- The surface from which the latent fingerprint is lifted adds to the background noise, thereby making detection of ridge flow challenging.

It can be observed from [Figure 5-2\(a\)](#) that the local region around a minutia has a different ridge structure than a non-minutia patch. However, as shown in [Figure 5-2\(b\)](#), latent fingerprint minutia patches lack a definite structure, making it challenging to learn meaningful information. Owing to the non-uniform and uncertain variations in latent fingerprints, it has been challenging for researchers to define a model for extracting minutiae. Human engineered features such as gradient information and frequency based information, provide limited performance due to the presence of background noise. Therefore, a feature descriptor can be learnt that could describe and differentiate a minutia patch from a non-minutia patch. The key idea in this research is to learn a descriptor for the neighbourhood of a minutiae, ie., the difference in the structure of patches having and not

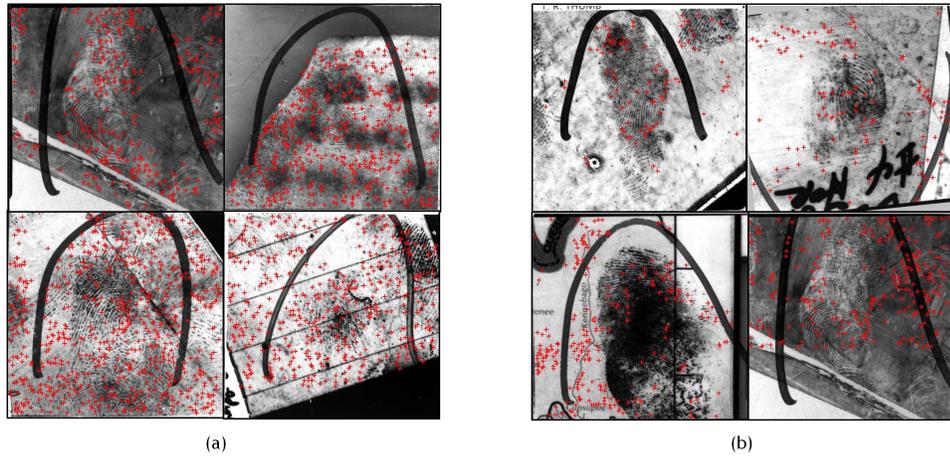


Figure 5-1: Sample latent fingerprints from [NIST SD-27](#) database [4] showing spurious minutiae extracted by (a) [NBIS](#) and (b) [VeriFinger 6.0 SDK](#)

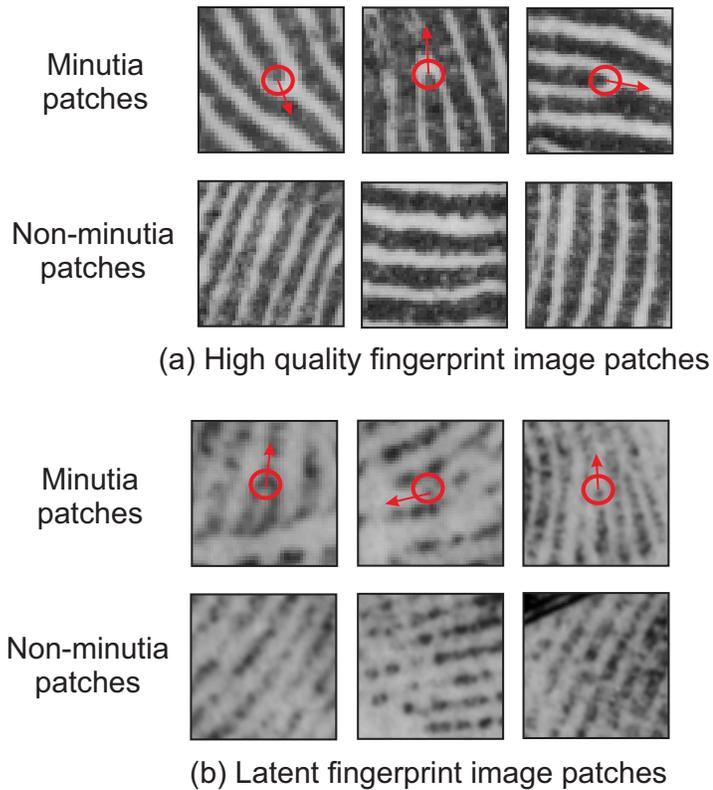


Figure 5-2: (a) High quality fingerprint patches illustrating the difference in the ridge structure between minugia and non-minugia patches, (b) Local patches from latent fingerprints illustrating the lack of well defined structures and noisy ridge patterns.

having a minugia. Adopting from the literature of Unsupervised Feature Learning, deep learning algorithms have been used to learn complex feature representations of the input data [172].

The popular formulations of deep learning algorithms such as an autoencoder or a [RBM](#) can be used to learn latent fingerprint patch descriptor in an unsupervised fashion. However, it is well known in the machine learning community that supervised feature extraction usually leads to better classification [173]. For instance, incorporating Fisher criterion for determining projections in subspace methods reduces the intra-class variability and increases the inter-class variability, thereby increasing discrimination capabilities. Motivated by this observation, in this research, we proposed supervised regularization for autoencoder and [RBM](#) called [GSAE](#) and [Class Sparsity Specific Restricted Boltzmann Machine \(cssRBM\)](#). The primary contributions of this research are as follows:

- Propose a supervised $\ell_{2,1}$ -norm regularization method for autoencoder called [GSAE](#) and derive a solution using majorization-minimization approach [174],
- Propose a [cssRBM](#), which makes use of class information to introduce within-group sparsity using $\ell_{2,1}$ -norm regularization. Construct deep networks, [cssDBM](#) and [cssDBN](#) using [cssRBM](#),
- A novel fingerprint patch descriptor learning algorithm using [GSAE](#), [cssDBM](#) and [cssDBN](#), individually,
- A binary classification model using 2ν -[SVM](#) for detecting minutiae in latent fingerprint, using the learnt patch descriptor, and
- Perform extensive testing and analysis of the proposed minutiae extractor on two public latent fingerprint databases, namely [NIST SD-27](#) [4] and [MOLF](#) [175].

5.2 Regularization in Deep Learning Algorithms

Deep learning architectures consist of multiple hidden layers with non-linear activation functions which enables them to learn a complex relationship between the input and the output. However, the deep architecture also implies the presence of a large number of hyperparameters to be optimized. At times, these parameters are strictly optimized for the training data distribution, thereby

restricting the usability for unseen test data which leads to a familiar problem of overfitting. Regularization is a common technique to address overfitting by introducing an additional term in the objective function that better guides the learning process. With increase in the number of parameters to learn and with increased complexity of the objective functions in deep learning architecture, there is a constant requirement of more efficient regularization methods to improve the learning capacity.

The overfitting resolution techniques employed on network-based-architectures aim at achieving one or both of these goals (i) avoid peaking of weights by adding penalty or normalizing the weights and (ii) introducing sparsity to the learned weights to avoid learning “noisy” patterns [176]. Generally, the additional regularization terms added to the loss function helps in achieving these objectives. Table 5.1 and Figure 5-3 gives an overview of the different kinds of regularization adopted in network based deep learning literature. These techniques can be broadly grouped as follows:

Early Stopping

To perform early stopping, a validation data set of known class labels is required apart from the train set and test set. The process of training is stopped when the error on the validation set starts increasing, while the train error may still be decreasing. This denotes the point where the classifier begins to overfit for the train data. Although the notion of early stopping is promising, stopping criteria is still indefinite. Lutz et al. [177] discussed 14 different criteria to avoid overfitting in network based architectures such as high generalization loss, increase in the validation error in consecutive iterations, and ratio of average training error to minimum training error over an iteration.

ℓ_2 -norm, Rectifiers, Max-norm and Maxout Regularization

These are the most common methods adopted to avoid overfitting. In ℓ_2 regularization [178], squared magnitude of all the parameters, $\frac{1}{2}\lambda w^2$, is added as a penalty to the optimization function. Here, λ is the regularization parameter that controls the amount of penalty to be applied on each parameter. The ℓ_2 regularizer (also called as ℓ_2 -norm weight decay) heavily penalizes peaked weight values maintaining the balance in the parameters. In max-norm regularization [179], the incoming weight vector on every hidden unit is constrained such that $\|w^2\| < c$. Here, c is a

Table 5.1: A comparison of different types of regularization methods used in the literature for network based architectures

Technique	Regularization	Working
ℓ_2 norm [178]	$\frac{1}{2}\lambda w^2$	Clips peak weights
Rectifiers [180]	$\ln(1 + e^y)$	Non-saturating activation function
Max-norm [179]	$\ w^2\ < c$	Clips peak weights
ℓ_1 norm [178]	$\lambda w $	Sparse weight matrix
KL-divergence [182]	$\sum_{j=1}^h KL(\hat{y}_j y)$	Average activation of hidden node is sparse
SGRBM [183]	ℓ_1/ℓ_2	Group level sparsity at hidden unit level
Dropout [179]	$z_i = w_i(r^{(l)} * y) + b_i$	Random sparse nodes at every level
Dropconnect [184]	$z_i = (w_i * r^{(l)})y + b_i$	Random sparse weights between levels

tunable hyperparameter that can be adjusted based on the learning data. This regularization method projects \mathbf{w} on a circle of radius c whenever it goes beyond that, making sure that the norm of any weight vector does not go beyond c . Rectifiers [180] apply a non-linear function, such as smooth softmax function, to the output of hidden units to avoid saturation of output. Maxout networks [181] avoids activation saturation by taking only the maximum of all activations while ignoring the remaining.

Sparsity based Regularization

In ℓ_1 -norm [178] regularization, a term $\lambda\|w\|_1$ is added to the optimization function to make sure that the learnt weight vector is sparse. KL-divergence based sparsity [182] limits the average activations of hidden units while minimizing the divergence between the produced and expected output. Other recent sparsity introduction methods are dropout [179] and dropconnect [184] networks. These are stochastic regularization techniques, generally used to prevent overfitting. In dropout networks, a certain percentage ($p \times 100$) of randomly selected visible and hidden nodes are dropped by making their activation zero. To compensate for the thinned networks obtained during training, while fine-tuning the weights of the entire network are increased by a factor of $1/p$. Dropconnect is a generalized version of dropout, where instead of removing the node, connections between the nodes are randomly removed.

Adding Training Noise

In case of limited variety in training data, the deep learning architectures might just function as

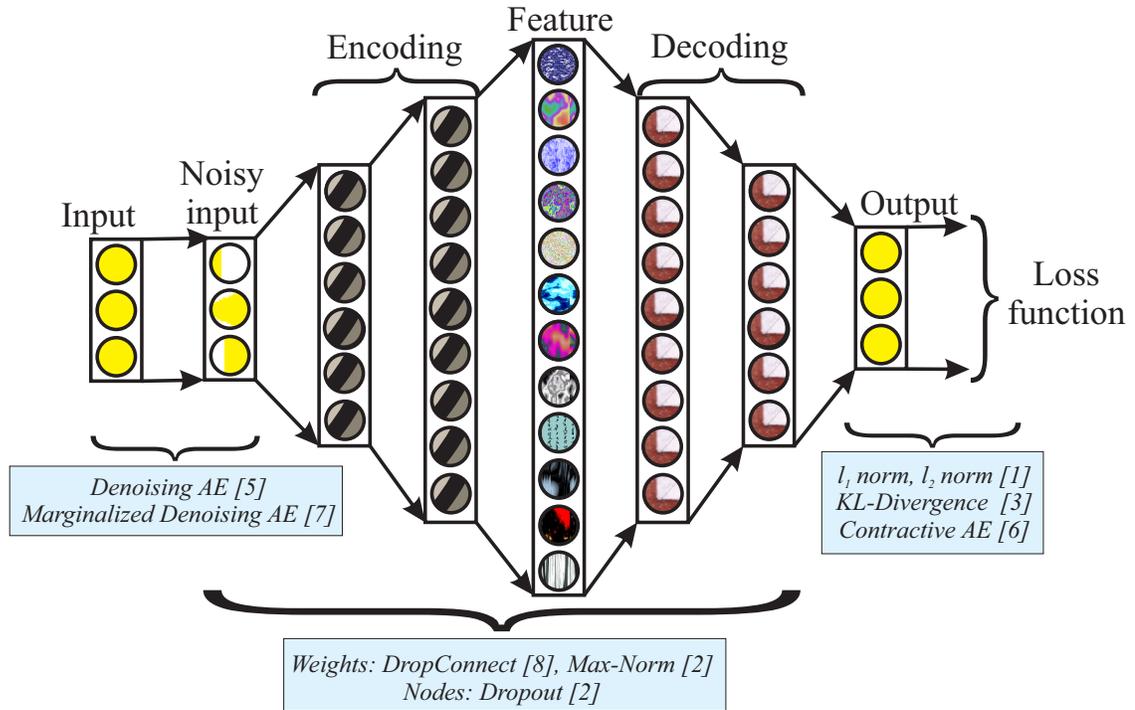


Figure 5-3: An overview of different kinds of regularization techniques used in the literature for an network based deep architecture. Note that the number of nodes in each hidden later is varying and may be dense or sparse depending on the application, data characteristics, and architecture.

a linear identity function, highly overfitting for the training data. Adding noise to the training data and training denoising architectures improves the generalization capability of deep learning algorithms. Vincent et al. [185] proposed denoising autoencoder that attempts to reconstruct the original image from a noisy image using stacked denoising autoencoder algorithm.

Hybrid Regularization

Generally, one or many of these regularizers are used together, thereby complementing their properties to achieve better learning; for example $\ell_2 + \ell_1$ -norm towards the end of training, Dropout+ ℓ_2 -norm, Dropout+max-norm. Further, group techniques such as *group lasso* (ℓ_1/ℓ_2 regularization) have also been explored in literature [183]. While ℓ_1/ℓ_2 regularization [183] succeeds in introducing sparsity at group (class) level, it is important to introduce sparsity within a class as well.

5.3 GSAE: Group Sparse AutoEncoders

Autoencoders are generally unsupervised in nature and leverage the availability of large unlabelled data for feature representation. However, if large amount of labelled data is available, the standard formulation of autoencoder needs to be updated to incorporate the labelled information. Sang et al. [186] recently proposed a supervised loss function in which they optimize the squared loss and the classification loss, simultaneously. Gao et al. [187] proposed a supervised deep autoencoder for face recognition by reducing the loss between a probe image and its corresponding gallery image. In this research, we propose a novel approach for modeling stacked sparse autoencoder that preserves group sparsity. The learning is performed such that the features of a single class will have the same sparsity signature. In other words, the non-zero values in the features occur at the same positions for a class. This is achieved by incorporating $\ell_{2,1}$ -norm regularization [188], [189], [190].

The description of the proposed algorithm starts with the explanation of an autoencoder. Let X be the input data, where

$$X = \left[\underbrace{\{x_{1,1} \dots x_{1,n_1}\}}_{X_1=\text{class 1}}, \dots, \underbrace{\{x_{C,1} \dots x_{C,n_c}\}}_{X_C=\text{class C}} \right] \quad (5.1)$$

Here, C is the number of classes, $\{n_1, n_2, \dots, n_C\}$ are the number of data points in each of the C classes, and the data is organized such that all the data columns belonging to class 1 appear first, followed by data columns of class 2, and so on till data columns of class C . To learn a single layer generative autoencoder model, the loss function J is defined as:

$$J(W) = \arg \min_{W,U} [\|X - U\phi(WX)\|_2^2 + \lambda R(W)] \quad (5.2)$$

where, ϕ is a non-linear activation function such as sigmoid function, W and U are the encoding and decoding weights, respectively. Higher order representation can be learnt by stacking multiple layers together and training them in a greedy layer wise fashion. $R(W)$ can be any regularization function, controlled by the parameter λ , to avoid overfitting by introducing some additional constraints while learning the weight matrix. Some popular regularization functions are (i) LASSO or

the ℓ_1 -norm enforces sparse learning of weights, (ii) Euclidean or the ℓ_2 -norm adds higher penalty to the peak weights, thereby enforcing diffused learning of weights, and (iii) Elastic net or $(\ell_1 + \ell_2)$ -norm adds both the norms in the optimization function. In the proposed group-sparse autoencoder framework, we introduce a $\ell_{2,1}$ -norm based regularization as follows:

$$J(W) = \arg \min_{W,U} [\|X - U\phi(WX)\|_2^2 + \lambda \sum_{c=1}^C \|WX_c\|_{2,1}] \quad (5.3)$$

where, $\|\bullet\|_{2,1} = \sum_j \|Z^{j\rightarrow}\|_2$ is the sum of ℓ_2 -norms of the rows (indicated by j). The inner ℓ_2 -norm promotes a dense (non-zero) solution within the selected rows, however the outer ℓ_1 -norm (sum) enforces sparsity in selecting the rows. In this proposed formulation, the regularizer enforces group sparsity within each class by adding the constraint that the features from the same group/class should have the similar sparsity signature. Note that, $\phi(\bullet)$ is only a clipping function applied term-by-term. Therefore, the second term of $\ell_{2,1}$ -norm can be applied to both $\phi(WX)$ or just WX as both promote row-sparsity. This makes the optimization supervised as the information regarding the class labels is utilized during training. However, we are not enforcing any discriminative property to the features as we are not enforcing features from different groups to have different sparsity signatures.

5.3.1 Solution using Majorization-Minimization

The objective function in Equation 5.3 is a non-convex optimization problem that can be solved using alternating minimization. At any k^{th} iteration, the solution for the non-convex problem can be split into two steps as follows:

$$\begin{aligned} \text{Step 1 : } U &= \arg \min_U \|X - U\phi(W_{(k-1)}X)\|_2^2 \\ \text{Step 2 : } W &= \arg \min_W \left[\|X - U_{(k)}\phi(WX)\|_2^2 + \lambda \sum_{c=1}^C \|WX_c\|_{2,1} \right] \end{aligned} \quad (5.4)$$

Step 1 is a linear least squares regression problem having a closed form solution. Step 2 is challenging, hence, we adopt the Majorization-Minimization [174] algorithm to solve it. In this approach, let $J(W)$ be the function to be minimized. For the initial point w_0 , a smooth function $G_0(W)$ is constructed through w_0 which has a higher value than $J(W)$ for all values of w apart

from w_0 , at which the values are the same. This is the Majorization step where a smooth function $G_0(W)$ is constructed which is easy to minimize. Iteratively at each step, $G_k(W)$ is minimized to obtain the next iteration x_{k+1} . It can be understood that the solution at every iteration gets closer to the actual solution. For mathematical convenience, Step 2 can be rewritten as,

$$\arg \min_Z \left[\| X - U_{(k)}\phi(Z) \|_2^2 + \lambda \sum_{c=1}^C \|Z_c\|_{2,1} \right] \quad (5.5)$$

where, $Z_c = WX_c$ and Z is obtained by stacking the Z_c 's in column. In this optimization problem, only the least square regression term has to be majorized and the penalty term is not affected. During the minimization step, the surrogate majorizer function, $G_k(Z)$ of the actual loss function $J(W)$ is chosen as follows,

$$\begin{aligned} G_k(Z) = & \| X - U_{(k)}\phi(Z) \|_2^2 + \lambda \sum_{c=1}^C \|Z_c\|_{2,1} \\ & + (\phi(Z) - \phi(Z)_{(k)})^T (aI - U_{(k)}^T U_{(k)}) (\phi(Z) - \phi(Z)_{(k)}) \end{aligned} \quad (5.6)$$

Here, a is the maximum eigenvalue of the matrix $U_{(k)}^T U_{(k)}$ and I is the identity matrix. By simplifying $G_k(Z)$, we obtain,

$$\begin{aligned} G_k(Z) = & X^T X - 2X^T U_{(k)}\phi(Z) + \phi(Z)^T U_{(k)}^T U_{(k)}\phi(Z) \\ & + (\phi(Z) - \phi(Z)_{(k)})^T (aI - U_{(k)}^T U_{(k)}) (\phi(Z) - \phi(Z)_{(k)}) \\ & + \lambda \sum_{c=1}^C \|Z_c\|_{2,1} \end{aligned} \quad (5.7)$$

$$\begin{aligned} \implies G_k(Z) = & X^T X + \phi(Z)_{(k)}^T (aI - U_{(k)}^T U_{(k)})\phi(Z) \\ & - 2 [X^T U_{(k)} + \phi(Z)_{(k)}^T (aI - U_{(k)}^T U_{(k)})] \phi(Z) \\ & + a\phi(Z)^T \phi(Z) + \lambda \sum_{c=1}^C \|Z_c\|_{2,1} \end{aligned} \quad (5.8)$$

Let $B = \phi(Z)_{(k)}^T + \frac{1}{a}U_{(k)}^T(X^T - U_{(k)}\phi(Z)_{(k)}^T)$, Equation 5.8 can be written as

$$G_k(Z) = a(-2B^T\phi(Z) + \phi(Z)^T\phi(Z)) + \lambda \sum_{c=1}^C \|Z_c\|_{2,1} + \mathcal{E} \quad (5.9)$$

where, \mathcal{E} consists of constant terms. Using the identity, $\|B - \phi(Z)\|_2^2 = B^TB - 2B^T\phi(Z) + \phi(Z)^T\phi(Z)$, Equation 5.9 can be rewritten as

$$G_k(Z) = a \left(\|B - \phi(Z)\|_2^2 + \frac{\lambda}{a} \sum_{c=1}^C \|Z_c\|_{2,1} \right) - aB^TB + \mathcal{E} \quad (5.10)$$

Removing the constant terms and re-writing in terms of W , the optimization function can be written as,

$$\arg \min_W \left(\|B^T - \phi(WX)^T\|_2^2 + \frac{\lambda}{a} \sum_{c=1}^C \|(WX_c)^T\|_{2,1} \right) \quad (5.11)$$

All the matrices are written in terms of transpose, as the activation function is computed element-wise. Blumensath [191] has shown that it is possible to replace the above non-linear problem, into a simple linear problem using one step of gradient descent, as follows:

$$\arg \min_W \left(\|P - W^T\|_2^2 + \frac{\lambda}{a} \sum_{c=1}^C \|X_c^T W^T\|_{2,1} \right) \quad (5.12)$$

where, $P = W_{(k)}^T - \sigma \nabla \|B^T - \phi(WX)^T\|_2^2 \Big|_{W_{(k)}}$, σ is the step size for gradient descent and can be found using Lipschitz bound. Summation can be removed by redefining Equation 5.12 as follows,

$$\arg \min_W \left(\|P - W^T\|_2^2 + \frac{\lambda}{a} \|VW^T\|_{2,1} \right) \quad (5.13)$$

where, V is defined as the block row concatenation of X_c^T 's. Taking the derivative of Equation 5.13 and setting it to zero, we obtain,

$$2P - 2W^T + \frac{\lambda}{a} V^T D V W^T = 0 \quad (5.14)$$

where, $D = \text{diag}(|VW^T|^{-1})$

$$\left(I + \frac{\lambda}{2a} V^T D V \right) W^T = P \quad (5.15)$$

Using matrix inversion lemma,

$$\begin{aligned} \left(I + \frac{\lambda}{2a} V^T D V \right)^{-1} &= I - V^T \left(\frac{2a}{\lambda} D^{-1} + V^T V \right)^{-1} V \\ \Rightarrow W^T &= P - V^T \left(\frac{2a}{\lambda} D^{-1} + V^T V \right)^{-1} V P \end{aligned} \quad (5.16)$$

If $T = \left(\frac{2a}{\lambda} D^{-1} + V^T V \right)^{-1} V P$, then Equation 5.16 becomes $W^T = P - V^T T$. The solution for T is as follows,

$$\begin{aligned} \left(\frac{2a}{\lambda} D^{-1} + V^T V \right) W^T &= V P \\ \Rightarrow T &= \left(\frac{2a}{\lambda} D^{-1} + C I \right)^{-1} (c T_{(k-1)} + V(P - V^T T_{(k-1)})) \end{aligned} \quad (5.17)$$

is obtained by adding cT on both sides of equation and subtracting with $V^T V W^T$. c is the maximum eigenvalue of $V^T V$. The complete algorithm is summarized as follows:

Initialize: W, V

For every iteration:

Step 1 : $U_{(k)} = \arg \min_U \| X - U \phi(WX) \|_2^2$

Step 2 : $B = \phi(Z)_{(k)}^T + \frac{1}{a} P^T (X^T - U_{(k)} \phi(Z)_{(k)}^T)$

Step 3 : $P = W_{(k)}^T - \sigma \nabla \| B^T - \phi(WX)^T \|_2^2 \Big|_{W_{(k)}}$

Step 4 : $T = \left(\frac{2a}{\lambda} D^{-1} + C I \right)^{-1} (c T_{(k-1)} + V(P - V^T T_{(k-1)}))$

Step 5 : $W = P^T - T^T V$

Another approach for solving the non-convex optimization problem is Alternating Direction Method of Multipliers [192]. However, this approach introduces a lot of hyper-parameters, that require fine-tuning. It can be understood that the proposed approach utilizes only the regularization constant λ as the parameter. The remaining parameters such as a , c , and σ can be computed and fixed.

