Investigating Combinations of Feature Extraction and Classification for Improved Image-Based Multimodal Biometric Systems at the Feature Level

Submitted in fulfilment

of the requirements for the degree of

Doctor of Philosophy

of Rhodes University

Dane Brown

Grahamstown, South Africa

March 19, 2018

Abstract

Multimodal biometrics has become a popular means of overcoming the limitations of unimodal biometric systems. However, the rich information particular to the feature level is of a complex nature and leveraging its potential without overfitting a classifier is not well studied. This research investigates feature-classifier combinations on the fingerprint, face, palmprint, and iris modalities to effectively fuse their feature vectors for a complementary result. The effects of different feature-classifier combinations are thus isolated to identify novel or improved algorithms.

A new face segmentation algorithm is shown to increase consistency in nominal and extreme scenarios. Moreover, two novel feature extraction techniques demonstrate better adaptation to dynamic lighting conditions, while reducing feature dimensionality to the benefit of classifiers. A comprehensive set of unimodal experiments are carried out to evaluate both verification and identification performance on a variety of datasets using four classifiers, namely Eigen, Fisher, Local Binary Pattern Histogram and linear Support Vector Machine on various feature extraction methods. The recognition performance of the proposed algorithms are shown to outperform the vast majority of related studies, when using the same dataset under the same test conditions. In the unimodal comparisons presented, the proposed approaches outperform existing systems even when given a handicap such as fewer training samples or data with a greater number of classes.

A separate comprehensive set of experiments on feature fusion show that combining modality data provides a substantial increase in accuracy, with only a few exceptions that occur when differences in the image data quality of two modalities are substantial. However, when two poor quality datasets are fused, noticeable gains in recognition performance are realized when using the novel feature extraction approach.

Finally, feature-fusion guidelines are proposed to provide the necessary insight to leverage the rich information effectively when fusing multiple biometric modalities at the feature level. These guidelines serve as the foundation to better understand and construct biometric systems that are effective in a variety of applications.

ACM Computing Classification System Classification

Thesis classification under the ACM Computing Classification System[1] (2012 version, valid through 2018):

Security and privacy – Biometrics

Computing methodologies – Image processing

Computing methodologies – Supervised learning by classification

General-Terms: multimodal biometrics, feature-level fusion, guidelines, face, fingerprint, palmprint, iris

[1] https://www.acm.