5.4 Class Sparsity based Restricted Boltzmann Machine

The proposed **cssRBM** is an extension of the generative model of the RBM which incorporates both unlabeled and labeled data. Given a set of training data containing both unlabeled and labeled data points, the **cssRBM** simultaneously learns two objectives as part of its optimization: (a) optimally reconstruct the input data to minimize reconstruction error and (b) learn discriminative features for each class while maintaining sparsity. While objective (a) is achieved primarily using the unlabeled data points, objective (b) is achieved using the labeled data points [193]. These **cssRBM** units are then combined with dropout and dropconnect regularization techniques to achieve robustness against overfitting and the extracted features are utilized with a 2ν -SVM classifier to obtain the final class label. In this section, we briefly discuss the details of **RBM** followed by presenting the formulation of the proposed **cssRBM**.

5.4.1 Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machines are undirected models that use stochastic hidden units to model the distribution over the stochastic visible units [194]. The hidden layer is symmetrically connected with the visible unit and the architecture is “restricted” as there are no connections between units of the same layer. Traditionally, RBM is used to model the distribution of the input data $p(\mathbf{v})$; however, it can be used to model the joint distribution between the input data and the target classes, $p(y, \mathbf{v})$. Let the hidden layer $\mathbf{h} = (h_1, h_2, \dots, h_n)$ consist of n nodes and the visible layer be $\mathbf{v} = (v_1, v_2, \dots, v_d)$, where d is the dimensionality of input data. Since the number of hidden nodes are fixed in this architecture, RBM becomes a parametric model. Let $D = \{(x_i, y_i)\}$ be the training data where x_i represents a data point belonging to class $y_i \in \{1, 2, \dots, C\}$. As presented by Larochelle and Bengio [195], the distribution modeled by RBM can be represented as

$$p(y, \mathbf{v}, \mathbf{h}) = e^{-E(y, \mathbf{v}, \mathbf{h})} \quad (5.18)$$

$$E(y, \mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{h}^T \mathbf{U} \mathbf{y} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h} - \mathbf{d}^T \mathbf{y} \quad (5.19)$$

where, $\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ are the parameters of the model and $\mathbf{y} = (1_{y=i})_{i=1}^C$. To obtain the

least energy model, it is necessary to update the weight matrices \mathbf{W} and \mathbf{U} in such a way that the hidden layer models the joint distribution between the input variables and the target classes. The conditional distributions of \mathbf{v} , y , and \mathbf{h} are given as follows,

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^d p(v_i|\mathbf{h}) \quad (5.20)$$

$$p(y|\mathbf{h}) = \frac{e^{d_y + \sum_{j=1}^h U_{jy} h_j}}{\sum_{y^*} e^{d_{y^*} + \sum_{j=1}^h U_{jy^*} h_j}} \quad (5.21)$$

$$p(\mathbf{h}|y, \mathbf{v}) = \prod_{i=1}^n p(h_i|\mathbf{v}) \quad (5.22)$$

where, the conditional distribution on y is modeled using a softmax function and the conditional distribution on \mathbf{h} depends on both y and \mathbf{v} . Assuming binary input variable, the probability that a node will be active can be given as follows,

$$p(v_i = 1|\mathbf{h}) = \text{sigm} \left(b_i + \sum_{j=1}^n W_{ji} h_j \right) \quad (5.23)$$

$$p(h_j = 1|y, \mathbf{v}) = \text{sigm} \left(c_j + U_{jy} + \sum_{i=1}^d W_{ji} v_i \right) \quad (5.24)$$

Although the conditional distributions are shown for binary valued input variables, a similar model can be easily generated for categorical and continuous valued input variables as well. The objective function for training an RBM is to minimize the negative log-likelihood over the entire training data [196] given as

$$\mathcal{L}(\mathcal{D}, \theta) = - \sum_{i=1}^{|\mathcal{D}|} \log p(y_i, v_i) \quad (5.25)$$

Computing the exact gradient of this loss function is almost intractable. However, there is a stochastic approximation to approximate the gradient, popularly known as the contrastive divergence gradient. A sequence of Gibbs sampling based reconstruction produces an approximation of the expectation of joint energy distribution, using which the gradient can be computed.

5.4.2 Class Sparsity Signature in RBM

Consider an RBM learning a joint distribution function as $p(y, \mathbf{v}, \mathbf{h})$, where y is the set of classes, \mathbf{v} and \mathbf{h} are the set of visible and hidden nodes, respectively. For all the training instances belonging to a specific class y_i , let \mathcal{H}_i be the set of hidden nodes being activated. \mathcal{H}_i is a matrix where the j^{th} row corresponds to the binary hidden layer representation of the j^{th} training sample belonging to class i . The $\ell_{2,1}$ -norm regularization is defined as

$$\sum_{i \in y} \|\mathcal{H}_i\|_{2,1} = \sum_{i \in y} \left[\sum_{j=1}^{|D_i|} \left(\sum_{k=1}^{|h|} f_{i,j,k}^2 \right)^{\frac{1}{2}} \right] \quad (5.26)$$

where, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, |D_i|$ iterates over the training samples belonging to class i , and $k = 1, 2, \dots, |h|$, where $|h|$ is the total number of nodes in the hidden layer. $f_{i,j,k} = p(h_k = 1 | y_i, x^{(j)})$ denotes the probability of the hidden node h_k being activated when provided with an input sample $x^{(j)}$ belonging to class y_i . As the matrix is row-sparse, the outer ℓ_1 -norm promotes sparse selection of rows (specific data points) while the inner ℓ_2 -norm chooses important (peak) features within each data point. $\ell_{2,1}$ has been extensively used in signal processing community to explore joint and class based sparsity [189].

Let \mathcal{L} be the loss function of the RBM that is to be optimized. The overall regularized objective function can be written as,

$$\arg \min_{\theta} \{ \mathcal{L} + \lambda \sum_{i \in y} \|\mathcal{H}_i\|_{2,1} \} \quad (5.27)$$

where, $\theta = \{W, U, b_1, b_2, b_3\}$ is the set of weights and biases for the RBM. This cost function is not smooth throughout, it has discontinuities and therefore cannot be solved using straightforward gradient descent. We can only solve an approximation of the actual objective function which is differentiable. Hence, we modify the overall function using [Iterative Reweighted Least Squares \(IRLS\)](#) technique. [IRLS](#) is a technique to find the maximum likelihood estimate of a generalized linear regression problem. [IRLS](#) is effective in solving the least square regression problem by mitigating the effect of outliers in the data. It is an iterative approach which solves a weighted least square regression at each step, thereby iteratively updating the parameters of the regression model. The non-differentiable $\ell_{2,1}$ norm is approximated using a weighted ℓ_2 norm which can be

easily trained using the standard contrastive divergence algorithm. Thus, the regularization factor can be reformulated as:

$$\|\mathcal{H}_i\|_{2,1} = \|c_i \mathcal{H}_i\|_2 \quad (5.28)$$

where, c_i is the set of weights associated with each class. Cotter et al. [197] proposed a simple and effective technique for calculating the value of c for the problem of joint sparse multiple measurement vector recovery as follows,

$$c_i = \text{diag} \left(\|\mathcal{H}_i^{(j)}\|_2^{\frac{1}{2}} \right) \quad (5.29)$$

where, $\text{diag}(x)$ is a function that returns a diagonal matrix with elements of vector x in the leading diagonal. In Equation 5.29, we are taking the row-wise ℓ_2 norm of H matrix to create a vector followed by creating the diagonal matrix from them. Rewriting Equation 5.27 using weighted ℓ_2 norm formulation, we get,

$$\arg \min_{\theta} \{ \mathcal{L} + \lambda \sum_{i \in y} \|c_i \mathcal{H}_i\|_2 \} \quad (5.30)$$

Algorithm 1 Training update for [cssRBM](#)

Data: Training data D as pairs (x_i, y_i) , learning rate λ

Initialize: Initialize W, U, b_1, b_2, b_3 , and c_i

▷ perform N iterations

for $n : 1 \rightarrow N$ **do**

 solve: $\arg \min_{\theta} \{ \mathcal{L} + \lambda \sum_{i \in y} \|c_i \mathcal{H}_i\|_2 \}$

 ▷ repeat until convergence

end for

Update: c_i

The resulting function is smooth and differentiable at all points in the space, ensuring optimization using contrastive divergence algorithm. A succinct representation of the proposed algorithm is shown in Algorithm 1. Each step in the algorithm has a closed form solution. However, the weights are dependent on the variable itself and hence keep changing after every iteration. Thus, the solution needs to be updated till some convergence is reached.

The [IRLS](#) approach is simple but slow, one needs to solve the full cost function iteratively

when the weights are updated. The sub-gradient based approach and proximity operators can also be used to solve the proposed optimization problem directly (as is done in signal processing literature); but such an approach would require deriving the algorithm from scratch - one could not use the time tested contrastive divergence technique. Moreover, to the best of our knowledge, the proximity operators do not function optimally with generative models such as the RBM.

5.4.3 Constructing Deep Networks using cssRBMs

As mentioned previously, a single RBM does not have sufficient modeling capacity for complex tasks. In order to progressively learn more complex functions of the input, a deep network architecture can be constructed using cssRBM units as well. In this section, we present the construction of cssDBN and cssDBM via the proposed cssRBM.

Constructing cssDBM

Deep Boltzmann Machine (DBM) [198] is an extension of RBM by stacking multiple hidden layers on top of each other. DBM is an undirected learning model and thus it is different from the other stacked network architectures in which each layer receives feedback from both the top-down and bottom-up layer signals. This feedback mechanism helps in managing the uncertainty in learning models. While the traditional RBM can model logistic units, a Gaussian-Bernoulli RBM can be also used with real valued visible units [199]. A two hidden layer Gaussian-Bernoulli DBM can be modeled as follows:

$$P(y, \mathbf{v}, \mathbf{h}^1, \mathbf{h}^2; \theta) = \frac{1}{Z(\theta)} e^{(-E(y, \mathbf{v}, \mathbf{h}^1, \mathbf{h}^2; \theta))} \quad (5.31)$$

$$\begin{aligned} E(y, \mathbf{v}, \mathbf{h}^1, \mathbf{h}^2; \theta) = & - \sum_{i=1}^{N_v} \sum_{j=1}^{N_{h1}} w_{ij}^1 \frac{v_i}{\sigma_i} h_j^1 - \sum_{j=1}^{N_{h1}} \sum_{l=1}^{N_{h2}} w_{jl}^2 h_j^1 h_l^2 - \sum_{l=1}^{N_{h2}} \sum_{m=1}^C w_{lm}^3 h_l^2 y_m \\ & - \sum_{i=1}^{N_v} \frac{(v_i - b_i)^2}{2\sigma^2} - \sum_{j=1}^{N_{h1}} a_j^1 h_j^1 - \sum_{l=1}^{N_{h2}} a_l^2 h_l^2 - \sum_{m=1}^C a_m^3 y_m \end{aligned} \quad (5.32)$$

Here, $\mathbf{v} \in \mathbb{R}^{N_v}$ denotes the real-valued visible vector, N_v, N_{h1}, N_{h2} are the number of units in the

visible and hidden layers respectively, and $\theta = \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^3, \mathbf{b}, \mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3, \sigma\}$ is the set of model parameters, representing visible-to-hidden and hidden-to-hidden symmetric connection weights, bias terms, and the Gaussian distribution standard deviation. Biases are equivalent to the weights of a connection with a static value of one and $Z(\theta)$ is the normalizing constant. The conditional distribution of the visible and hidden layers are given as:

$$P(v_i = 1 | \mathbf{h}^1; \theta) = \mathcal{N}(v_i | \mu, \sigma^2) \quad (5.33)$$

$$P(h_j^1 = 1 | \mathbf{v}, \mathbf{h}^2; \theta) = \text{sigm} \left(\sum_{i=1}^{N_v} \frac{v_i}{\sigma^2} w_{ij}^1 + \sum_{k=1}^{N_{h^2}} h_k^2 w_{jk}^2 + \sum_{m=1}^C y w_{jm}^3 + a_j^1 \right) \quad (5.34)$$

where, $\mu = \sum_{j=1}^{N_{h^1}} h_j^1 w_{ij}^1 + b_i$ is the mean of the visible layer. $\mathcal{N}(\cdot | \mu, \sigma^2)$ is the probability density function of a normal distribution with mean μ and standard deviation σ . If only \mathbf{W}^1 is considered and the other weights are set to zero, the derivative of the log-likelihood with respect to the model parameters is:

$$\frac{\delta \log P(\mathbf{v}; \theta)}{\delta \mathbf{W}^1} = \mathbb{E}_{P_{data}}[\mathbf{v} \mathbf{h}^{1T}] - \mathbb{E}_{P_{model}}[\mathbf{v} \mathbf{h}^{1T}] \quad (5.35)$$

Here, $\mathbb{E}_{P_{data}}[\cdot]$ denotes the expectation with respect to the data distribution and $\mathbb{E}_{P_{model}}[\cdot]$ is the expectation with respect to the distribution defined by the **DBM** as in Equation 5.31. Similar derivatives are obtained for \mathbf{W}^2 with the product $\mathbf{v} \mathbf{h}^1$ replaced by $\mathbf{h}^1 \mathbf{h}^2$ and also for \mathbf{W}^3 .

Constructing cssDBM

A **Deep Belief Network (DBN)** [182] is a generative network consisting of a stack of RBMs, where the hidden layer of each RBM acts as the visible layer for the next RBM in the stack. A **DBN** is trained one RBM at a time, in a greedy manner since training all the layers simultaneously is highly computationally expensive. Contrastive divergence learning can be applied to greedily learn individual **cssRBM** since the cost function is differentiable. In the **cssDBN**, the hidden layer in the first **cssRBM** acts as the visible layer for the second **cssRBM** and so on. A two layer **DBN**

formulation can be defined as:

$$P(y, \mathbf{v}, \mathbf{h}^1, \mathbf{h}^2; \theta) = P(y, \mathbf{h}^2, \mathbf{h}^1)P(\mathbf{v}|\mathbf{h}^1) \quad (5.36)$$

where, \mathbf{v} is the set of visible nodes, and \mathbf{h}^1 and \mathbf{h}^2 are the two hidden layers. Let $\mathbf{W}_{\mathbf{v}\mathbf{h}^1}$ and $\mathbf{W}_{\mathbf{h}^1\mathbf{h}^2}$ denote the weights between the visible layer and the first hidden layer, and the weights between the first and the hidden layer, respectively. The joint distribution of the top two layers and conditional distribution of the visible layer are given as

$$P(y, \mathbf{h}^2, \mathbf{h}^1; \theta) = \frac{1}{Z(\theta)} e^{-E(y, \mathbf{h}^2, \mathbf{h}^1; \theta)} \quad (5.37)$$

where,

$$E(y, \mathbf{h}^2, \mathbf{h}^1; \theta) = \mathbf{h}^{1T} \mathbf{W}_{\mathbf{h}^1\mathbf{h}^2} \mathbf{h}^2 + \mathbf{h}^{2T} \mathbf{W}_{\mathbf{h}^2\mathbf{y}} y + \mathbf{a}^{1T} \mathbf{h}^1 + \mathbf{a}^{2T} \mathbf{h}^2 + \mathbf{a}^{3T} y \quad (5.38)$$

$$P(\mathbf{v}|\mathbf{h}^1) = \prod_{i=1}^{N_v} P(v_i = 1|\mathbf{h}^1) = \text{sigm} \left(b_i + \sum_{j=1}^{N_{h^1}} w_{ij} h_j^1 \right) \quad (5.39)$$

Here, θ denotes the cssDBN parameters, b_i is the bias associated with the hidden unit h_i , v_j is the j^{th} unit in the visible layer, and w_{ij} is the weight of the connection between h_i and v_j ($\mathbf{W}_{\mathbf{v}\mathbf{h}^1}$).

In both cssDBM and cssDBN models, $\ell_{2,1}$ -norm regularization is applied in a greedy layer wise approach where each layer is trained as an cssRBM, as explained in Equation 5.27. Further, the overall fine-tuning is performed using the regularization constraint as well.

5.4.4 cssDBM and cssDBN with Dropout and Dropconnect

The idea behind class sparsity signature constraint is to make sure that the sparse hidden representation for a class is similar. This regularization method can be also complemented with other popular regularization methods as shown in Table 5.1. Two successful methods, dropout and dropconnect, are discussed in this section. In this research, we also formulate how dropout and dropconnect can be integrated with cssDBM and cssDBN.

Dropout is a technique where randomly selected subset of activations are set to zero with

each layer. Dropconnect generalizes this idea by randomly setting a subset of weights within the network to zero. Consider a Bernoulli random variable, r , which takes a value of 1 with probability p . In dropout, a vector of binary mask is formed for each hidden layer, such that the hidden node h_i is retained if $r_i = 1$. Thus for dropout, the conditional distribution can be reformulated as

$$p(h_i = 1|r_i, \mathbf{v}) = 1 \cdot (r_i = 1) * \frac{1}{1 + \exp(-b_i - \sum_j v_j w_{ij})} \quad (5.40)$$

For dropconnect, a matrix of binary mask is formed to mask the weights between every two consecutive layers, such that the weight w_{ij} is retained if $r_{ij} = 1$. For dropconnect, the conditional distribution should be reformulated as

$$p(h_i = 1|r_{ij}, \mathbf{v}) = \frac{1}{1 + \exp(-b_i - \sum_j v_j (1 \cdot (r_{ij} = 1) * w_{ij}))} \quad (5.41)$$

With the updated probability distribution for the hidden layers in Equations 5.40 and 5.41, the energy function for cssDBM (Equation 5.31) and cssDBN (Equation 5.36) gets updated accordingly. Thus, class sparse signature constraint can be complemented with dropconnect or dropout regularization in both DBM and DBN. Particularly, during the fine-tuning stage, $\ell_{2,1}$ -norm regularization with dropout or dropconnect enforces the supervised feature learning of the model.

5.5 Latent Fingerprint Minutia Extraction

The main idea is to use the proposed regularization for the unsupervised deep learning algorithm so as to better discriminate between the minutia and non-minutia patches from latent fingerprints. Though minutia extraction from inked and live-scan fingerprints are well addressed problems [9], extracting minutia from latent fingerprint images is still an open research problem [140]. Due to the challenging nature of the problem, not many algorithms exist for automated latent fingerprint minutiae extraction. In some of the earlier research, existing tenprint matchers are utilized to extract minutia information from latent fingerprints [200], [85], [137]. However, with poor performance of these algorithms, researchers have realized the need for latent specific minutia extractor which is able to handle poor quality information in a more robust way. Paulino et al. [85], [65] proposed a MCC [142] based descriptor for manually annotated minutia features. It can be ob-

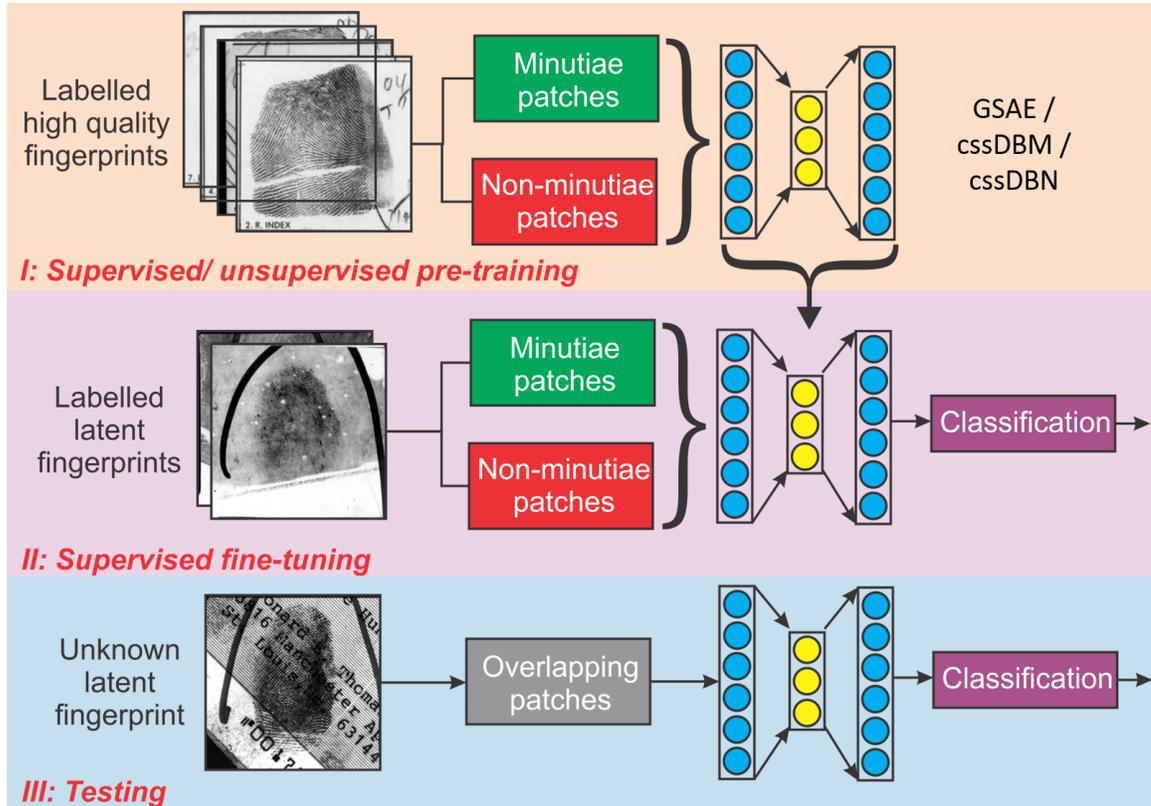


Figure 5-4: Block diagram to explain the various stages in the proposed algorithm. I: Pre-training stage where the group sparse deep autoencoder is learnt from labelled high quality fingerprints, II: Fine-tuning stage where the feature learner and classifier are trained with labelled latent fingerprints, and III: Testing stage in which local patches from unknown latent fingerprint are classified as minutia and non-minutia patches.

served from Figure 5-2(a) that the local region around a minutia has a different ridge structure than a non-minutia patch. However, as shown in Figure 5-2(b), latent fingerprint minutia patches lack a definite structure, making it challenging to learn meaningful information. Owing to the non-uniform and uncertain variations in latent fingerprints, it has been challenging for researchers to define a model for extracting minutiae. Human engineered features such as gradient information and frequency based information, provide limited performance due to the presence of background noise. In our initial study [107], we proposed the first automated algorithm for latent fingerprint minutiae extraction using SDSAE [201] to learn latent fingerprint local patch description. In this research, we design an automated minutiae extraction algorithm for latent fingerprint using GSAE, cssDBM, and cssDBN. Figure 5-4 illustrates the three main stages of the formulation for latent fingerprint minutiae detection and are discussed as follows:

1. **Learning feature descriptor:** Minutiae patches and non-minutiae patches are extracted separately from tenprint fingerprint images. The aim is to learn separately a minutiae patch descriptor and a non-minutiae patch descriptor from these local patches.
2. **Training binary classifier:** Minutiae extraction in latent prints is presented as a binary classification problem - whether the given latent patch is a minutia patch or a non-minutia patch. Labeled latent print patches (both minutia and non-minutia) are represented using the descriptors learnt in the previous step. A 2ν -SVM based binary supervised classifier is then learnt to classify between the minutiae and non-minutiae patches.
3. **Detecting minutiae patch:** Whenever an unseen latent print patch is provided, the minutiae and non-minutiae descriptor of the patch are extracted and classified using the trained 2ν -SVM classifier. Minutia extraction in the entire latent print is performed by classifying every local block as a minutiae or non-minutiae patch.

We have used 2ν -SVM [202] with radial basis function kernel for classification. 2ν -SVM is a "cost-sensitive" version of SVM that penalizes the training errors of one class more than the other by assigning class specific weights to both the classes. This explicit penalty minimizes the false negatives while restricting the false positives below a certain significance level. Hence, in case of imbalanced class data or in case of different cost of error, different importance can be given to the two types of errors, making sure that the majority class is not creating a bias. Further, in case of c -class classification problems, c different binary classifiers are created using the "one-vs-all" approach to train binary 2ν -SVM. The primal form of 2ν -SVM optimization function [202] is given as

$$\min_{w,b,\psi,\rho} \frac{1}{2} \|w\|^2 - \nu\rho + \frac{\gamma}{n} \sum_{i \in I_+} \psi_i + \frac{1-\gamma}{n} \sum_{i \in I_-} \psi_i \quad (5.42)$$

such that, (i) $y_i(k(w, x_i) + b) \geq \rho - \psi_i$, (ii) $\psi_i \geq 0$, and (iii) $\rho \geq 0$. Here, w is the decision boundary, x are the support vectors, y are the corresponding class labels, $k(w, x_i)$ is the kernel function, ψ_i are the slack variables, $\gamma \in \{0, 1\}$ is a parameter controlling the trade-off between false positives and false negatives, and $i = \{1, 2, \dots, n\}$ for n support vectors.

After classifying a patch as minutiae patch, the center of the patch is assumed to be the location of minutiae. The gradient of the ridge information is computed for the patch of size $w \times w$ with

respect to the center pixel (i, j) , as follows:

$$\mathcal{V}_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\delta_x(u, v)\delta_y(u, v) \quad (5.43)$$

$$\mathcal{V}_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\delta_x^2(u, v)\delta_y^2(u, v)) \quad (5.44)$$

$$\theta(i, j) = \frac{1}{2}\tan^{-1}\left(\frac{\mathcal{V}_x(i, j)}{\mathcal{V}_y(i, j)}\right) \quad (5.45)$$

Minutiae matching algorithms, such as bozorth3 (NBIS from NIST), use a bounding box approach to match minutiae ie., the distance between the matching pair of minutiae should be less than a threshold¹. As the match happens with a bounding box (default 32x32), if the location predicted minutiae is anywhere within the bounding box limits of the actual minutiae, then it is still counted as a match. This aspect of a matching algorithm is leveraged to predict the minutiae patch. We further observe that predicting the precise location of the minutiae may be skipped without much loss in accuracy. Similarly, while pairing with the full fingerprints, the bounding box approach chooses only one latent minutiae from a neighborhood region. Hence, in our case it is not imperative to address with the overlapping patches during minutiae extraction.

5.6 Experimental Protocol and Performance Analysis

5.6.1 Fingerprint Datasets

The unsupervised deep learning algorithms require a large database for learning a robust feature representation. Since the collection of latent fingerprints is a time consuming and challenging task, there are only a few latent fingerprint datasets available in the public domain. Therefore, we first prepare the heterogenous fingerprint database by combining four publicly available live-scan fingerprint databases and use it as the pre-training data set. The four databases are: CASIA-FingerprintV5 [131], NIST SD-14 v2 [124], FingerPass [203], and MCYT [132]. The description

¹ Refer: <https://github.com/lessandro/nbis/blob/master/bozorth3/src/lib/bozorth3/bozorth3.c>

Table 5.2: Summary of the composition and characteristics of the heterogenous fingerprint database. This database is used as the pre-training dataset for the proposed deep learning approach.

Database	Capture type	#Images	#Minutiae
CASIA-FingerprintV5 [131]	Optical	20,000	515,641
NIST SD-14 v2 [124]	Card print	54,000	8,188,221
FingerPass [203]	Optical, capacitive	34,560	812,643
MCYT [132]	Optical, capacitive	24,000	571,713
Total		132,560	10,088,218

Table 5.3: Summary of latent fingerprint databases used in our experiments, including the number of train patches and test patches used in each of the three cross validation experiments.

Database	#Images	#Train Patches	#Test Patches
NIST SD-27 [4]	258	9757	65,274
	Fold 1	5,503	65,274
	Fold 2	5,439	65,274
	Fold 3	5,441	65,274
MOLF [175]	4,400	-	422,400

and properties of these datasets are summarized in Table 5.2. To make the feature learning supervised, minutiae are extracted from all the fingerprints using an open source minutia extractor *mindtct* of the NBIS [5]. An image patch of size 64×64 ($w = 64$) is extracted with minutia at the center, thereby creating 10,088,218 number of minutia patches extracted from all the images. From every fingerprint, same number of non-minutia patches and minutia patches are extracted to ensure same number of samples from both the classes. The proposed algorithm is trained with raw image intensities of these image patches (vector size 1×4096) as input. For evaluation, the following two publicly available latent fingerprint databases are selected. A summary of the latent fingerprint datasets is shown in Table 5.3 and explained below:

- **NIST SD-27 dataset** [4]: The database has 258 latent fingerprints, pre-classified as good, bad, and ugly, based on their biometric quality along with minutiae points, manually annotated by forensic experts. Since the minutia patches of high quality latent fingerprints (as a part of heterogenous database) are different from field quality latent fingerprints, 50% randomly chosen images from the NIST SD-27 database are used to fine-tune the deep learning

model learnt using the heterogenous fingerprints. The remaining 50% images (129 latent fingerprints) are used for testing the classification performance. For fine-tuning the proposed deep learning model and learning the classification model, same number of minutia and non-minutia patches of size 64×64 are extracted from each training image. Three times random cross validation is performed to remove any training bias.

- **MOLF dataset** [175]: It consists of 4,400 latent fingerprints from 100 different subjects (all 10 fingers). All the latent fingerprints are lifted using black powder from a tile background. The manually annotated minutiae are also available along with this database. Since the pre-defined protocol of **MOLF** does not provide any training subset, we have used the entire **MOLF** dataset as test set and the best trained model obtained from the **NIST SD-27** database is used for performance evaluation. It contains 422,000 test samples.

5.6.2 Evaluation Metrics

The primary objective of this algorithm in fingerprints recognition is correctly extracting minutiae from latent fingerprint images. Therefore, the performance metric used in all these experiments is Correct Classification Accuracy (CCA %), which denotes the ratio of correctly classified patches with the total number of patches. The overall accuracy is further split into class-specific classification accuracy: Minutiae Detection Accuracy (MDA) and Non-Minutiae Detection Accuracy (NMDA). In terms of MDA and NMDA, although both the accuracies should be high, it is important to detect all the minutia patches accurately along with minimizing the occurrence of spurious minutia patches.

$$\text{MDA} = \frac{\text{No. of correctly classified minutia patches}}{\text{Total no. of minutia patches}} \times 100 \quad (5.46)$$

$$\text{NMDA} = \frac{\text{No. of correctly classified non-minutia patches}}{\text{Total no. of non-minutia patches}} \times 100 \quad (5.47)$$

5.6.3 Latent Fingerprint Minutiae Extraction Performance

The experimental performance of **GSAE** on benchmark image datasets such as MNIST, CIFAR-10, and SVHN are discussed in the appendix (See Appendix A). The performance of the proposed

Table 5.4: Classification results of the proposed and existing algorithms on the NIST SD-27 latent fingerprint dataset. The results are reported in terms of CCA (%). MDA is the Minutia Detection Accuracy and NMDA is the Non-Minutia Detection Accuracy.

Algorithm	Classifier	CCA	MDA	NMDA
VeriFinger	VeriFinger	90.33	20.41	96.80
Sankaran et. al. [107]	Softmax	46.80	65.18	41.21
KLD	2ν -SVM	99.31	91.90	100
GSAE	2ν -SVM	99.53	94.48	100
KLD + GSAE	2ν -SVM	99.61	95.37	100
cssDBN	2ν -SVM	99.67	96.16	100
cssDBM	2ν -SVM	99.68	96.21	100

approach is evaluated on two different datasets, NIST SD-27 and MOLF, under four different experimental scenarios: (i) using VeriFinger, which is a popular commercial tool for fingerprints, (ii) using the proposed architecture with only **KL-Divergence (KLD)**, (iii) using the proposed architecture with only **GSAE**, and (iv) using the proposed architecture with **KLD + GSAE**. We also compared the results with current state-of-the-art algorithm proposed by Sankaran et al. [107]. The results on **NIST SD-27** and **MOLF** are summarized in Table 5.4 and Table 5.5 respectively.

Table 5.5: Classification results of the proposed and existing algorithms on the MOLF latent fingerprint dataset. The results are reported in terms of correct classification accuracy (%), minutia detection accuracy, and non-minutia detection accuracy.

Algorithm	Classifier	CCA	MDA	NMDA
VeriFinger	VeriFinger	78.52	21.33	92.92
KLD	2ν -SVM	59.25	84.17	52.97
GSAE	2ν -SVM	90.14	90.44	90.07
KLD + GSAE	2ν -SVM	90.74	90.63	90.37
cssDBN	2ν -SVM	92.16	92.37	92.11
cssDBM	2ν -SVM	92.19	92.45	92.12

As shown in Table 5.4, on the **NIST SD-27** database, the correct patch classification accuracy of the proposed algorithm is as high as 99% when using **KLD + GSAE**. The standard deviation of cross-validation experiments is in the range of ± 0.1 , denoting very little training bias. However, the MDA of VeriFinger is around 20% showing that it rejects a lot of genuine minutia patches. The

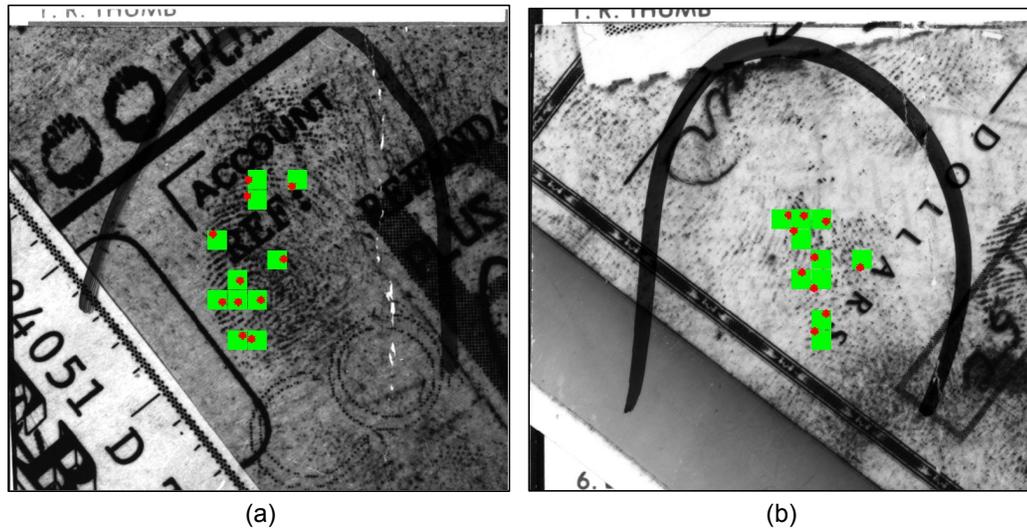


Figure 5-5: Sample example latent fingerprints from [NIST SD-27](#) database showing correct results of the proposed algorithm. Red dots denote manually annotated minutiae and green patches represent the minutia patches predicted using the [GSAE](#) algorithm.

architecture of Sankaran et al. [107] yield the MDA of 65%. In comparison to that, the proposed algorithms using [GSAE](#), [cssDBN](#), and [cssDBM](#) yields an improvement of more than 30%.

While detecting non-minutia patches, we observed that the algorithm of Sankaran et al. [107] yields the lowest accuracy of 41.21% followed by VeriFinger which yields 96.80%. This shows that VeriFinger can efficiently detect the background patches. The proposed algorithms yield 100% NMDA on the same experimental protocol. Such a high accuracy can be attributed to 2ν -SVM classification, which supports in making the false positive error almost zero. As shown in [Figure 5-5](#) and [Figure 5-6](#), on the [NIST SD-27](#) database, the minutia detection accuracy is very high and very small number of spurious minutia patches are extracted.

The second database used for performance evaluation is the [MOLF](#) database. This is a very large database, however, there is no defined training database. Therefore, the results on the [MOLF](#) database are obtained by training the model on the [NIST SD-27](#) dataset and testing the best learned model with the [MOLF](#) database. Since Sankaran et al. yields lower accuracies on the [NIST SD-27](#) database, on the [MOLF](#) database, we have only compared with VeriFinger, [KLD](#) and [GSAE](#). As shown in [Table 5.5](#), the proposed architecture yields classification accuracies of over 90% with standard deviation in the range of ± 0.15 . Comparing with different regularizations reveals that [KLD + GSAE](#) provides the best results on both the datasets and the performance of [GSAE](#) is better

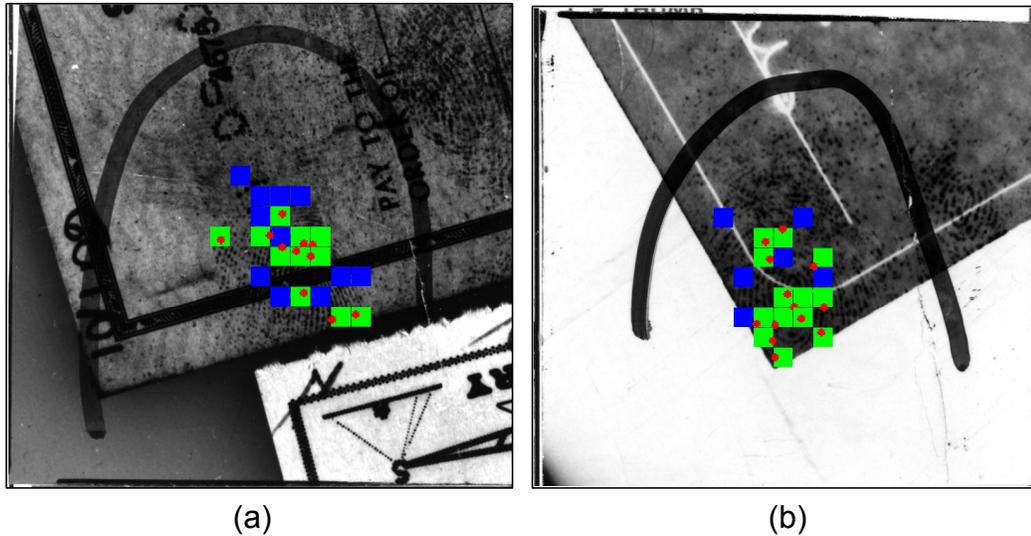


Figure 5-6: Sample latent fingerprints from [NIST SD-27](#) database showing some incorrect predictions obtained using the proposed algorithm. Red dots represent the annotated minutiae, green patches represent the correct minutiae patches predicted using the [GSAE](#) algorithm and blue patches show the incorrect ones (false positive). It can be observed that no genuine minutiae is rejected by the proposed algorithm.

than the popular [KLD](#) regularization. On 3349 images out of the total 4400 images present in the [MOLF](#) database, VeriFinger failed to extract any minutia. This shows that VeriFinger yields poor results in extracting genuine minutiae whereas using only [KLD](#) regularization extracts lots of spurious minutiae - the non-minutiae detection accuracy is only 52.97%.

The results on both the database show that the performance of the proposed minutiae extraction algorithm is better than the existing algorithms. It is our assertion that the performance of the proposed algorithm on the [MOLF](#) database is not as good as on the [NIST SD-27](#) because, (i) the number of testing data points on the [MOLF](#) database is very large compared to the [NIST SD-27](#) dataset and (ii) there are significant variations in the characteristics of the two databases. Using the model trained with [NIST SD-27](#) also contributes to lower accuracies on the [MOLF](#) database.

Further, for a latent fingerprint image of size 800×768 , typically 600 patches has to be classified. A Matlab 2014 implementation on a 2.66 GHz dual-core processor with 16GB RAM extracts all the minutiae from a single image in around 627 milliseconds. Therefore, the proposed approach is suitable for real-time system as well.

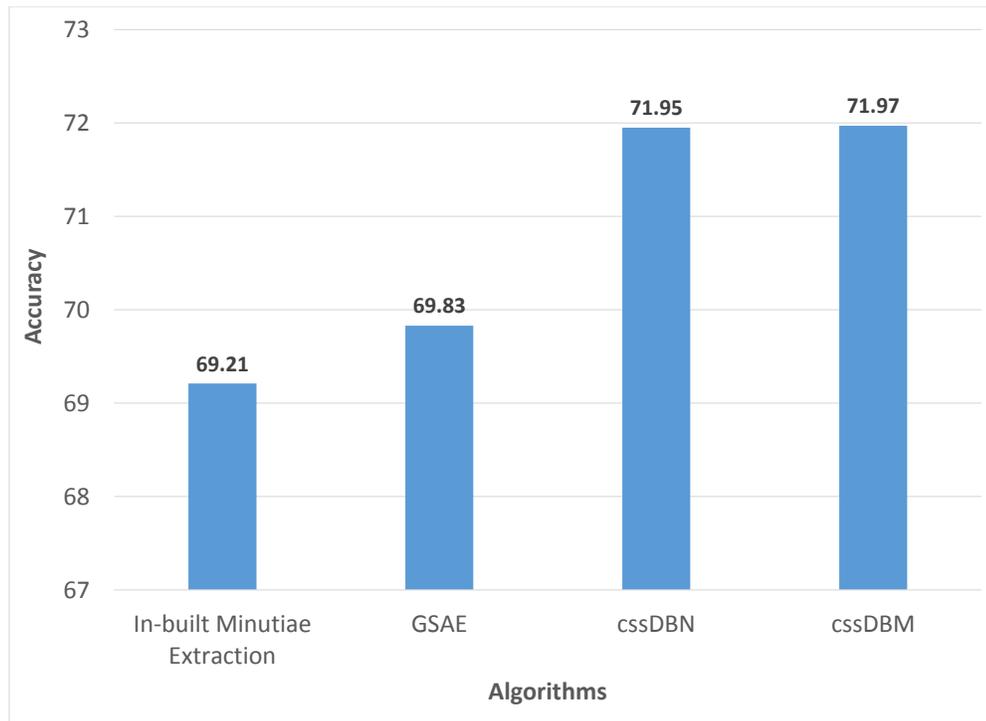


Figure 5-7: Recognition performance on the [MOLF](#) dataset using a latent fingerprint identification system. The in-built minutiae extractor is compared against the proposed minutiae extraction algorithms including [GSAE](#), [cssDBM](#), and [cssDBN](#), while the matching algorithm is kept constant.

5.7 Latent Fingerprint Recognition Performance

We further evaluate the efficacy of the proposed [GSAE](#) based minutiae extraction approach using a popular latent fingerprint identification system². The identification system is modular in nature which provides the flexibility of keeping the entire pipeline constant while changing only one component. This facilitates evaluating the performance of the proposed approach with minutiae extracted from other approaches as well. The system has over 2 million pre-enrolled identities in the database that can be used as the large gallery in the experiments. Since the [NIST SD-27](#) database may have been used to train the system, we have performed matching experiments only with the [MOLF](#) database. Gallery images from the [MOLF](#) database are enrolled³ and probe

²The license agreement does not allow us to name the commercial system in any kind of comparison.

³After the experiments, these enrollments are deleted from the system.

images are used for testing. From a probe fingerprint image, first the minutiae points are extracted using the proposed algorithm by first finding the minutiae patches and then taking its center as the minutia point. The minutiae template/feature is given to latent fingerprint matching system which matches against the large gallery (which also includes gallery from the [MOLF](#)). We also compare the performance with the in-built approach of latent fingerprint system in which minutiae are extracted by the system itself and matched against the gallery. We obtain rank-50 accuracies pertaining to both these experiments. For the proposed approach, rank-50 accuracy is 69.83% whereas for the inbuilt feature extraction approach in latent fingerprint system yields 69.21%, as shown in Figure 5-7. Though this seems slight improvement by the proposed algorithm, this is still noteworthy because of the large scale matching (using more than 2 million gallery identities) with over 4,000 probe latent fingerprint images. This experiment demonstrates that the proposed algorithm is highly promising for automatic latent fingerprint feature extraction.

The performance of [cssDBM](#) and [cssDBN](#) for latent fingerprint minutiae extraction are experimentally studied with the same datasets and protocols adopted from evaluated [GSAE](#). The latent fingerprint identification system provided a rank-50 identification accuracy of 71.95% using the minutiae extracted from [cssDBN](#) and 71.97% using the minutiae extracted from [cssDBM](#). In comparison, minutiae extracted from [GSAE](#) provided a rank-50 identification accuracy of 69.83% while the inbuilt minutiae extraction of the identification system provided an accuracy of 69.21%. This demonstrates the efficacy of the proposed minutiae extraction algorithms.

5.8 Summary

This research presents a novel supervised regularization method for autoencoders using $\ell_{2,1}$ -norm which utilizes class labels to learn supervised features for the specific task at hand. The optimization function is solved using a majorization-minimization approach and applied in two different unsupervised feature learning algorithms. $\ell_{2,1}$ regularized stacked autoencoder is called as group sparse autoencoder ([GSAE](#)) and regularized [RBM](#) is called as class sparsity specific restricted Boltzmann machines ([cssRBM](#)). Further using the representation learning approach, an automatic latent fingerprint minutia extraction algorithm is formulated as a binary classification algorithm. The minutiae extraction algorithm is evaluated on two publicly available latent finger-

print databases, [NIST SD-27](#) and [MOLF](#). The results show that the proposed algorithm improves the performance of automated latent fingerprint feature extraction. It is our assertion that the effectiveness of [GSAE](#), [cssDBN](#), and [cssDBM](#) can be further utilized to improve the classification performance with challenging and noisy databases.

Chapter 6

Smartphone Fingerphoto Recognition

In today's world, smartphones and handheld devices are omnipresent and they are identified as one of the fastest spreading technologies [32]. With such a penetrating growth, smartphones have become an inevitable part of our day-to-day life, holding all our personal data in one place. With increasing capabilities and power of mobile phones, its use in e-commerce applications such as mobile banking is also increasing. Hence, access control to these devices should be secure, flexible, and easy-to-use. Traditionally, the access control mechanisms adopted in smartphones include pins, passwords, and patterns. These lock-screen authentication mechanisms are popularly used and well explored in existing smartphones. However, these mechanisms are prone to attacks including over-the-shoulder copy attack. Therefore, it is important to search for another authentication mechanism that is accurate and less prone to presentation or copy attack. In the realm of biometrics, other modalities such as fingerprint, face, and iris have been explored [33], [34], [35] along with biometric gesture based mechanisms in touch screen smartphones [36], [37], [38]. One such example is BioID facial recognition application (<https://mobile.bioid.com/>), which is a multifactor user authentication application using face biometrics. From a usability point of view, fingerprints have been found to be more adoptable and easier than pin based access control mechanism [39].

There are two broad approaches on how finger impression can be used for providing access control to smartphones: (i) fingerprint based authentication, and (ii) fingerphoto based authentication. Fingerprint recognition is performed with the use of specially designed fingerprint sensors. An embedded sensor (external or attached within the display unit) is used to capture the fingerprint,

and minutia based approaches are used for matching. Some of the existing smartphones have integrated fingerprint sensors for authentication [33]. However, adding a capacitive or resistive sensor to a smartphone further adds to the cost of the device. Fingerphoto¹ based authentication utilizes the in-built camera to capture a photo of the finger which can then be used for authentication. With improvements in technology, every smartphone has a good resolution rear-camera that is extensively being used. Using smartphone camera for fingerphoto capture can provide a cost effective method for user authentication.

This research focuses on designing efficient algorithm for automatic fingerphoto recognition. Inspired by the application of scattering network based features for other biometric modalities such as face [204] and iris [205], in our preliminary research [145] we proposed a deep ScatNet based matching pipeline for smartphone based fingerphoto images. We created IIITD Smartphone FingerPhoto Database-v1 (ISPF²-v1)², focusing on background and environmental illumination variations as two challenges for fingerphoto matching. An end-to-end matching pipeline was constructed with background segmentation, ridge structure enhancement, feature representation, and classifier based verification as the four modules of the pipeline. This research is an extension of the preliminary work with the following key contributions:

- a novel combination of saliency based and skin color based fingerphoto region of interest segmentation algorithm and removing the highly varying noisy background. The preliminary version of the algorithm [145] used only a skin color based segmentation.
- propose a novel application of Scattering Network (ScatNet) based feature representation [206] and matching algorithm for fingerphoto images,
- study the effect of camera resolution and the environmental variation in the captured image on the authentication performance of fingerphoto images, using Scattering Network (ScatNet) based feature representation [206] and matching algorithm for smartphone captured fingerphoto images,
- creating and releasing public IIITD SmartPhone Fingerphoto Database v2 (ISPF³-v2)³ con-

¹Fingerphoto images are hand finger ridge impressions captured directly using a camera in a touchless method. Fingerprint images are ridge impressions captured using a touch based live-scan sensor.

²Available at <http://iab-rubric.org/resources/spfd.html>

³Available at <http://iab-rubric.org/resources/spfd2.html>

sisting of more than 16,000 images obtained from 300 unique fingers. The fingerphoto images are taken using two smartphones: OnePlus One (OPO) and MicroMax Canvas Knight. The preliminary version of the research [145] used IIITD SmartPhone Fingerphoto Database v1 (ISPF-D-v1) having 5,100 images obtained from 128 unique fingers using only one smartphone.

- extensive experimental analysis on the proposed dataset studying the effect of segmentation, enhancement approaches. Further, we study the individual impact of (i) noisy background, (ii) environmental illumination, and (iii) camera resolution on the performance of fingerphoto authentication.

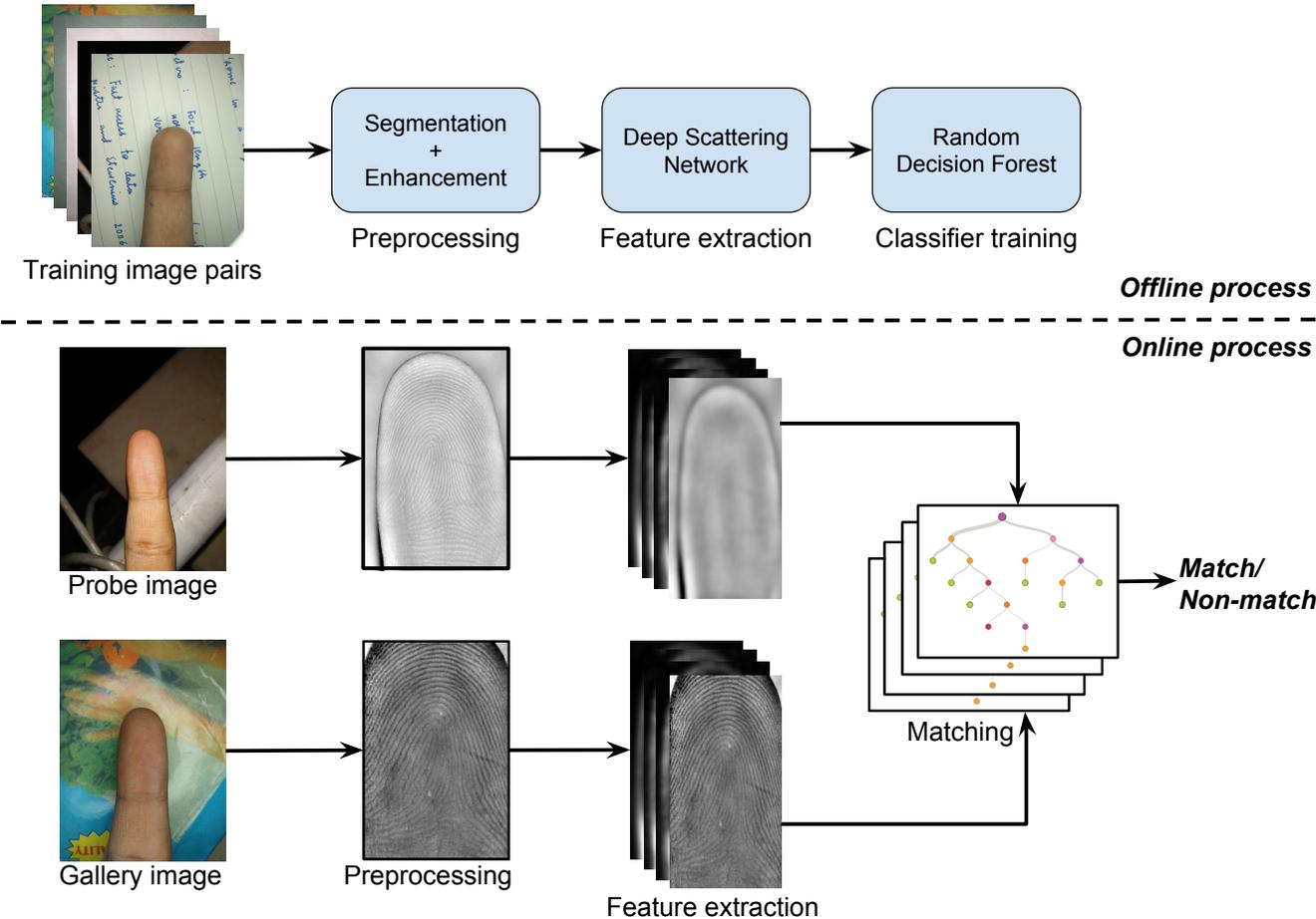


Figure 6-1: Illustrating the different steps involved in the proposed fingerphoto verification pipeline.

6.1 Proposed Fingerphoto Matching Algorithm

The proposed fingerphoto matching pipeline consists of four major steps: (i) fingerphoto segmentation, (ii) ridge pattern enhancement, (iii) ScatNet based feature representation, and (iv) matching. The individual steps are illustrated in Fig. 6-1 and explained in the sections below.

6.1.1 Fingerphoto Segmentation

A fingerphoto image, as discussed, can be captured under varying environmental conditions. For matching, it is essential to accurately segment the region of interest, i.e., the foreground region from the background objects in the image. The process of segmentation involves finding a binary mask in a captured fingerphoto image, that represents the *distal phalanges* of the finger. Even though a captured image can contain many background objects, it is safe to assume that, during capture, the finger is the closest object to the smartphone camera. Under this assumption, we observe two distinguishing features that separate the fingerphoto from the rest of the background: (i) the skin color of the finger region, and (ii) the salient nature [207] of the finger region in the image. As observed in Fig. 6-2, individually, skin color features could lead to both false positive and false negative while the salient region generally provides more false positive results in segmentation around the finger. Further analyzing the results of individual algorithms on different fingerphoto images, we observe that combining these two segmentation methods provide less erroneous results in segmenting the foreground finger region. Thus, the proposed algorithm combines the existing region covariance based saliency [208] along with skin color measurements for effective fingerphoto segmentation. The steps involved in the proposed segmentation algorithm are as follows:

Step 1: For every pixel, seven basic visual features (represented further as d -dimensional) are extracted: (i) intensity of the pixel in $L * a * b$ color space⁴, (ii) the edge orientation of the pixel along x and y directions, and (iii) x and y location of the pixel.

Step 2: For a region R_i in an image, a 7×7 covariance matrix C_i is constructed using the 7-dimensional feature vector. The first order statistics of C_i are computed and the extracted statistics

⁴<http://in.mathworks.com/help/images/examples/color-based-segmentation-using-the-l-a-b-color-space.html>

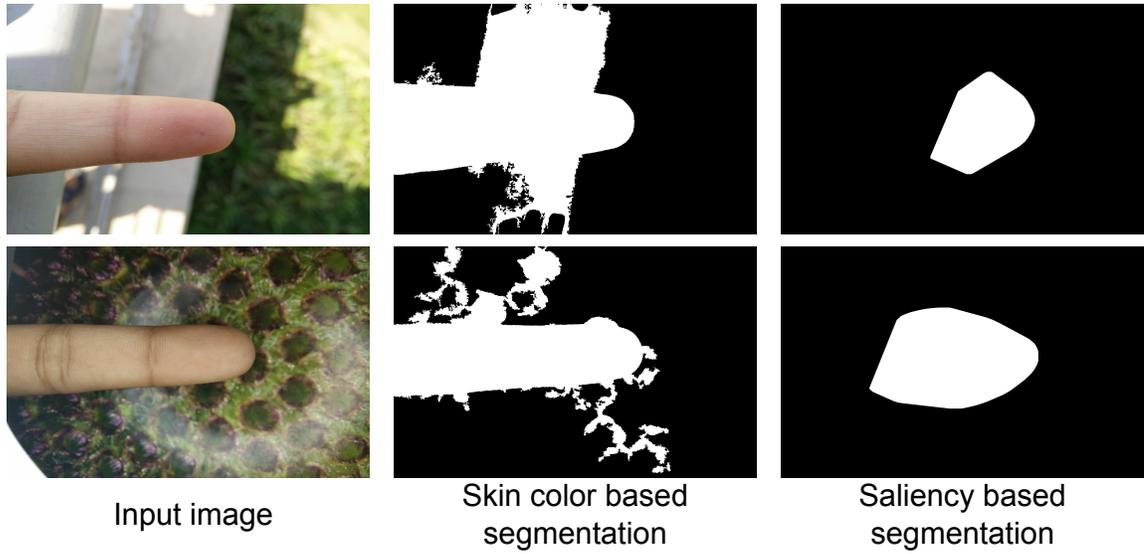


Figure 6-2: Example images showing the segmentation results of using only skin color based features or only saliency based features.

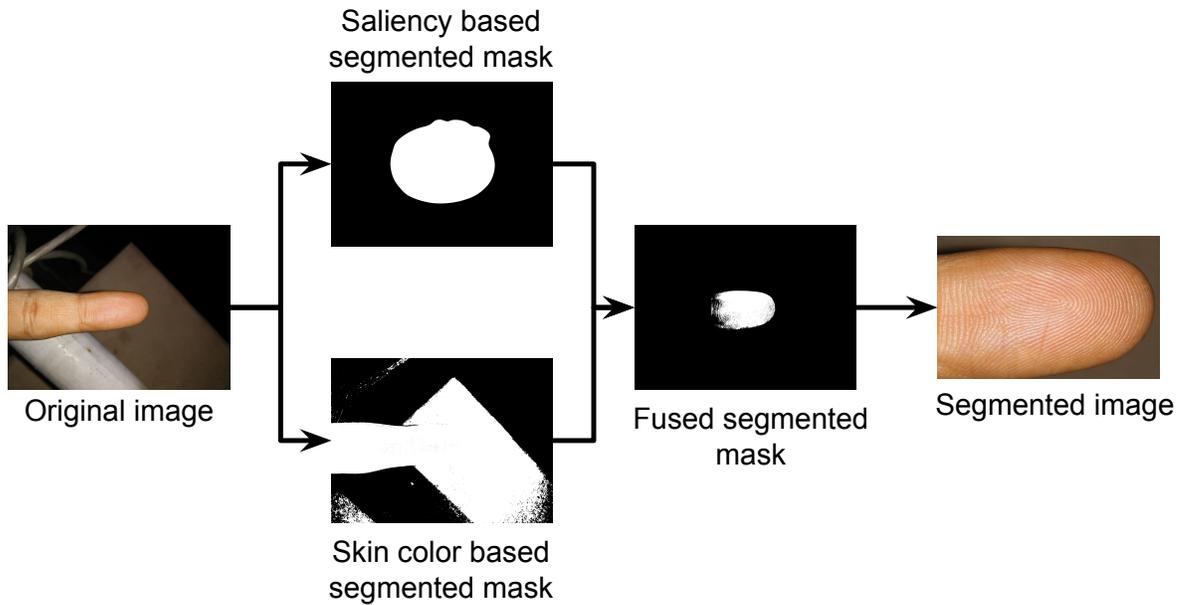


Figure 6-3: Illustration of the segmentation of algorithm combining the saliency based and skin color based segmentation maps.

are non-linearly aggregated, as follows,

$$\psi(C_i) = (\mu, s_1, \dots, s_d, s_{d+1}, \dots, s_{2d})^T \quad (6.1)$$

where, μ and s are the first order statistics mean and standard deviation, respectively. The size of the region, R_i is chosen based on the resolution of the image. In our research, we found a window of size 16×16 to be ideal.

Step 3: The dissimilarity between any two regions, $D(R_i, R_j)$ is computed as a normalized difference between their feature representation obtained using their covariance matrices, $\psi(C_i), \psi(C_j)$ as follows,

$$D(R_i, R_j) = \frac{\|\psi(C_i) - \psi(C_j)\|}{1 + \|x_i - x_j\|} \quad (6.2)$$

where, x_i and x_j denote the pixel location of the center of the image regions R_i and R_j , respectively.

Step 4: The covariance based saliency map [208] of any region, $map_{sal}(R_i)$, is computed by a non-linear aggregation of the dissimilarities of the current region R_i with the surrounding m most similar regions of the image, as follows,

$$map_{sal}(R_i) = \frac{1}{m} \sum_{j=1}^m D(R_i, R_j) \quad (6.3)$$

Step 5: To compute the skin color based segmentation, the entire **Red Green Blue (RGB)** image is converted into **Cyan Magenta Yellow Key (black) (CMYK)** color space. The normalized magenta channel image is used as the skin color based saliency map, map_{skin} , as magenta color channel retains most of the skin color.

Step 6: The overall segmentation map is obtained by combining the saliency based map and the skin color based map using a weight sum fusion. The segmentation map map_{seg} is obtained

by,

$$map_{seg} = w_1 \times map_{sal} + w_2 \times map_{skin} \quad (6.4)$$

where, w_1 and w_2 represent the weights associated to the normalized saliency map and normalized skin map, respectively.

Step 7: Otsu's thresholding method [209] is applied on the obtained map, map_{seg} , to obtain the binary segmented image of the fingerphoto.

Fig. 6-3 shows a visual illustration of the fusion of skin color based segmentation and saliency based segmentation maps to obtain the final segmented result.

6.1.2 Fingerphoto Enhancement

The aim of enhancement is to improve the image quality so that the finger ridge patterns can be efficiently extracted from the captured fingerphoto. The primary challenge for ridge extraction is the noise induced by the surrounding illumination variation which affects the contrast between the ridge and valley patterns. To remove the noise and enhance the ridge information, the segmented image obtained from the previous step is first converted to gray scale. To remove speckle noise introduced during capture, median filtering is applied on the segmented image. Further, histogram equalization is performed to address the illumination variation and the resulting image is sharpened to improve the contrast between ridge and valley structures and reduce the blur. In the resultant image, the ridge information constitutes the high-frequency components and valley and noise components constitute the low-frequency information. To enhance the difference between the ridge and valley information, sharpening is performed by subtracting the Gaussian blurred image ($\sigma = 2$) from the original image itself. Fig. 6-4 shows as example of the output provided by the enhancement algorithm.

6.1.3 ScatNet based Feature Representation

For feature extraction, researchers have explored the representation techniques, such as minutia, that are well established for fingerprint matching. We have observed that the variations in resolution and clarity among different fingerprint images are significantly less compared to that of

fingerphoto images. Therefore, it is challenging to accurately extract these features from fingerphoto images. Li et al. [40] have also shown that minutia based matching does not perform well for fingerphoto recognition.

As discussed earlier, the challenges in fingerphoto matching include variations due to illumination, resolution, noise, translation, and rotation. With the effect of illumination and background noise being addressed in the preprocessing stage (segmentation and enhancement), it is important to find a feature representation that is translation and rotation invariant [210] [211]. To encode the discriminative properties of fingerphoto images, we propose feature representation for fingerphoto images using Scattering Networks [212] [213]. It has been shown that ScatNet based features effectively encode texture patterns in images [206]. As shown in Fig. 6-4, the enhanced fingerphoto image has good ridge-valley texture patterns and it is our assertion that ScatNet features can effectively represent these patterns.

Deep Scattering Network⁵ is wavelet filter based network that produces a representation which is stable to local affine transformation. Let $x(u)$ be any signal in \mathbb{R}^2 (an image, in this case) and $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$ be a low-pass averaging filter and u be the set of parameters corresponding to the locations x and y in the image space. A locally affine invariant representation is obtained by the following convolution:

$$S_0x(u) = x \star \phi_J(u) \quad (6.5)$$

This representation is translation invariant up to 2^J pixels and also loses all the high frequency information. To utilize the high frequency information as well, a high frequency wavelet bank ψ is constructed by varying the rotation θ and the scale 2^j parameters. The high-frequency, quadrature phase, complex wavelet filterbank is given as, $\psi_{\theta,j}(u) = 2^{-2j}\psi(2^j\theta^{-1}u)$. The wavelet-modulus transform for high frequency components are obtained by:

$$|W_1|x = (x \star \phi(u), |x \star \psi_{\lambda_1}(u)|) \quad (6.6)$$

where, λ_1 corresponds to the set of first level filtering parameters (θ, j) . These high frequency filters provide additional information to the features obtained in Equation 6.5. Convolving the wavelet transform coefficients with a low pass filter produces an affine invariant representation of

⁵MATLAB toolbox publicly available at <http://www.di.ens.fr/data/software/scatnet/>

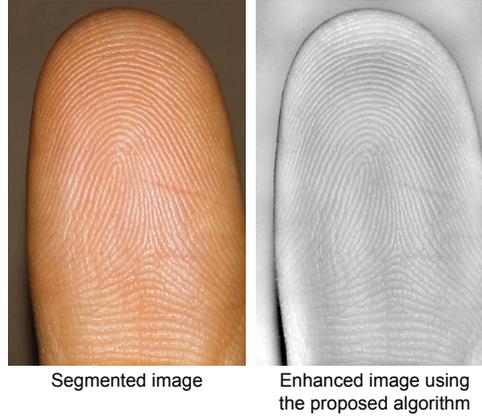


Figure 6-4: A segmented fingerphoto image and its enhanced image using the proposed algorithm, along with the output of applying a band pass filter, removing most of the high-frequency components.

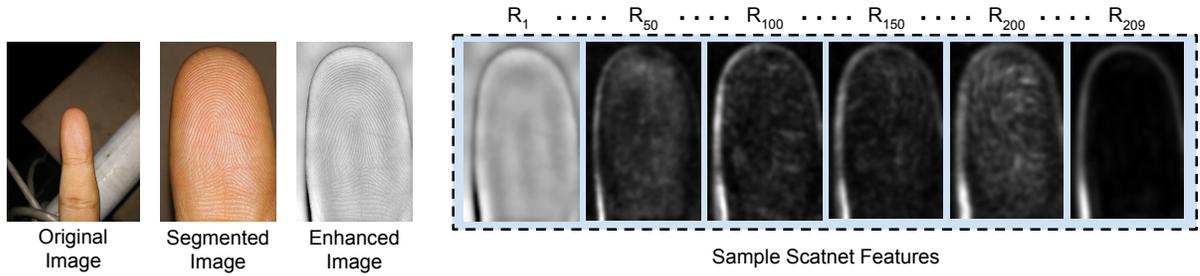


Figure 6-5: ScatNet feature representation of the enhanced image where R_1 to R_{209} represent the responses obtained from a wavelet filter bank of 209 filters, which is up to a depth of two in the scattering network.

the high frequency components, as follows

$$S_1 x(u, \lambda_1) = |x \star \psi_{\lambda_1}(u)| \star \phi_J(u) \quad (6.7)$$

These coefficients are called the first-order scattering network coefficients and represent the concatenation of all the filter responses in the wavelet bank $\psi_{\lambda_1}(u)$. If four different frequency bands are chosen in λ_1 , the overall response S_1 is the concatenation of wavelet responses of all four filters. Higher-order scattering network coefficients can be obtained by recursively constructing deeper wavelet filter banks as follows:

$$|W_2||x \star \psi_{\lambda_1}(u)| = (|x \star \psi_{\lambda_1}(u)| \star \phi, ||x \star \psi_{\lambda_1}(u)| \star \psi_{\lambda_1}|) \quad (6.8)$$

These higher order ScatNet coefficients provide a more stable translation and rotation invariant representation for the fingerphoto images, as shown in Fig. 6-5. Also, as these filters are pre-designed, calculating a ScatNet representation is convolving these filters over the image and there is no learning involved. Thus, it is easy to extract ScatNet features using only the computation power of smartphones. The effective representation for a fingerphoto is the concatenation of all n -order responses such as $\{S_0, S_1, \dots, S_n\}$.

In this research, we experimentally observe an optimal depth of the scattering network, $n = 2$ i.e., computing the second order ScatNet coefficients for all the fingerphoto images. Let the enhanced fingerphoto image, I_{enh} be of size $w \times h$. The concatenation of all responses upto the second order, $\{S_0, S_1, S_2\}$ contains a total of 209 filters, with each response of dimension $\frac{w}{8} \times \frac{h}{8}$. Thus, the overall ScatNet feature representation of the enhanced image is $209 \times \frac{w}{8} \times \frac{h}{8}$.

6.1.4 Feature Matching

Let P and G be the $1 \times N$ length vectorized ScatNet representations of the probe and the gallery fingerphoto images, respectively. A supervised binary classifier $g:(X \rightarrow Y)$ can be learned to classify a pair of ScatNet feature representations (P, G) as a match or non-match pair. The feature set X is the difference between the two feature representations ($P - G$) and the classification labels Y are $\{Match, Non-match\}$. From the difference of representations, the supervised classifier learns whether an image pair is a match or a non-match pair. Two fingerphoto images that belong to the same finger are defined as match pair, while images captured from different fingers or different subjects are considered as non-match pairs. In this research, we use [RDF](#) [214] as the binary classifier for verifying the pair of fingerphotos. [RDF](#) is a non-linear ensemble based classifier consisting of multiple decision trees [215]. It employs a repetitive random sub-sampling strategy for bagging which helps in providing robust and quicker results for correlated features. For the total number of \mathcal{D} fingerphoto images in the training set, several bootstrap aggregates of size $r \times \mathcal{D}$ are created with replacement from the data, for a ratio r ($0.5 < r \leq 1$). A forest containing T trees is trained where every decision tree, t in the forest is trained with a single bootstrap of the data

Table 6.1: Summarizing the characteristics and the variations captured in the [ISPF-D-v2](#).

Set	Challenge	Variations			Classes	Images
		Illumination	Background	Resolution		
Set I	White Indoor	Controlled	White	13 MP	304	2400
	White Outdoor	Uncontrolled	White	13 MP	304	2400
Set II	Natural Indoor	Controlled	Natural	13 MP	304	2400
	Natural Outdoor	Uncontrolled	Natural	13 MP	304	2400
Set III	Resolution	Controlled	White	5 MP	304	2400
	Resolution	Controlled	White	8 MP	304	2400
	Resolution	Controlled	White	16 MP	304	2400

creating an ensemble of classifiers. Let M be the length of the vectorized ScatNet representation obtained for the fingerphoto image. At every node in the decision tree, a random feature sample, m (typically $m = \sqrt[2]{M}$) is used to take the split decision based on an objective function of maximizing information gain. Each tree, t is designed as a binary decision tree by assigning leaf nodes as $\{Match, Non-Match\}$ corresponding to the training sample. Thus, each decision tree in the forest classifies the input pair of fingerphoto images as a matching pair or a non-match pair. The final decision is computed by taking a majority vote of all the decision trees in the ensemble.

6.2 IIITD SmartPhone Fingerphoto Database v2

In this research, we present [ISPF-D-v2](#) dataset consisting of more than 16,000 images obtained from 300 unique fingers. The fingerphoto images are taken using two smartphones: OnePlus One (OPO)⁶ and MicroMax Canvas Knight⁷. The preliminary version of the research [145] used [ISPF-D-v1](#) having 5,100 images obtained from 128 unique fingers using only one smartphone. Indoor images are captured in both constrained and uncontrolled environments, while in outdoor conditions, images are captured without flash during daylight and with flash during night. Auto-focus is always kept at ON status. Based on the challenges considered, the database is divided into three subsets and the summary of all three subsets is provided in Table 6.1. The three subsets are:

- **Set I - White Background:** Fingerphoto images are captured in both indoor (controlled

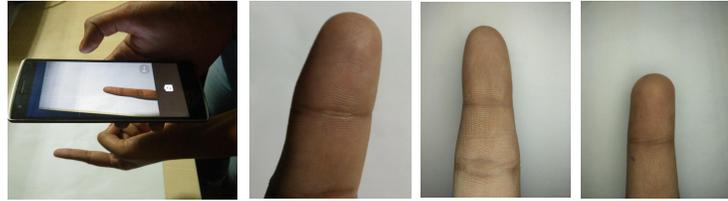
⁶<https://oneplus.net/>

⁷<http://www.micromaxinfo.com/canvasknight/>

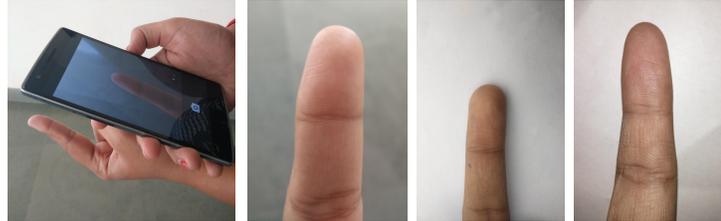
illumination) and outdoor (with uncontrolled lighting) environment with white background, as shown in Fig. 6-6(a) and Fig. 6-6(b). The two subsets, White Indoor (*WI*) and White Outdoor (*WO*) show the effect of varying illumination with a constant uniform white background. The images are taken using OnePlus One phone at 13MP resolution. Each subset has 8 images each of right index, right middle, left index, and left middle fingers of 76 subjects. This results in $76 \text{ subjects} \times 4 \text{ fingers} \times 2 \text{ lighting variations} \times 2 \text{ sessions} \times 4 \text{ instances} = 4864 \text{ images}$ for Set I.

- **Set II - Natural Background:** Fingerphoto images are captured in both indoor and outdoor environment, allowing any natural background to be present, as shown in Fig. 6-6(c) and Fig. 6-6(d). In real world applications, the natural background available in an indoor environment is very different as the background objects are much closer to the fingerphoto as compared to the outdoor background. Thus, the Natural Indoor (*NI*) subset shows the effect of background variation under controlled illumination while the subset Natural Outdoor (*NO*), shows the effect of background and illumination variations occurring together. The images are captured using OnePlus One phone at 13MP resolution. Similar to Set I, Set II also has 4864 images .
- **Set III - Resolution:** This set consists of fingerphoto images captured in three different resolutions with uniform controlled illumination and white background, as shown in Fig. 6-6(e). Two different smartphones, OnePlus One and MicroMax Canvas Knight, are used to capture the images at three different resolutions 5MP, 8MP, and 16MP. Flash LEDs are turned off while the auto-focus is kept ON. All the images are captured in an indoor lab environment, with uniform lighting and a blank white paper as the background. Under these settings, four instances of the index finger and middle finger of the right and left hand of 76 subjects are captured at all three resolutions. This results in $76 \text{ subjects} \times 4 \text{ fingers} \times 3 \text{ resolutions} \times 2 \text{ sessions} \times 4 \text{ instances} = 7296 \text{ images}$.

Fig. 6-6 shows sample fingerphoto images from the proposed database. The database will be made publicly available for academic research at the following link: <http://iab-rubric.org/resources/spfd2.html>.



(a) Indoor Illumination variation with white background



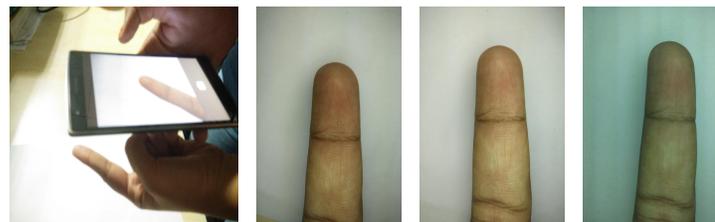
(b) Outdoor Illumination variation with white background



(c) Indoor background variation



(d) Outdoor background variation



(e) Camera resolution variation

Figure 6-6: Sample images showing various challenges addressed in [ISPFD-v2](#). (a)-(b) illumination variation with white background, (c)-(d) background variation, and (e) camera resolution variation. Multiple samples showing the intra-class variations and noise present in the database.

Table 6.2: The algorithmic pipeline of the different techniques used for comparison along with the proposed pipeline.

Type	Technique	Segmentation	Enhancement	Feature Extraction	Matching
Minutiae based	MCC	✓ (saliency + + skin color)	✓ (median filter + sharpen)	Minutiae Cylinder Code [142]	using VeriFinger SDK
	NBIS	✓ (<i>nfseg</i>)	✗	<i>mindtct</i>	<i>bozorth3</i>
Non-Minutiae	CompCode	✓ (Heuristics)	✓	Competitive Code [216]	ℓ_1 -distance
ScatNet based	Minaee and Wang [115]	✗ (saliency + skin color)	✗ (median filter + sharpen)	ScatNet + PCA	SVM
	ScatNet + ℓ_1	✓ (saliency skin color)	✓ (median filter + sharpen)	ScatNet + PCA	ℓ_1 -distance
	ScatNet + NN	✓ (saliency skin color)	✓ (median filter + sharpen)	ScatNet + PCA	Neural network
	ScatNet + RDF	✓ (saliency skin color)	✓ (median filter + sharpen)	ScatNet + PCA	RDF

6.3 Results and Analysis

The primary purpose of the experimental analysis is to evaluate the effectiveness of the proposed algorithm for fingerphoto to fingerphoto matching. The results are demonstrated in terms of EER and ROC curves.

6.3.1 Experimental Setting

A 50% unseen train-test protocol is followed, where half of the gallery and probe sets are used to create match and non-match pairs for training the supervised classifiers. To avoid any training bias, three times random cross-validation is performed by repeated random subsampling of non-match pairs with replacement. In order to maintain consistency across different experiments, the performance of all the algorithms is reported on the remaining 50% test set. The effect of fingerphoto

Table 6.3: Performance comparison of different feature extraction and matching algorithms in terms of EER (%). The preprocessing pipeline (segmentation + enhancement) is consistent across all feature extractors.

	Gallery	Probe	MCC	Comp Code	NBIS	Minacc [115]	ScatNet + ℓ_2	ScatNet + NN	ScatNet + RDF
Exp. 1	White	White Outdoor	16.50	19.07	50.00	23.34	16.95	3.27	3.00
	Indoor	Natural Indoor	12.36	16.22	49.87	22.04	20.39	6.50	3.21
	(WI)	Natural Outdoor	17.03	21.40	49.99	22.96	20.59	5.34	2.11
Exp. 2	Resolution (13 MP)	5 MP	10.35	14.32	50.00	13.65	15.72	7.53	5.23
		8 MP	10.01	13.03	49.98	16.21	17.93	5.42	4.74
		16 MP	48.48	10.84	50.00	21.38	17.07	3.73	2.98

matching performance is studied under two experimental variations, *Exp.1* and *Exp.2*, explained as follows:

- **Background-Illumination (*Exp.1*):** With the assumption that White Indoor (*WI*) images are the most stable capture of fingerphoto images, *WI* subset is fixed as gallery while the other three subsets $\{WO, NI, NO\}$ are used as probe images, independently. *WI-WO* experiment shows the performance of fingerphoto matching under the influence of illumination, *WI-NI* experiment shows the matching performance in the presence of background variation, and *WI-NO* shows the impact of illumination and background together on the matching performance.
- **Resolution (*Exp.2*):** With fingerphoto images captured at 13MP in White Indoor (*WI*) environment, as the standard resolution gallery images, we perform experiments for cross resolution matching. Three different probe sets, with resolution of captured fingerphoto images at 5MP, 8MP, and 16MP, are used. Thus, two probe sets have resolution lower than the gallery while one probe set has fingerphoto images with a higher resolution than gallery.

Thus, for both *Exp.1* and *Exp.2*, the same gallery images are used while the probe sets are varied to study the impact of capture variations on the matching performance.

6.3.2 Comparison Algorithms

For comparative analysis, neural network (NN) and normalized ℓ_2 distance based matching are used. Also, the proposed pipeline is compared with various other algorithms and matching pipelines, as shown in Table 6.2 and explained as follows:

1. Minutiae based:

- **MCC:** Seg + Enh + Minutiae Cylinder Code (MCC) features [142] + matching using VeriFinger SDK
- **NBIS:** *nfseg* based Seg + No Enh + *mindtct* based minutiae features + *bozorth3* based matching

2. Non-Minutiae based:

- **CompCode:** Seg + Enh + CompCode features [216] + ℓ_1 -distance based matching

3. ScatNet based:

- **Minaee and Wang** [115]: No Seg + No Enh + ScatNet + PCA + SVM based matching
- **ScatNet + ℓ_1 :** Seg + Enh + ScatNet + PCA + ℓ_1 based matching
- **ScatNet + NN:** Seg + Enh + ScatNet + PCA + neural network based matching
- **ScatNet + RDF** (proposed): Seg + Enh + ScatNet + PCA + RDF based matching

6.3.3 Implementation Details

For region covariance based saliency map extraction, $\alpha = \sqrt{2}$ is used. From all the images, the segmentation algorithm yields a fixed-size window of 400×840 as the output. The second-order ScatNet representations used in this research yield a feature representation of length 1,097,250 per fingerphoto image. Due to the high dimensionality of representation, PCA [217] is applied. In our experiments, 99% Eigen energy preserving PCA yields a succinct representation of fingerphoto of length 95, which is provided as input to the classifier. In the RDF implementation, $T = 1000$ independent decision trees are created with a bootstrap ratio of $r = 0.66$. At every node in a decision tree, $m = 1,048$ features are randomly sampled from $M = 1,097,250$ features. With a

Matlab 2014 implementation on a 2.66GHz dual-core processor with 16GB RAM, authenticating a fingerphoto takes, on an average, less than two seconds.

6.3.4 Proposed Pipeline Performance Analysis

The proposed matching pipeline is validated with the [ISPF2](#) and the results are tabulated in Tables [6.3](#) and Figures [6-10](#) to [6-11](#). As briefly described, the comparisons are made across classifiers and across different variations. Analyzing the results in a column-wise fashion gives comparison of the proposed algorithm with existing algorithms. It can be distinctly observed that the ScatNet features learnt with a supervised classifier provide better performance as compared to other algorithms, suggesting the importance of training a classifier for verification. ScatNet + [RDF](#) and ScatNet + NN provide [EER](#) in the range of 2.11 – 7.53%, while ScatNet + ℓ_2 distance matching provides [EER](#) in the range 15.72 – 20.59%. Further, it can be observed that [RDF](#) performs better than neural network under all capture variations. We also observed that there is very negligible (almost zero) deviation in the performance of the proposed algorithm across the cross-validation experiments. With respect to existing feature descriptors, CompCode and [MCC](#), ScatNet + [RDF](#) provides up to 20% improvement in [EER](#). This can be attributed to the rich feature representation obtained using the high-frequency information in ScatNet, and also its affine invariance property.

We next evaluate the performance across different variations - *Exp. 1* and *Exp. 2*. From Table [6.3](#), it can be inferred that the proposed matching pipeline with [RDF](#) classifier provides the best performance across different variations. The consistently low error rates of (2.1% – 3.2%) in the first three rows of Table [6.3](#) show that the proposed algorithm is robust against the variations in background (*White Outdoor and Natural Outdoor*) and illumination (*Natural Indoor and Natural Outdoor*). In the cross resolution matching experiments (*Exp. 2*) summarized in the last three rows of Table [6.3](#), we can observe that matching high-resolution images ([EER](#): 2.98%) yields slightly better results than matching low-resolution images ([EER](#): 5.23%). It is important to note that different individuals have different kinds of phones and the camera resolution across different phones also varies. This result shows that using a phone camera with lower resolution leads to a smaller reduction in accuracy but not very significant - matching 8 MP with 13 MP fingerphotos gives 4.74% [EER](#) while matching 5 MP with 13 MP yields 5.23% [EER](#).

Table 6.4: Understanding the effect of resolution on fingerphoto matching with *Exp.2*. The results are reported in terms of **EER** (%).

Algorithm	Gallery	Probe	Enhancement	
			Without	With
MCC	13 MP	5 MP	11.95	10.35
		8 MP	12.61	10.02
		16 MP	46.32	48.48
CompCode	13 MP	5 MP	15.69	14.32
		8 MP	15.14	13.03
		16 MP	13.15	10.84
ScatNet + ℓ_2	13 MP	5 MP	15.44	15.72
		8 MP	19.48	17.93
		16 MP	26.29	17.07
ScatNet + NN	13 MP	5 MP	7.41	7.53
		8 MP	5.59	5.42
		16 MP	4.22	3.73
ScatNet + RDF	13 MP	5 MP	9.67	5.23
		8 MP	11.71	4.74
		16 MP	9.48	2.98

6.3.5 Minitiae and Non-minutiae Matching Pipeline

Analyzing the performance of **MCC** descriptor shows that in *Exp. 1* yields poor results for outdoor images, as minutiae extraction is highly spurious due to the capture variations. Further in *Exp. 2*, the performance of **MCC** descriptor drops suddenly when the resolution of probe images is higher than the gallery images. This can be attributed to the observation that **MCC** descriptor constructs fixed radius cylinders around each minutia to extract its descriptor. When the resolution of the probe image becomes higher than that of the gallery, no minutia is found within the constructed cylinder, and hence the matching performance drops. Dynamic prediction of the cylinder parameters can be performed for **MCC** descriptor, however, it is an independent research challenge.

The results of CompCode descriptor show that it is better at handling resolution variations as compared to handling environmental noise. However, ScatNet + **RDF** is more robust and is little affected by the capture variations. Overall, we observe that partial fingerphotos and images with *out-of-focus* regions are better handled by ScatNet based matching algorithm. Similarly, using **NBIS** based pipeline also reduces the fingerphoto matching performance. Based on manual observation, we found that *nfseg* has a very high failure rate in segmenting the fingerphoto fore-

ground region. Further, *mindtct* is not trained for extracting minutiae from fingerphoto images, thus, we observed more than 35% of the images to have zero minutiae extracted. Because of these reasons, the public NBIS matcher from NIST cannot be used for matching smartphone captured fingerphoto images.

6.3.6 Existing Algorithms

The closest work in the literature by Minaee and Wang [115] provides an EER in the range of 15 – 20%. In their original research, they had shown experimental results on PolyU HRF dataset, which has high resolution images captured in a highly controlled environment without the presence of environmental variations. Thus, evaluating their approach in a much challenging task of matching smartphone captured fingerphoto images in a challenging ISPF2 yields worse performance. However, the proposed pipeline has better preprocessing stages in terms of foreground segmentation and enhancement catering to handle challenging capture mechanisms.

6.3.7 Effectiveness of Enhancement

We next study the importance of ridge-valley enhancement algorithm in the proposed fingerphoto verification pipeline. While segmentation is required to remove the background noise, the impact of enhancement in the overall matching performance is studied and summarized in Fig. 6-7 to 6-9, Table 6.4, and the ROC curves shown in Fig. 6-10 and Fig. 6-11. To set the benchmark, the matching results of only the segmented images without any enhancement is shown for both the experimental protocols, *Exp. 1* and *Exp. 2*. Without any enhancement, ScatNet + RDF produces EER in the range of 2.8 – 5.6% for *Exp. 1* experiments and in the range of 9.5 – 11.7% for *Exp. 2*. After applying the proposed enhancement algorithm, the overall matching performance improves and the EER reduces to 2.1 – 3.2% for *Exp. 1* and 3.0 – 5.2% for *Exp. 2*. As observed from Fig. 6-4, the enhanced image has noisy frequencies removed, thereby, allowing better ScatNet features to be extracted. Thus, the proposed ridge-valley enhancement in the preprocessing is essential to improve the matching performance of ScatNet + RDF. The results also show that the enhancement algorithm has minimal influence on CompCode features. Similarly, as shown in Table 6.4, enhancement algorithm has minimum influence on MCC descriptor during cross-

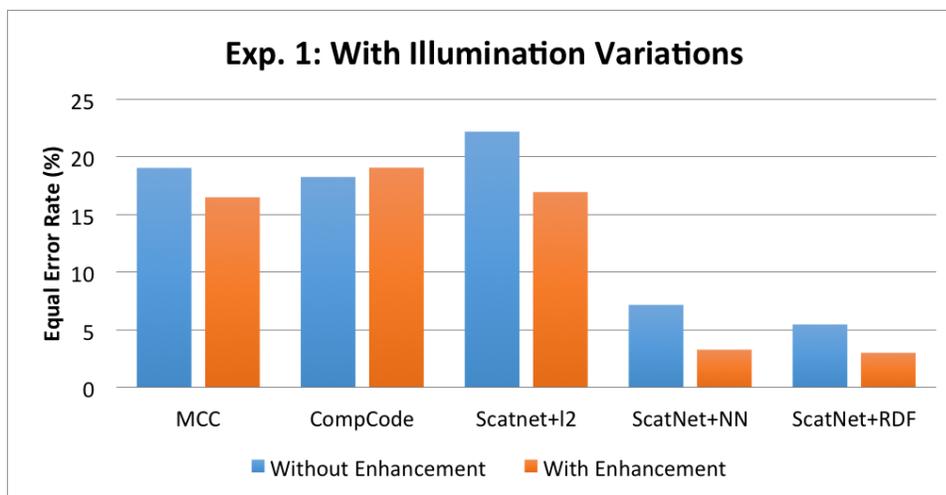


Figure 6-7: Analyzing the effect of enhancement on matching fingerphoto images with illumination variations.

resolution matching.

Another phenomenon is predominantly observed in Fig. 6-7 to 6-9, where CompCode features perform better before enhancement. CompCode features aims to extract the Gabor (variant) filter response of an image which allows the signals of a specific frequency band. During the enhancement phase, as image sharpening and smoothing are performed, certain frequency signals are removed that is essential for CompCode features. On the other hand, ScatNet extracts signals from all the frequency bands in a tree-like fashion and combines them. Therefore, enhancement stage has little impact on the performance of ScatNet based matching. Further, ScatNet + NN provided better results than ScatNet + RDF before enhancement in the experiments with resolution variation . While there were no theoretical evidence to explain this pattern, it is hypothesized that the neural network architecture is robust against the proposed enhancement procedure. It can be observed from Table 6.4 that ScatNet + NN provides almost similar performance before and after enhancement.

6.4 Summary

This research work summarizes the major challenges associated with matching fingerphoto images captured using a smartphone camera. A novel ScatNet based affine invariant fingerphoto represen-

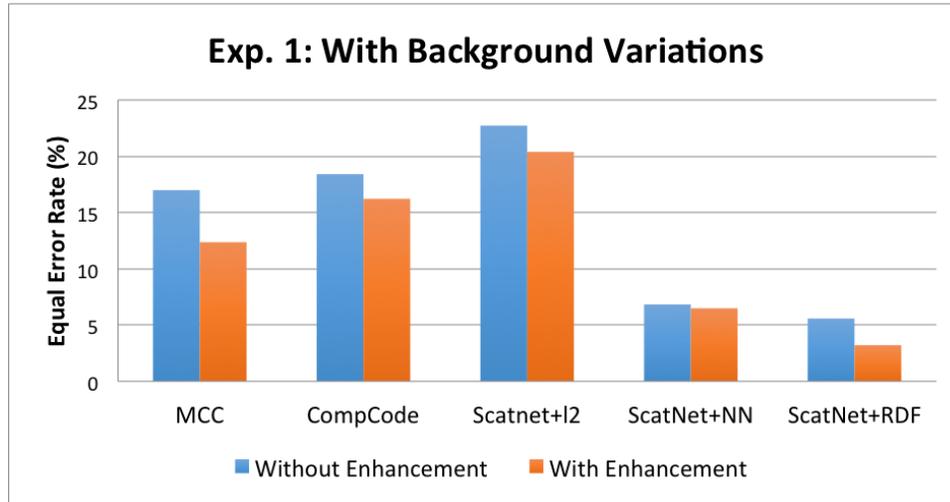


Figure 6-8: Analyzing the effect of enhancement on matching fingerphoto images with background variations.

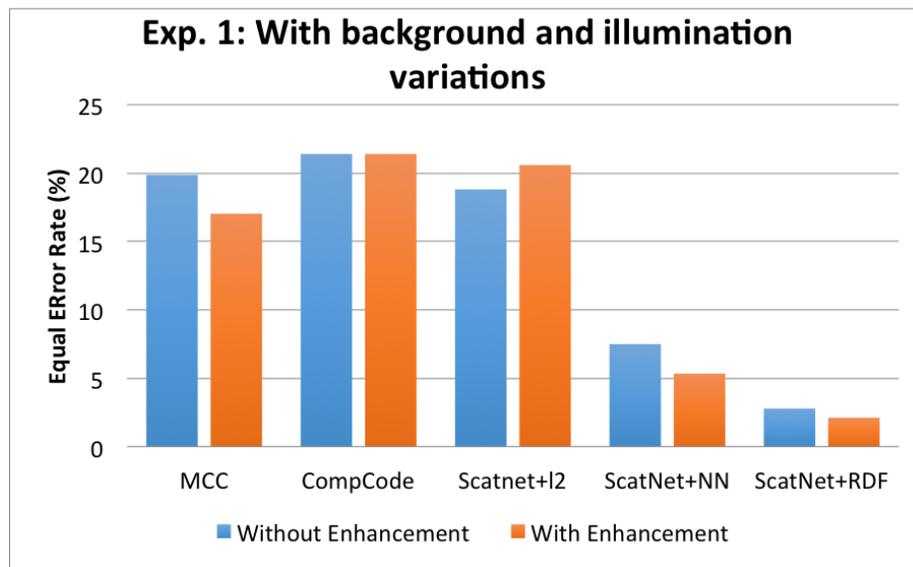


Figure 6-9: Analyzing the effect of enhancement on matching fingerphoto images with background and illumination variations.

tation is proposed. Feature matching is performed using a RDF classification based approach and compared with minutiae based and CompCode based methods. A fingerphoto segmentation and enhancement algorithm is proposed to aid the matching process. Three different challenges affected the performance of fingerphoto matching: (i) background variation, (ii) environmental il-

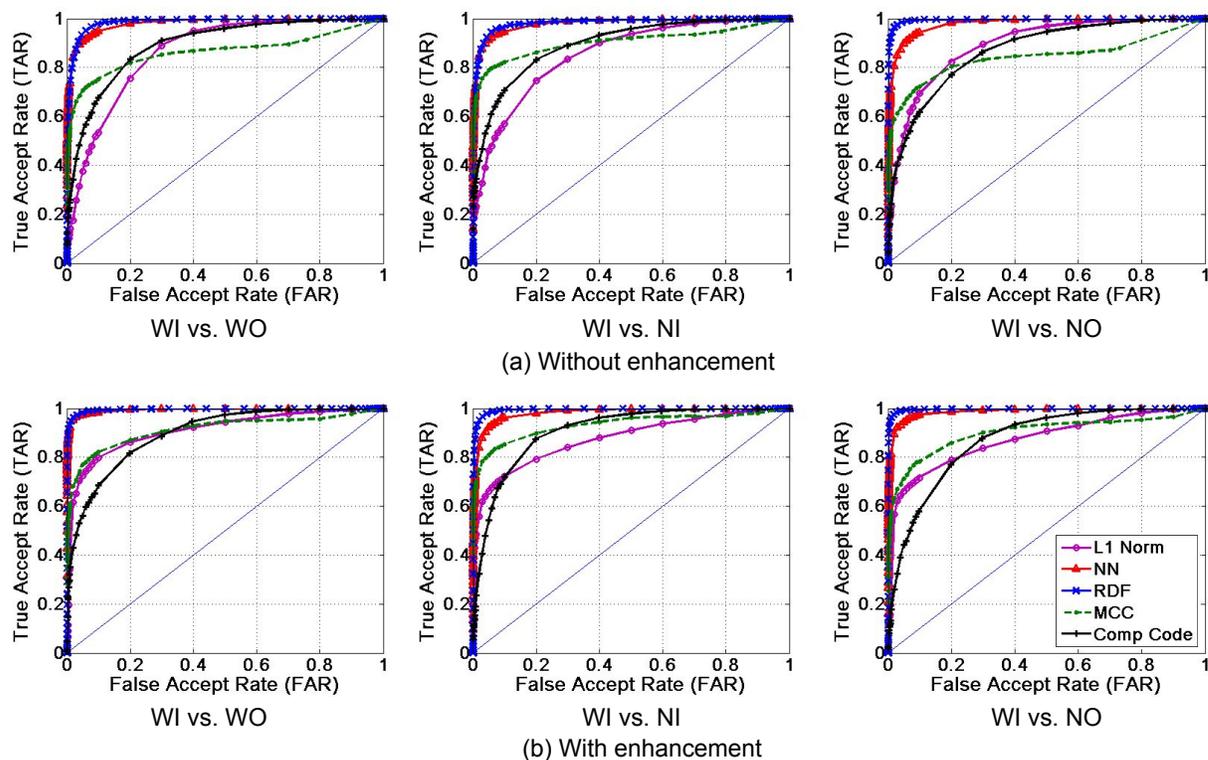


Figure 6-10: ROC curves illustrating the effect of the proposed enhancement algorithm on the verification performance of different fingerprint recognition algorithms with background and illumination variations (Exp. 2): (a) without enhancement and (b) with enhancement.

illumination, and (iii) resolution of camera. To be able to study and address the different challenges, [ISPF2-v2](#) is created and made publicly available for research purpose. The database consists of three sets having 304 classes from 76 subjects with 16,800 images. The performance of the proposed ScatNet + [RDF](#) based matching pipeline is compared with other successful methods in literature such as neural networks, minutiae based [MCC](#), and CompCode features. Also, two different enhancement algorithm such as image based enhancement and [Local Binary Pattern \(LBP\)](#) based enhancement are studied and compared. The experimental results show a considerable performance improvement of the proposed ScatNet + [RDF](#) algorithm over the existing algorithms in different experimental settings.

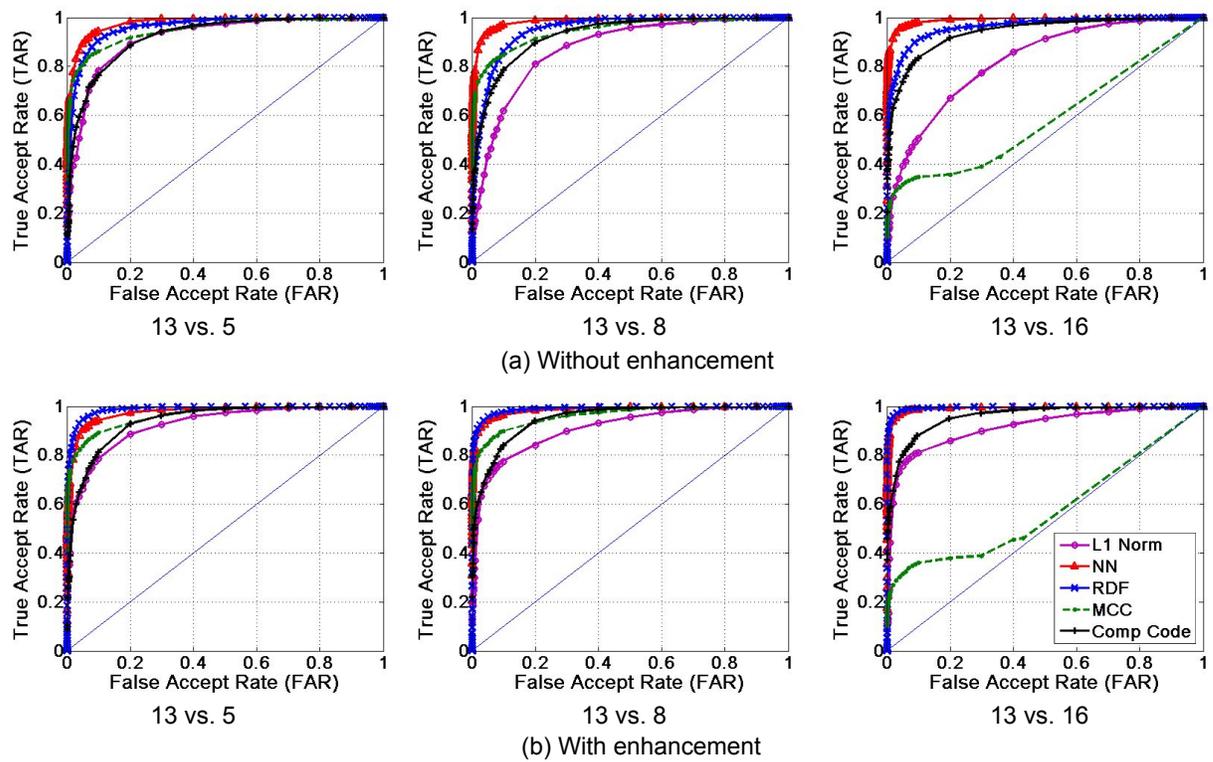


Figure 6-11: ROC curves illustrating the effect of the proposed enhancement algorithm on the verification performance of different fingerphoto recognition algorithms with resolution variations (Exp. 2): (a) without enhancement and (b) with enhancement.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Conclusion and Future Research

Supported by the recent advancements in technology and data handling capacity, automated fingerprint recognition systems are extensively used in many civil and law enforcement applications such as access control systems, financial transaction systems, and cross-border security at immigrations. Inked fingerprint or live-scan fingerprints, used in these applications, are captured in a highly controlled environment producing high clarity, continuous ridge valley patterns with very little or no background variations. However, as we move into more uncontrolled fingerprint capture environments such as latent fingerprints or contactless capture of fingerphoto images, the obtained ridge valley pattern is very different from the traditional capture mechanisms. The ridge flow becomes highly discontinuous, a lot of background noise is introduced during capture, and only a partial fingerprint is obtained while the rest is either lost or smudged during capture. These underpinned challenges and capture variations prohibit the advocacy of existing automated algorithms as the extraction of minutiae and other features become very challenging. To be able to uniquely identify a fingerprint, it is imperative to comprehensively represent its features despite the available noisy and incomplete ridge information. Therefore, in this dissertation we have made an attempt to build automated solutions for matching the uncontrolled capture variants of fingerprint images by proposing novel feature learning and representation algorithm in an unsupervised fashion.

7.1 Conclusion

In this dissertation, we have addressed some of the challenges pertinent to uncontrolled capture of fingerprint images. The major research contributions are discussed below:

- **Created benchmark public dataset:** The primary challenge in actively pursuing to address the important research challenges in uncontrolled fingerprint matching, is the lack of publicly available large scale benchmark dataset. The performance of existing algorithms is significantly affected when the fingerprints are captured with diverse acquisition methods (multi-session, multi-resolution, highly varying background, and with latent fingerprints). A large amount of time and efforts of this dissertation was devoted in creating two large fingerprint datasets: (i) [MOLF](#) database with 19,200 fingerprint images from 1000 classes and (ii) [ISPFV-v2](#) containing 16,800 images from 300 classes. Detailed protocols are established corresponding to various matching scenarios and the entire dataset along with their baseline results are made publicly available for research.
- **Latent fingerprint segmentation:** Typically, the first and important stage in the matching pipeline for fingerprints is the segmentation of foreground ridge like patterns from any kind of background noise. We proposed an automated segmentation algorithm for latent fingerprint foreground segmentation and three contributions of the proposed algorithm are: (i) to propose visual saliency as the major contributing feature in segmentation along with a framework for combining five different categories of features for automatic latent fingerprint segmentation, (ii) a feature selection technique using modified RELIEF formulation for dynamically analyzing the usefulness of multiple category features for a given quality of latent fingerprint, and (iii) a [SIVV](#) based metric to measure the effect of the segmentation algorithm without the requirement to perform the entire matching process.
- **Latent fingerprint minutiae extraction:** The performance of an automated latent fingerprint identification is limited due to imprecise automated feature (minutiae) extraction, specifically due to noisy ridge pattern and presence of background noise. Unsupervised feature learning paradigm is adopted to learn robust descriptors from noisy local latent fingerprint patches. To learn better feature descriptors, a novel $\ell_{2,1}$ based regularization method

was proposed on two basic architectures - autoencoders and [RBM](#), to construct group sparse autoencoders ([GSAE](#)) and class sparsity specific restricted Boltzmann machine ([cssRBM](#)). Using the learnt descriptor, minutiae detection is formulated as a binary classification problem, where each local patch is classified as a minutiae or a non-minutiae containing patch. To the best of our knowledge, this was the first automated minutiae extraction algorithm for latent fingerprint images.

- **Smartphone fingerphoto matching:** Contactless capture of fingerprint image using smartphone camera highly reduces the clarity between ridge and valley patterns, thus making the minutiae extraction very spurious. Further, during fingerphoto capture certain variations are introduced that are not available in live-scan fingerprint images, such as highly varying background information, random environmental illumination, and changing resolution of the smartphone camera during capture. In this dissertation, an end-to-end automated solution was proposed which is robust against these capture variations in verifying fingerphoto images. Here again, it is shown that visual saliency is a major contributing feature in segmenting the foreground region from the varying background. Further, as minutiae extraction is highly spurious in fingerphoto images, a novel feature descriptor using deep scattering networks was proposed to represent fingerphoto images. It was experimentally observed that high resolution fingerphoto images (as high as 16 megapixels) captured in uncontrolled outdoor illumination with random background noise eventually provided the best verification performance, using the proposed algorithm.

7.2 Future Research

Some challenging problems that could further advance the study in matching fingerprint images captured using uncontrolled mechanisms are discussed as follows:

- The latent fingerprint segmentation algorithm could be made more robust and generalizable by learning the ridge features in an unsupervised fashion. Intuitions obtained from the proposed feature selection algorithm and the classifier could be transferred to a deep learning algorithm to obtain improved segmentation algorithm.

- The **MOLF** dataset, created in this dissertation, consists of latent fingerprint lifted from only plain ceramic tile surface. Characterizing different types of surfaces for segmenting latent fingerprint images is a potential research problem. This warrants the study of latent fingerprint lifted from multiple surfaces and the noise introduced by each type of surface while lifting the latent fingerprint.
- Some forensic experts are highly experienced and skilled to mark minutia features in latent fingerprints despite the poor quality of the ridge patterns. It is an interesting research challenge to understand the human expert cognition while working on a latent fingerprint, such that, the important observations can be translated to an improved automated algorithm.
- The performance of minutiae extraction from both latent fingerprints and fingerphoto images can be further improved by learning deeper and more local features from ridge patterns. The proposed $\ell_{2,1}$ regularization can be applied to deep networks such as Convolution Neural Network (CNN) to learn complex representation.
- Implementing the solution for fingerphoto matching as an application for Android operating system or for iOS, is a potential direction to proceed to make the proposed solution available as a consumer product.

Appendix A

Results on Standard Image Datasets

To showcase the effectiveness and compare with existing (similar) approaches, we demonstrate the results on two standard databases namely, MNIST [8] and CIFAR-10 [218].

A.1 Standard Image Datasets

The properties of the datasets are summarized in Table A.1 and a short description of the standard image datasets are provided below:

- **MNIST:** It is a handwritten digit classification problem from gray-scale images of size 28×28 . The dataset consists of 60,000 training images with approximately equal number of images from each class. 50,000 of these images are used for training while the remaining 10,000 are randomly held out as the validation set. Hyper-parameters of the model are tuned using the validation set. The architecture is finetuned using the training and validation sets combined together, whereas the hyper-parameters of the classifier are learnt only using the validation set. Classification results are shown on a separate test set of 10,000 images. An architecture of [784 500 1000] network is trained. The learnt weights and momentum are updated as explained by Srivastava et al. [179]. A 2ν -SVM classifier [219] with degree 5 polynomial kernel is used as the classifier. Sample images from the MNIST database are presented in Figure A-1(b).
- **CIFAR-10:** It is a labelled subset of the 80 million tiny images dataset [220]. It contains

Table A.1: Overview of the standard image data sets used in the experiments.

Dataset	Image size	Train set	Test set
MNIST [8]	28×28 gray (784)	60,000	10,000
CIFAR-10 [218]	32×32 color (3072)	60,000	10,000

colored images corresponding to ten different object classes, and the size of the images are 32×32 . The ten classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. 6000 images are available per class, with 50,000 images for training and 10,000 images for testing. The dataset is split into five training sets and one testing set, each containing 10,000 images. The testing set consists of exactly 1000 images from each class. Even though the number of classes is not large, the images in the database are largely unconstrained in viewpoint and occlusion, thereby making the database challenging. A network of architecture [3072 500 1000 2000] is learnt for the CIFAR dataset. The remaining hyper-parameters are the same as we used for the MNIST dataset. Sample images from the CIFAR-10 database are presented in Figure A-1(a).

It is important to note that the protocols followed for all three databases are standard benchmark protocols (including train-test partitioning). Therefore, the results of the proposed formulation can be directly compared with the results available in literature.

A.2 Evaluation Metrics

The standard image datasets have different metrics designed to evaluate the performance of algorithms. On MNIST dataset, error rate (%) is used as performance metric which measures the percentage of misclassifications. For instance, an error rate of 0.68 means 68 images out of 10,000 images are misclassified. For CIFAR-10, the accuracy (%) of correct classification is used as a metric to evaluate the performance of algorithms.



Figure A-1: Sample images from the CIFAR-10 and MNIST databases without any preprocessing.

A.3 Results Using GSAE

In the experimental setup, each of the hidden layers is pre-trained in a greedy layer-wise fashion, and the overall network is further fine-tuned using the training set. A 2ν -SVM based 10-class classifier is trained for the respective datasets. The effectiveness of the proposed algorithm is demonstrated with the following regularizers:

1. **KLD**: In the standard architecture, only KL-divergence based regularization is used during both pre-training and fine-tuning.
2. **GSAE**: This utilizes only the proposed $\ell_{2,1}$ norm to introduce group-sparsity into the training loss function. In this architecture, $\ell_{2,1}$ -norm based supervision is used during both greedy layer-wise pre-training, as well as, the overall architecture fine-tuning.
3. **KLD + GSAE**: This architecture shows that the proposed $\ell_{2,1}$ -norm can be used to complement other existing regularization methods. In this method, greedy layer-wise pre-training is performed using KL-divergence based sparsity regularizer (without class labels). $\ell_{2,1}$ -norm based supervision is performed only during fine-tuning. This hybrid architecture suggests that the existing pre-trained architectures can be fine-tuned using the proposed regularizer to

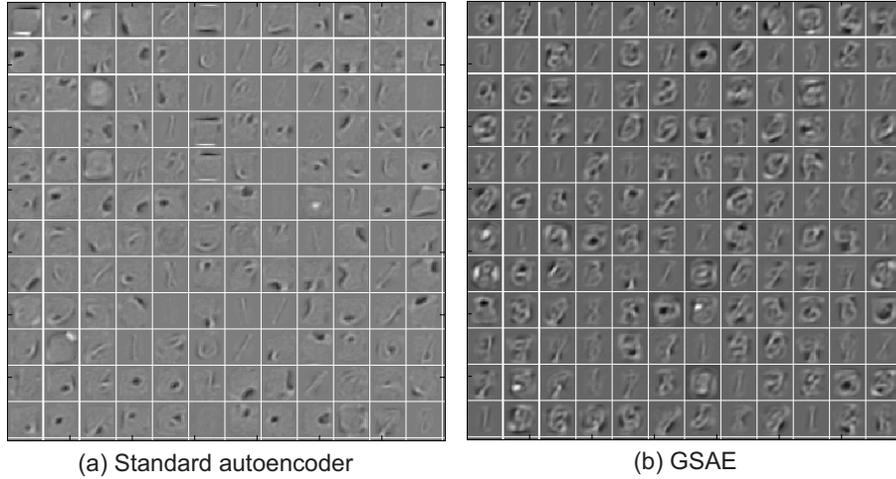


Figure A-2: Visualization of the features learnt in the first hidden layer of the autoencoder on MNIST dataset with (a) standard autoencoder using only KL-divergence based sparsity, (b) proposed GSAE learning algorithm.

obtain an improvement in performance.

Table A.2: The performance of the proposed GSAE algorithm using different regularizers on the standard image data sets. The architecture of the autoencoder used for each dataset is also provided.

Dataset	Architecture	KLD	GSAE	KLD + GSAE
MNIST (Error Rate %)	[784 500 500]	1.71	1.19	1.10
CIFAR-10 (Accuracy %)	[3072 2000 2000]	74.3	76.8	77.4

Table A.2 shows the results of the proposed GSAE algorithm on the three standard image datasets along with the performance of some existing algorithms. On the MNIST dataset, apart from the aforementioned experimental setup, we compare our best reported result along with the best reported results of variants of autoencoder proposed in the literature. The existing variants of autoencoder algorithms that are compared are: Marginalized Denoising AutoEncoder (MDAE) [221], Stacked AutoEncoder (SAE) [201], Stacked Denoising AutoEncoder (SDAE) [201], Contractive AutoEncoder (CAE) [222], and Autoencoder Scoring [223]. The following are some key observations from the set of experiments performed:

- In all three datasets, a similar trend can be observed across the architectures and it can be found that KLD + GSAE provides the best performance. This confirms that the group sparsity constraint assists in learning improved features for the classification tasks.

- Figure A-2 shows the obtained hidden layer visualizations of the autoencoder trained using KL-divergence and the proposed GSAE algorithm. It can be visually observed that GSAE algorithm learns better descriptive features that improve the classification performance.
- On the MNIST dataset, Table A.3 shows comparative performance of the proposed algorithm along with variants of autoencoder, as reported in the literature. It can be observed that the proposed GSAE provides comparable performance with the existing autoencoder variates. However, it is to be noted that the architectures across these variates are different. For autoencoder variates, the architecture providing the best performance is reported.
- In the proposed formulation, λ controls the dominance of $\ell_{2,1}$ regularization during learning. In the literature, Vincent et. al. [201] have shown that the regularization constant could strongly influence the learnt features. Therefore, we have performed experiments with varying λ in the range of 0 to 1 with a varying step sizes, i.e. $\lambda = \{(0.00001 : 0.00001 : 0.0001), (0.0001 : 0.0001 : 0.01), (0.01 : 0.01 : 0.1), (0.1 : 0.1 : 1)\}$. On the MNIST dataset, the lowest error rate of 1.10 is obtained with $\lambda = 0.08$. Similarly, for other two databases, the best performing results are obtained with $\lambda = 0.09$ and $\lambda = 0.08$ respectively. With group sparsity, we observe that smaller values of λ yields better results compared to higher values and the optimal performance is obtained for the values in the range of 0.05 - 0.1.
- The aim of any supervised classifier is to learn a function that maps a learnt feature representation to a set of appropriate classes. In machine learning paradigm, Wolpert [224] formulated the "No free lunch" theorem stating that, under noise-free environment there is no prior for the distinction in supervised learning algorithms based on training-set error. According to the understanding of this theorem, it is very challenging to hypothesize that a particular supervised classifier is going to perform better than other classifiers, without performing experimental evaluation. Hence, in comparison with 2ν -SVM, we evaluate the performance of other popular classifiers, the softmax classifier, multilayer neural network (with 2 hidden layers), and classic SVM classifier. On the MNIST dataset, the proposed algorithm with 2ν -SVM classifier provides the lowest error rate of 1.10 which is at least 0.75% better than other three classifiers.

Table A.3: Comparing the performance of different variates of autoencoder based algorithms and the proposed algorithm, GSAE, on the MNIST dataset [8].

Algorithm	Error %
MDAE [221]	1.29
SAE [201]	1.40
SDAE [201]	1.28
CAE [222]	1.14
Autoencoder Scoring [223]	1.27
Proposed	1.10

It is to be noted that state-of-the-art performance for the datasets used in this research are: (i) MNIST is 0.21%, obtained using ConvNet architecture and dropconnect regularization [184], (ii) CIFAR-10 is 91.78%, obtained using deeply supervised ConvNet architecture [225], and (iii) SVHN is 1.92%, obtained using deeply supervised ConvNet architecture [225]. We would like to mention that the main motivation is to show that adding group sparse regularizer can improve the performance of stacked autoencoder based feature representation. The results presented in this section showcase that $\ell_{2,1}$ norm, solved via majorization-minimization approach, helps in improving the classifier performance.

A.4 Results using cssDBM and cssDBN

We demonstrate the results of the proposed supervised formulations with three different regularizers: standard (ℓ_2 -norm), dropout, and dropconnect. As explained in Section 5.4.4, the proposed class sparsity signature is applied in continuation with other regularizers on the weight matrix. The results are compared with the respective DBM and DBN architectures without the class sparsity signatures.

- **MNIST:** Table A.4 summarizes the error (%) obtained on the MNIST database using the proposed and existing DBM and DBN formulations respectively. The best error of 0.53% is obtained using DBM and 2ν -SVM classifier, with dropout + class sparsity regularization. Creating sparse hidden representation using dropout and further constraining them using a class label based weighted ℓ_2 norm helps the architecture learn better discriminative features.

This validates our initial hypothesis that the sparsity signature at the hidden layer for a given class would be similar. A similar architecture without using class sparsity yields an error rate of 0.78%. It can be observed that on all the architectures, under all the configurations, class sparsity improves the classification performance. As expected, the worst performing configuration with 1.19% error rate is DBN with only ℓ_2 norm, without class sparsity, and softmax classifier. Figure A-3 shows sample visualizations learnt using the standard RBM and the proposed RBM. It can be visually observed that the features of cssRBM are more informative than standard RBM.

- **CIFAR-10:** Similar to MNIST database, Table A.5 summarizes the results of DBM and DBN with different regularizes, with and without class sparsity signatures and different classifiers. Regardless of the regularization utilized, the proposed cssRBM based architectures outperform the traditional DBM and DBN on the CIFAR-10 database. The best performance of 82.9% is achieved by cssDBN using dropoutconnect [184] followed by cssDBM at 82.8% with dropout/dropconnect regularization. The best accuracies achieved by the traditional DBM and DBN architectures are 79.5% and 75.8% respectively. We also observe that 2ν -

Table A.4: Classification error on the MNIST database obtained using Deep Boltzmann Machines and Deep Belief Networks. The table shows the error (%) of the classifier, where 0.53 means 53 out of 10,000 images are misclassified.

Algorithm	Regularization	Class Sparsity Signature	Classifiers	
			Softmax	2ν -SVM
Deep Boltzmann Machines	Standard (ℓ_2 -norm)	without css	0.95	0.94
		with css	0.87	0.83
	Dropout	without css	0.79	0.78
		with css	0.61	0.53
	Dropconnect	without css	0.87	0.79
		with css	0.67	0.58
Deep Belief Networks	Standard (ℓ_2 -norm)	without css	1.19	1.03
		with css	1.02	0.95
	Dropout	without css	0.92	0.89
		with css	0.89	0.81
	Dropconnect	without css	0.95	0.90
		with css	0.90	0.82

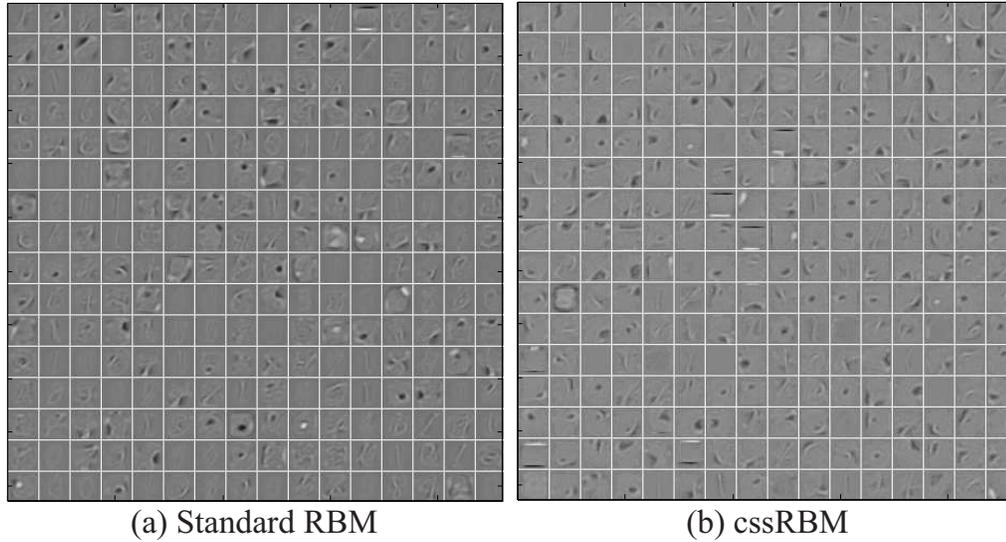


Figure A-3: Sample representation showing the features learnt in MNIST database using (a) standard RBM and (b) proposed cssRBM.

Table A.5: Results on CIFAR-10 database using Deep Boltzmann Machines and Deep Belief Networks. The table shows the matching accuracy (%) of the classifier.

Algorithm	Regularization	Class Sparsity Signature	Classifiers	
			Softmax	2ν -SVM
Deep Boltzmann Machines	Standard (ℓ_2 -norm)	without css	75.2	76.1
		with css	77.9	79.3
	Dropout	without css	78.6	79.5
		with css	79.5	82.8
	Dropconnect	without css	77.2	77.4
		with css	79.9	82.8
Deep Belief Networks	Standard (ℓ_2 -norm)	without css	74.8	75.3
		with css	77.6	79.2
	Dropout	without css	75.4	75.8
		with css	79.5	82.6
	Dropconnect	without css	74.9	75.5
		with css	79.8	82.9

SVM classifier performs better than the softmax classifier. The best performance achieved by the softmax classifier is 79.9% which is approximately 3% lower than the results of 2ν -SVM classifier. Regularization yields noticeable improvements in the results where the traditional versions of the DBM and DBN architectures are only able to achieve up to 76.1% accuracy.

- **Comparison with State-of-the-art Algorithms:** The state-of-the-art performance on these two datasets are: (i) 0.21% on the MNIST database obtained using Convolutional Neural Network (ConvNet) architecture with dropconnect regularization [184] and (ii) 91.78% on the CIFAR-10 obtained using deeply supervised nets based on ConvNet architecture [225]. It is to be noted that most of the best performing algorithms on these databases use deep ConvNet based architectures. In this research, we have proposed the class sparse regularization approach for RBM based deep networks: DBN and DBM. We have compared the performance of these architectures, with and without the application of class sparsity and experimentally showed an improved performance with class sparsity.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

- [1] M. Králík and L. Nejman. Fingerprints on artifacts and historical items: examples and comments. *Journal of Ancient Fingerprints*, 2007. [XV, 2](#)
- [2] Evaluation of Latent Fingerprint Technologies. <http://www.nist.gov/itl/iad/ig/latent.cfm>. [XV, XVII, 4, 37](#)
- [3] Ajay Kumar and Yingbo Zhou. Contactless fingerprint identification using level zero features. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 114–119, 2011. [XVI, 11](#)
- [4] Fingerprint minutiae from latent and matching tenprint images. *NIST Special Database 27*, 2010. [XVII, XIX, XX, 26, 30, 47, 69, 71, 89, 104, 105, 125](#)
- [5] NBIS (NIST Biometric Image Software). *Developed by National Institute of Standards and Technology*, <http://www.nist.gov/itl/iad/ig/nbis.cfm>, . [XVII, 26, 28, 39, 51, 56, 58, 125](#)
- [6] H.-S. Choi, M. Boaventura, I. A. G. Boaventura, and A. K. Jain. Automatic segmentation of latent fingerprints. In *IEEE International Conference on Biometrics: Theory Applications and Systems*, 2012. [XVII, 25, 26, 27, 28, 41, 79](#)
- [7] Jiangyang Zhang, R. Lai, and C.J. Kuo. Adaptive directional total-variation model for latent fingerprint segmentation. *IEEE Transactions on Information Forensics and Security*, 8(8): 1261–1273, 2013. [XIX, 25, 96, 97](#)
- [8] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. *AT&T Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2010. [XXV, 161, 162, 166](#)

- [9] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition, 2nd edition*. Springer-Verlag, 2009. [1](#), [4](#), [28](#), [32](#), [60](#), [70](#), [72](#), [80](#), [121](#)
- [10] J. A. Bailey. Fingerprints from antiquity: Casting impressions on elamite brick. *Minutiae*, 2003. [1](#)
- [11] H. Cummins. Ancient finger prints in clay. *The Scientific Monthly*, 1941. [1](#)
- [12] G. Lambourne. The fingerprint department. *The Fingerprint Story, Harrap of London*, pages 70–72, 1984. [1](#)
- [13] Capsule reveals cream of Roman society. *BBC News 2003*. [1](#)
- [14] Shashikant Oak. *Naadi Predictions*. Diamond Pocket Books Pvt Ltd, 2014. [1](#)
- [15] F. Galton. Personal identification and description 1. *Nature*, 21:173–177, 1888. [2](#)
- [16] Vijay Sathe. The world’s most ambitious id project (innovations case narrative: India’s project aadhaar). *innovations*, 6(2):39–65, 2011. [2](#)
- [17] Lisa M Seghetti and Stephen R Viña. Us visitor and immigrant status indicator technology (us-visit) program. DTIC Document, 2005. [2](#)
- [18] Adam Warren and Elizabeth Mavroudi. Managing surveillance? the impact of biometric residence permits on uk migrants. *Journal of Ethnic and Migration Studies*, 37(9):1495–1511, 2011. [2](#)
- [19] P. Komarinski and Ed. Automated Fingerprint Identification Systems. *Elsevier Academic Press*, 2001. [4](#)
- [20] H. C. Lee and R. E. Gaensslen. Methods of latent fingerprint development. *Advances in Fingerprint Technology, second edition, CRC Press*, pages 105–175, 2001. [4](#)
- [21] C. Wilson. Fingerprint Vendor Technology Evaluation 2003: Summary of Results and Analysis Report, NISTIR 7123. http://fpvte.nist.gov/report/ir_7123_analysis.pdf, 2004. [4](#)
- [22] A.K. Jain and Jianjiang Feng. Latent fingerprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):88–100, 2011. [5](#), [19](#), [36](#), [41](#), [53](#), [55](#)

- [23] FBI IAFIS - CJIS Division, Latent Hit of the Year Award Videos. http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis_latent_hit_of_the_year, . 6
- [24] FBI IAFIS - CJIS Division, 2012 Latent Hit of the Year. <http://www.fbi.gov/about-us/cjis/cjis-link/july-2012/2012-latent-hit-of-the-year>, . 6
- [25] FBI IAFIS - CJIS Division, 2010 Latent Hit of the Year. http://www.fbi.gov/news/stories/2010/october/latent_102510/latent_102510, . 6
- [26] The Innocence Project. <http://www.innocenceproject.org/>. 6
- [27] S. .A. Cole. More than zero: Accounting for error in latent fingerprint identification. *Journal of Criminal Law & Criminology*, 95:985, 2004. 6
- [28] Case report - Shirley Mckie fingerprint scandal. <http://www.shirleymckie.com/reports.htm>. 6
- [29] A Review of the FBI's Handling of the Brandon Mayfield Case. *Unclassified executive summary*, <http://www.justice.gov/oig/special/s0601/exec.pdf>. 7
- [30] Max M Houck. Csi: reality. *Scientific American*, 295(1):84–89, 2006. 8
- [31] Itiel E Dror and Jennifer L Mnookin. The use of technology in human expert domains: challenges and risks arising from the use of automated fingerprint identification systems in forensic science. *Law, Probability and Risk*, 9(1):47–67, 2010. 8
- [32] Tomi Ahonen. Phone book 2012: Statistics and facts on the mobile phone industry, 2012. 10, 133
- [33] John T Apostolos, William Mouyos, Judy Feng, and Dwayne T Jeffrey. Fingerprint based smartphone user verification, 2014. US Patent 20,140,310,804. 10, 133, 134
- [34] Kiran B Raja, Ramachandra Raghavendra, Vinay Krishna Vemuri, and Christoph Busch. Smartphone based visible iris recognition using deep sparse filtering. *Pattern Recognition Letters*, 57:33–42, 2015. 10, 133

- [35] David J Robertson, Robin SS Kramer, and A Mike Burton. Face averages enhance user recognition for smartphone security. *PloS one*, 10(3):e0119460, 2015. [10](#), [133](#)
- [36] Jaroslav Sedenka, Sathya Govindarajan, Paolo Gasti, and Kiran S Balagani. Secure out-sourced biometric authentication with performance evaluation on smartphones. *IEEE Transactions on Information Forensics and Security*, 10(2):384–396, 2015. [10](#), [133](#)
- [37] Chao Shen, Yong Zhang, Xiaohong Guan, and R. A. Maxion. Performance analysis of touch-interaction behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, 11(3):498–513, 2016. [10](#), [133](#)
- [38] Z. Sitová, J. Łedenka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, 2016. [10](#), [133](#)
- [39] Shri Karthikeyan, Sophia Feng, Ashwini Rao, and Norman Sadeh. Smartphone fingerprint authentication versus pins: A usability study. Carnegie Mellon University Technical Reports: CMU-CyLab-14-012, July, 2014. [10](#), [133](#)
- [40] Guoqiang Li, Bian Yang, M.A. Olsen, and C. Busch. Quality assessment for fingerprints collected by smartphone cameras. In *Computer Vision and Pattern Recognition Workshops*, pages 146–153, 2013. [12](#), [42](#), [44](#), [140](#)
- [41] D.R. Ashbaugh. Quantitative-Qualitative friction ridge analysis: An introduction to basic and advanced ridgeology. *CRC Press, FL*, pages 134–135, 1999. [18](#)
- [42] Kasey Wertheim, Glenn Langenburg, and Andre Moenssens. A report of latent print examiner accuracy during comparison training exercises. *Journal of forensic identification*, 56(1):55, 2006. [19](#), [20](#)
- [43] G Langenburg. A performance study of the ace-v process: A pilot study to measure the accuracy, precision, reproducibility, repeatability, and biasability of conclusions resulting from the ace-v process. *Journal of Forensic Identification*, 59(2):219–257, 2009. [20](#)

- [44] Itiel E Dror, Ailsa E Peron, Sara-Lynn Hind, and David Charlton. When emotions get the better of us: the effect of contextual top-down processing on matching fingerprints. *Applied Cognitive Psychology*, 19(6):799–809, 2005. [20](#)
- [45] Itiel E Dror, David Charlton, and Ailsa E Péron. Contextual information renders experts vulnerable to making erroneous identifications. *Forensic Science International*, 156(1):74–78, 2006. [20](#)
- [46] Itiel E Dror and David Charlton. Why experts make errors. *Journal of Forensic Identification*, 56(4):600, 2006. [20](#)
- [47] Lisa J Hall and Emma Player. Will the introduction of an emotional context affect fingerprint analysis and decision-making? *Forensic Science International*, 181(1):36–39, 2008. [20](#)
- [48] I.E. Dror, C. Champod, G. Langenburg, D. Charlton, H. Hunt, and R. Rosenthal. Cognitive issues in fingerprint analysis: Inter-and intra-expert consistency and the effect of a \hat{S} target \check{S} comparison. *Forensic Science International*, 208(1):10–17, 2011. [20](#), [22](#)
- [49] Peter AF Fraser-Mackenzie, Itiel E Dror, and Kasey Wertheim. Cognitive and contextual influences in determination of latent fingerprint suitability for identification judgments. *Science & Justice*, 53(2):144–153, 2013. [20](#)
- [50] Standard for the Documentation of Analysis, Comparison, Evaluation and Verification (ACE-V) (latent), v2.0. *SWGFAST*, http://www.swgfast.org/documents/documentation/121124_Standard-Documentation-ACE-V_2.0.pdf. [19](#)
- [51] CDEFFS: The Committee to Define an Extended Fingerprint Feature Set. <http://fingerprint.nist.gov/standard/cdeffs/index.html>, 2008. [19](#)
- [52] Melissa K. Taylor, Will Chapman, Austin Hicklin, George Kiebusinski, Peter Komarinski, John Mayer-Splain, and Rachel Wallner. Latent Interoperability Transmission Specification, LITS. http://www.nist.gov/manuscript-publication-search.cfm?pub_id=913170, 2013. [19](#)

- [53] Itiel Dror and Robert Rosenthal. Meta-analytically quantifying the reliability and biasability of forensic experts. *Journal of Forensic Sciences*, 53(4):900–903, 2008. 21
- [54] B.T. Ulery, R.A. Hicklin, J.A. Buscaglia, and M.A. Roberts. Accuracy and reliability of forensic latent fingerprint decisions. *Proceedings of the National Academy of Sciences*, 108(19):7733–7738, 2011. 21, 90
- [55] Bradford T. Ulery, R. Austin Hicklin, JoAnn Buscaglia, and Maria Antonia Roberts. Repeatability and reproducibility of decisions by latent fingerprint examiners. *PLoS ONE*, 7:e32800, 03 2012. 21
- [56] Fact sheet - Integrated Automated Fingerprint Identification System. http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis_facts, . 23
- [57] T. Kent. Visualization or development of crime scene fingerprints. In *Encyclopedia of Forensic Sciences*, pages 117 – 129. Academic Press, 2013. 23
- [58] S. Karimi-Ashtiani and C.-C.J. Kuo. A robust technique for latent fingerprint image segmentation and enhancement. In *International Conference on Image Processing*, pages 1492–1495, 2008. 25, 26
- [59] N.J. Short, M.S. Hsiao, A.L. Abbott, and E.A. Fox. Latent fingerprint segmentation using ridge template correlation. In *International Conference on Imaging for Crime Detection and Prevention*, pages 1–6, 2011. 25, 26, 41
- [60] K. Cao, E. Liu, and A Jain. Segmentation and enhancement of latent fingerprints: A coarse to fine ridge structure dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1847–1859, 2014. 25, 31, 35, 41
- [61] Xiao Yang, Jianjiang Feng, Jie Zhou, and Shutao Xia. Detection and segmentation of latent fingerprints. In *International Workshop on Information Forensics and Security*, pages 1–6, 2015. 25
- [62] Jude Ezeobiesi and Bir Bhanu. Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble. In *Computer Vision and Pattern Recognition Workshops*, pages 146–154, 2016. 25

- [63] Jiangyang Zhang, Rongjie Lai, and C.-C.J. Kuo. Latent fingerprint segmentation with adaptive total variation model. In *5th IAPR International Conference on Biometrics*, pages 189–195, 2012. [27](#), [41](#)
- [64] Jiangyang Zhang, Rongjie Lai, and C.-C. Jay Kuo. Latent fingerprint detection and segmentation with a directional total variation model. In *19th IEEE International Conference on Image Processing*, pages 1145–1148, 2012. [27](#), [41](#)
- [65] A. A. Paulino, J. Feng, and A. K. Jain. Latent fingerprint matching using descriptor-based hough transform. *IEEE Transactions on Information Forensics and Security*, 8(1):31–45, 2013. [27](#), [35](#), [36](#), [41](#), [121](#)
- [66] Qijun Zhao and A.K. Jain. Model based separation of overlapping latent fingerprints. *IEEE Transactions on Information Forensics and Security*, 7(3):904–918, 2012. [27](#)
- [67] Jianjiang Feng, Yuan Shi, and Jie Zhou. Robust and efficient algorithms for separating latent overlapped fingerprints. *IEEE Transactions on Information Forensics and Security*, 7(5):1498–1510, 2012. [28](#)
- [68] M. Schott, R. Merkel, and J. Dittmann. Sequence detection of overlapping latent fingerprints using a short-term aging feature. In *IEEE International Workshop on Information Forensics and Security*, pages 85–90, 2012. [28](#)
- [69] Austin Hicklin. Latent quality survey. *Noblis Inc.*, 2007. [29](#)
- [70] R. Austin Hicklin, JoAnn Buscaglia, Maria Antonia Roberts, Stephen B. Meagher, William Fellner, Mark J. Burge, Matthew Monaco, David Vera, Larry R. Pantzer, Calvin C. Yeung, and Ted N. Unnikumaran. Latent fingerprint quality: A survey of examiners. *Journal of Forensic Identification*, 61(4):385–418, 2011. [30](#), [41](#)
- [71] Martin Aastrup Olsen, Haiyun Xu, and Christoph Busch. Gabor filters as candidate quality measure for NFIQ 2.0. In *International Conference on Biometrics*, pages 158–163, 2012. [30](#), [41](#), [84](#)

- [72] S. Yoon, E. Liu, and A. K. Jain. On latent fingerprint image quality. In *International Conference on Pattern Recognition Workshops*, Nov. 2012. 30, 41
- [73] S. Crihalmeanu, A. Ross, S. Schuckers, and L. Hornak. A protocol for multibiometric data acquisition, a protocol for multibiometric data acquisition, storage and dissemination. *WVU Technical report*, 2007. 30, 47
- [74] Soweon Yoon, Jianjiang Feng, and Anil K. Jain. On latent fingerprint enhancement. *SPIE Biometric Technology for Human Identification*, 2010. 30, 41
- [75] Soweon Yoon, Jianjiang Feng, and Anil K. Jain. Latent fingerprint enhancement via robust orientation field estimation. In *International Joint Conference on Biometrics*, pages 1 –8, 2011. 30, 41
- [76] J. Feng, J. Zhou, and A. Jain. Orientation field estimation for latent fingerprint enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):925–940, 2013. 30, 41
- [77] Kai Cao, Tarang Chugh, Jiayu Zhou, Elham Tabassi, and Anil K Jain. Automatic latent value determination. 31
- [78] R. Austin Hicklin, JoAnn Buscaglia, and Maria Antonia Roberts. Assessing the clarity of friction ridge impressions. *Forensic Science International*, 226(1Ü3):106 – 117, 2013. ISSN 0379-0738. doi: 10.1016/j.forsciint.2012.12.015. URL <http://www.sciencedirect.com/science/article/pii/S037907381200552X>. 31
- [79] Anush Sankaran, Mayank Vatsa, and Richa Singh. Automated clarity and quality assessment for latent fingerprints. In *International Conference of Biometrics: Theory, Applications and Systems*, pages 1–6. IEEE, 2013. 31
- [80] Samarth Bharadwaj, Mayank Vatsa, and Richa Singh. Biometric quality: a review of fingerprint, iris, and face. *EURASIP Journal on Image and Video Processing*, 2014(1):1–28, 2014. 32

- [81] E. R. Henry. Classification and uses of fingerprints. *Routledge & Sons, London*, pages 54–58, 1900. [33](#)
- [82] A. K. Jain, Y. Chen, and M. Demirkus. Pores and ridges: High resolution fingerprint matching using level 3 features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):15–27, 2007. [33](#)
- [83] Mayank Vatsa, Richa Singh, and Afzel Noore. Unification of evidence-theoretic fusion algorithms: a case study in level-2 and level-3 fingerprint features. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(1):47–56, 2009.
- [84] Mayank Vatsa, Richa Singh, Afzel Noore, and Sanjay K. Singh. Combining pores and ridges with minutiae for improved fingerprint verification. *Signal Processing*, 89(12):2676 – 2685, 2009. [33](#)
- [85] Alessandra A Paulino, Jianjiang Feng, and Anil K Jain. Latent fingerprint matching using descriptor-based hough transform. *IEEE Transactions on Information Forensics and Security*, 8(1):31–45, 2013. [35](#), [56](#), [98](#), [121](#)
- [86] Jianjiang Feng and Anil K. Jain. Filtering large fingerprint database for latent matching. In *International Conference on Pattern Recognition*, pages 1–4, 2008. [33](#), [38](#)
- [87] Chang Su and S. Srihari. Latent fingerprint core point prediction based on Gaussian processes. In *20th International Conference on Pattern Recognition*, pages 1634 –1637, 2010. [34](#)
- [88] Masahiro Kawagoe and Akio Tojo. Fingerprint pattern classification. *Pattern Recognition*, 17(3):295 – 303, 1984. [34](#)
- [89] Maria Puertas, Daniel Ramos, Julian Fierrez, Javier Ortega-Garcia, and Nicomedes Exposito. Towards a better understanding of the performance of latent fingerprint recognition in realistic forensic conditions. In *International Conference on Pattern Recognition*, pages 1638–1641, 2010. ISBN 978-0-7695-4109-9. [35](#), [41](#)

- [90] Alessandra A. Paulino, Anil K. Jain, and Jianjiang Feng. Latent fingerprint matching: Fusion of manually marked and derived minutiae. In *23rd Conference on Graphics, Patterns and Images*, pages 63–70, 2010. ISBN 978-0-7695-4230-0. 35, 41
- [91] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia Cylinder-Code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, 2010. 36
- [92] M. Vatsa, R. Singh, A. Noore, and S.K. Singh. Quality induced fingerprint identification using extended feature set. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2008. 36
- [93] A.K. Jain, Jianjiang Feng, A. Nagar, and K. Nandakumar. On matching latent fingerprints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008. 38, 41
- [94] J. Feng, S. Yoon, and A. Jain. Latent fingerprint matching: Fusion of rolled and plain fingerprints. *Advances in Biometrics*, pages 695–704, 2009. 38
- [95] V.N. Dvornychenko. Evaluation of fusion methods for latent fingerprint matchers. In *International Conference on Biometrics*, 2012. 39
- [96] A. Mikaelyan and J. Bigun. Ground truth and evaluation for latent fingerprint matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 83–88, 2012. 39
- [97] Sharat Chikkerur, Er N. Cartwright, and Venu Govindaraju. K-plet and CBFS: A graph based fingerprint representation and matching algorithm. In *International Conference on Biometrics*, 2006. 39
- [98] Rainer Kärger, Mario Hildebrandt, and Jana Dittmann. An evaluation of biometric fingerprint matchers in a forensic context using latent impressions. In *Multimedia and Security*, pages 133–138, 2012. ISBN 978-1-4503-1417-6. 39
- [99] Robert Vazan. SourceAFIS, 2012. <http://sourceforge.net/projects/sourceafis>. 39

- [100] S. Patel. Fingerprint Verification System, 2012. Available: <http://fvs.sourceforge.net>. 39
- [101] Biometric SDK, 2012. <http://sourceforge.net/projects/biometricsdk>, . 39
- [102] INNOVATRICES s.r.o. IDKit PC SDK,. <http://www.innovatrics.com/products/fingerprint-identification-sdk>, . 39
- [103] E. Liu, S. Arora, K. Cao, and A. K. Jain. A feedback paradigm for latent fingerprint matching. In *International Conference on Biometrics*, 2013. 40
- [104] Mayank Vatsa, Richa Singh, Afzel Noore, and Keith Morris. Simultaneous latent fingerprint recognition. *Applied Soft Computing*, 11(7):4260 – 4266, 2011. 40, 55
- [105] A. Sankaran, M. Vatsa, and R. Singh. Hierarchical fusion for matching simultaneous latent fingerprint. In *International Conference on Biometrics: Theory, Applications and Systems*, 2012. 40, 47, 52, 55, 71, 89
- [106] S. Shapoori and N. Allinson. GA-Neural approach for latent finger print matching. In *International Conference on Intelligent Systems, Modelling and Simulation*, pages 49–52, 2011. 40
- [107] Anush Sankaran, Parul Pandey, Mayank Vatsa, and Rajdeep Singh. On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders. In *International Joint Conference on Biometrics*, pages 1–7, 2014. 41, 122, 127, 128
- [108] Chulhan Lee, Sanghoon Lee, Jaihie Kim, and Sung-Jae Kim. Preprocessing of a fingerprint image captured with a mobile camera. In *Advances in Biometrics, LNCS*, pages 348–355. 2005. 42, 44
- [109] Dongjae Lee, Kyoungtaek Choi, Heeseung Choi, and Jaihie Kim. Recognizable-image selection for fingerprint recognition with a mobile-device camera. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(1):233–243, 2008. 42, 44
- [110] Chris Stein, Claudia Nickel, and Christoph Busch. Fingerphoto recognition with smartphone cameras. In *International Conference of the Biometrics Special Interest Group*, pages 1–12, 2012. 42, 44

- [111] Mohammad Omar Derawi, Bian Yang, and Christoph Busch. Fingerprint recognition with embedded cameras on mobile phones. In *Security and Privacy in Mobile Information and Communication Systems*, pages 136–147. Springer, 2012. 44
- [112] Guoqiang Li, Bian Yang, R Raghavendra, and Christoph Busch. Testing mobile phone camera based fingerprint recognition under real-life scenarios. In *Norwegian Information Security Conference*, 2012. 42, 43, 44
- [113] Chris Stein, Vincent Bouatou, and Christoph Busch. Video-based fingerphoto recognition with anti-spoofing techniques with smartphone cameras. In *International Conference of the Biometrics Special Interest Group*, pages 1–12, 2013. 42, 44
- [114] K. Tiwari and P. Gupta. A touch-less fingerphoto recognition system for mobile hand-held devices. In *International Conference on Biometrics*, pages 151–156, 2015. 44
- [115] Shervin Minaee and Yao Wang. Fingerprint recognition using translation invariant scattering network. In *IEEE Signal Processing in Medicine and Biology Symposium*, pages 1–6, 2015. 42, 44, 146, 147, 148, 151
- [116] Biometrics Research Group Inc. <http://www.biometricupdate.com>. 45
- [117] United States Visitor and Immigrant Status Indicator Technology program (US-VISIT). <http://www.dhs.gov/obim>. 45
- [118] Aadhaar, Unique Identification Authority of India (UIDAI). Planning Commission, Government of India, <http://uidai.gov.in/>. 45
- [119] Integrated Automated Fingerprint Identification System. http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis, . 45
- [120] Dual resolution images from paired fingerprint cards. *NIST Special Database 30*, 2010. 47
- [121] Plain and rolled images from paired fingerprint cards. *NIST Special Database 29*, 2010. 47
- [122] Fingerprint minutiae from latent and matching tenprint images. *NIST Special Database 4*, 2010. 47, 89, 98

- [123] Nist Supplemental Fingerprint Card Data (SFCD - NIST special database 10. <http://www.nist.gov/srd/nistsd10.cfm>, 2010. 47
- [124] NIST 8-bit gray scale images of Fingerprint Image Groups(FIGS). *NIST Special Database 14*, 2010. 47, 124, 125
- [125] Prasanna Krishnasamy, Serge Belongie, and David Kriegman. Wet fingerprint recognition: Challenges and opportunities. In *International Joint Conference on Biometrics*, 2011. 47
- [126] J Galbally, J Fierrez, F Alonso-Fernandez, and M Martinez-Diaz. Evaluation of direct attacks to fingerprint verification systems. *Telecommunication Systems*, 47(3):243–254, 2011. 47
- [127] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. FVC2000: Fingerprint Verification Competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002. 47
- [128] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2002: second Fingerprint Verification Competition. In *International Conference on Pattern Recognition*, pages 811–814, 2002. 47
- [129] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim Wayman, and Anil Jain. FVC2004: Third Fingerprint Verification Competition. In *Biometric Authentication*, pages 1–5, 2004. 47
- [130] D.T. Toledano J. Fierrez-Aguilar, J. Ortega-Garcia and J. Gonzalez-Rodriguez. BIOSEC baseline corpus: A multimodal biometric database. *Pattern Recognition*. 47
- [131] Casia-fingerprintv5. *Chinese Academy of Sciences’ Institute of Automation (CASIA) Fingerprint Image Database Version 5.0*. URL <http://biometrics.idealtest.org/>. 47, 124, 125
- [132] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro. MCYT

- baseline corpus: a bimodal biometric database. *IEE Proceedings on Vision, Image and Signal Processing*. 47, 124, 125
- [133] NIST digital video of live-scan fingerprint database. *NIST Special Database 24*, 2010. 47
- [134] Ajay Kumar and Yingbo Zhou. Contactless fingerprint identification using level zero features. In *Computer Vision and Pattern Recognition Workshops*, pages 114–119, 2011. 47
- [135] The Hong Kong Polytechnic University (PolyU) High-Resolution-Fingerprint (HRF) database. URL <http://www.comp.polyu.edu.hk/~biometrics/HRF/HRF.htm>. 47
- [136] Jianjiang Feng, Yuan Shi, and Jie Zhou. Robust and efficient algorithms for separating latent overlapped fingerprints. *IEEE Transactions on Information Forensics and Security*, 7(5):1498–1510, 2012. ISSN 1556-6013. doi: 10.1109/TIFS.2012.2204254. 47
- [137] A. Sankaran, T. I. Dhamecha, M. Vatsa, and R. Singh. On matching latent to latent fingerprints. In *International Joint Conference on Biometrics*, 2010. 47, 55, 71, 89, 121
- [138] F. Alonso-Fernandez, R.N.J. Veldhuis, A.M. Bazen, J. Fierrez-Aguilar, and J. Ortega-Garcia. Sensor interoperability and fusion in fingerprint verification: A case study using minutiae-and ridge-based matchers. In *International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006. 46
- [139] Strengthening forensic science in the United States: a path forward, National Research Council. 2009. 46
- [140] A. Sankaran, M. Vatsa, and R. Singh. Latent fingerprint matching: A survey. *IEEE Access*, 2:982–1004, 2014. 46, 121
- [141] VeriFinger by NeuroTechnology. www.neurotechnology.com/verifinger.html. 56, 58
- [142] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, 2010. 56, 98, 121, 146, 148

- [143] Matteo Ferrara, Davide Maltoni, and Raffaele Cappelli. Noninvertible minutia cylinder-code representation. *IEEE Transactions on Information Forensics and Security*, 7(6):1727–1737, 2012. [56](#), [98](#)
- [144] Elham Tabassi, CL Wilson, and C Watson. NIST Fingerprint Image Quality (NFIQ). *NIST Report NISTIR7151*, 2004. [56](#)
- [145] Anush Sankaran, Aakarsh Malhotra, Apoorva Mittal, Mayank Vatsa, and Richa Singh. On smartphone camera based fingerphoto authentication. In *International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–7, 2015. [64](#), [134](#), [135](#), [143](#)
- [146] Haiying Guan, AM. Dienstfrey, M.F. Theofanos, and B. Stanton. *A Measurement Metric for Forensic Latent Fingerprint Preprocessing*. 2014. [69](#), [87](#), [88](#)
- [147] Asker M Bazen and Sabih H Gerez. Segmentation of fingerprint images. In *Workshop on Circuits Systems and Signal Processing*, pages 276–280, 2001. [70](#), [72](#)
- [148] NIST. NFIQ 2.0, Quality features definition. http://biometrics.nist.gov/cs_links/quality/NFIQ_2/NFIQ-2_Quality_Feature_Defin-Ver05.pdf. [70](#), [72](#)
- [149] Yongfang Zhu, S. C. Dass, and A. K. Jain. Statistical models for assessing the individuality of fingerprints. *IEEE Transactions on Information Forensics and Security*, 2(3), September 2007. [70](#), [72](#), [79](#)
- [150] A. Borji. What is a salient object? a dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing*, 24(2):742–756, 2015. [73](#)
- [151] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems*, pages 545–552, 2006. [73](#), [74](#)
- [152] Xinjian Chen, Jie Tian, Jiangang Cheng, and Xin Yang. Segmentation of fingerprint images using linear classifier. *EURASIP Journal on Advances in Signal Processing*, 2004(4):480–494, 2004. [75](#)

- [153] C. Gottschlich. Curved-region-based ridge frequency estimation and curved gabor filters for fingerprint image enhancement. *IEEE Transactions on Image Processing*, 21(4):2220–2227, 2012. [76](#)
- [154] Sharat Chikkerur, Venu Govindaraju, and Alexander N. Cartwright. Fingerprint image enhancement using STFT analysis. *Pattern Recognition and Image Analysis*, 3687:20–29, 2005. [77](#), [81](#)
- [155] Hartwig Fronthaler, Klaus Kollreider, and Josef Bigun. Local features for enhancement and minutiae extraction in fingerprints. *IEEE Transactions on Image Processing*, 17(3):354–363, 2008. [79](#)
- [156] Josef Ström Bartunek, Mikael Nilsson, Benny Sallberg, and Ingvar Claesson. Adaptive fingerprint image enhancement with emphasis on preprocessing of data. *IEEE Transactions on Image Processing*, 22(2):644–656, 2013. [79](#)
- [157] Yong Jian Chin, Thian Song Ong, Andrew Beng Jin Teoh, and KOM Goh. Integrated biometrics template protection technique based on fingerprint and palmprint feature-level fusion. *Information Fusion*, 18:161–174, 2014. [81](#)
- [158] Mohamed Ben Salah, Amar Mitiche, and Ismail Ben Ayed. Multiregion image segmentation by parametric kernel graph cuts. *IEEE Transactions on Image Processing*, 20(2):545–557, 2011. [82](#)
- [159] S. Yoon, E. Liu, and A. K. Jain. On latent fingerprint image quality. In *International Workshop on Computational Forensics*, pages 1–6, 2012. [82](#)
- [160] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. Diversity in search strategies for ensemble feature selection. *Information fusion*, 6(1):83–98, 2005. [83](#)
- [161] Alexey Tsymbal, Seppo Puuronen, and David W Patterson. Ensemble feature selection with the simple bayesian classification. *Information Fusion*, 4(2):87–100, 2003. [83](#)
- [162] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992. [83](#)

- [163] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relief and rrelief. *Machine learning*, 53(1-2):23–69, 2003. 84
- [164] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. 84
- [165] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 84
- [166] Lior Rokach. Decision forest: Twenty years of research. *Information Fusion*, 27:111–125, 2016. 84
- [167] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991. 87
- [168] Haiying Guan, AM. Dienstfrey, and M.F. Theofanos. A new metric for latent fingerprint image preprocessing. In *Computer Vision and Pattern Recognition Workshops*, pages 84–91, 2013. 87
- [169] John M. Libert, J. Grantham, and S. Orandi. *A 1D Spectral Image Validation/Verification Metric for Fingerprints*. Technical report, National Institute of Standards and Technology (NIST), 2009. 87
- [170] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 91
- [171] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13(1):27–66, 2012. 95
- [172] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. 104
- [173] Zhonglong Zheng and Jie Yang. Supervised locality pursuit embedding for pattern classification. *Image and Vision Computing*, 24(8):819–826, 2006. 105

- [174] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, pages 2283–2291, 2013. [105](#), [110](#)
- [175] A. Sankaran, M. Vatsa, and R. Singh. Multisensor optical and latent fingerprint database. *IEEE Access*, 3:653–665, 2015. [105](#), [125](#), [126](#)
- [176] Nan-Nan Ji, Jiang-She Zhang, and Chun-Xia Zhang. A sparse-response deep belief network based on rate distortion theory. *Pattern Recognition*, 47(9):3179–3191, 2014. [106](#)
- [177] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. 1998. [106](#)
- [178] Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight-sharing. *Neural Computation*, 4(4):473–493, 1992. [106](#), [107](#)
- [179] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. [106](#), [107](#), [161](#)
- [180] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, pages 807–814, 2010. [107](#)
- [181] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International Conference on Machine Learning*, 2013. [107](#)
- [182] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. [107](#), [119](#)
- [183] Heng Luo, Ruimin Shen, Changyong Niu, and Carsten Ullrich. Sparse group restricted boltzmann machines. In *AAAI*, 2011. [107](#), [108](#)
- [184] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013. [107](#), [166](#), [167](#), [169](#)

- [185] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008. [108](#)
- [186] Ruoxin Sang, Peiquan Jin, and Shouhong Wan. Discriminative feature learning for action recognition using a stacked denoising autoencoder. In *Intelligent Data analysis and its Applications, Volume I*, pages 521–531. Springer, 2014. [109](#)
- [187] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang. Single sample face recognition via learning deep supervised autoencoders. *IEEE Transactions on Information Forensics and Security*, 10(10):2108–2118, 2015. [109](#)
- [188] Angshul Majumdar and Rabab Kreidieh Ward. Synthesis and analysis prior algorithms for joint-sparse recovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3421–3424, 2012. [109](#)
- [189] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint l_2 , l_1 -norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010. [109](#), [116](#)
- [190] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. l_2 , l_1 -norm regularized discriminative feature selection for unsupervised learning. In *International Joint Conference on Artificial Intelligence*, volume 22, page 1589, 2011. [109](#)
- [191] Thomas Blumensath. Compressed sensing with nonlinear observations and related nonlinear optimization problems. *IEEE Transactions on Information Theory*, 59(6):3466–3474, 2013. [112](#)
- [192] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. [113](#)
- [193] Yan Liu, Shusen Zhou, and Qingcai Chen. Discriminative deep belief networks for visual data classification. *Pattern Recognition*, 44(10):2287–2296, 2011. [114](#)

- [194] Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014. [114](#)
- [195] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *International Conference on Machine Learning*, pages 536–543, 2008. [114](#)
- [196] Hugo Larochelle, Michael Mandel, Razvan Pascanu, and Yoshua Bengio. Learning algorithms for the classification restricted boltzmann machine. *Journal of Machine Learning Research*, 13(1):643–669, 2012. [115](#)
- [197] Shane F Cotter, Bhaskar D Rao, Kjersti Engan, and Kenneth Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53(7):2477–2488, 2005. [117](#)
- [198] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *AISTATS*, pages 448–455, 2009. [118](#)
- [199] Ling Shao, Di Wu, and Xuelong Li. Learning deep and wide: A spectral method for learning deep networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12):2303–2308, 2014. [118](#)
- [200] Jianjiang Feng, Jie Zhou, and A.K. Jain. Orientation field estimation for latent fingerprint enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):925–940, 2013. [121](#)
- [201] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010. [122](#), [164](#), [165](#), [166](#)
- [202] Mark Davenport, Richard G Baraniuk, Clayton D Scott, et al. Controlling false alarms with support vector machines. In *International Conference on Acoustics, Speech and Signal Processing, 2006*, volume 5, 2006. [123](#)

- [203] Xiaofei Jia, Xin Yang, Yali Zang, Ning Zhang, and Jie Tian. A cross-device matching fingerprint database from multi-type sensors. In *International Conference on Pattern Recognition*, pages 3001–3004, 2012. [124](#), [125](#)
- [204] Kuang-Yu Chang, Cheng-Fu Lin, Chu-Song Chen, and Yi-Ping Hung. Applying scattering operators for face recognition: A comparative study. In *International Conference on Pattern Recognition*, pages 2985–2988, 2012. [134](#)
- [205] Shervin Minaee, AmirAli Abdolrashidi, and Yao Wang. Iris recognition using scattering transform and textural features. In *IEEE Signal Processing and Signal Processing Education Workshop*, pages 37–42, 2015. [134](#)
- [206] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Computer Vision and Pattern Recognition*, pages 1233–1240, 2013. [134](#), [140](#)
- [207] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015. [136](#)
- [208] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):1–20, 2013. [136](#), [138](#)
- [209] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11 (285-296):23–27, 1975. [139](#)
- [210] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. [140](#)
- [211] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015. [140](#)
- [212] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013. [140](#)

- [213] Joan Bruna and Stéphane Mallat. Classification with scattering operators. In *IEEE CVPR*, pages 1561–1566, 2011. [140](#)
- [214] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. [142](#)
- [215] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. [142](#)
- [216] AW-K Kong and David Zhang. Competitive coding scheme for palmprint verification. In *International Conference on Pattern Recognition*, volume 1, pages 520–523, 2004. [146](#), [148](#)
- [217] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. [148](#)
- [218] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 1(4):7, 2009. [161](#), [162](#)
- [219] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. [161](#)
- [220] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. [161](#)
- [221] Minmin Chen, Kilian Q Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In *International Conference on Machine Learning*, pages 1476–1484, 2014. [164](#), [166](#)
- [222] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning*, pages 833–840, 2011. [164](#), [166](#)

- [223] Kevin Swersky, Marc’Aurelio Ranzato, David Buchman, Benjamin Marlin, and Nando Freitas. On autoencoders and score matching for energy based models. In *International Conference on Machine Learning*, pages 1201–1208, 2011. [164](#), [166](#)
- [224] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996. [165](#)
- [225] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. *Deeply-supervised nets*. arXiv preprint arXiv:1409.5185, 2014. [166](#), [169](#)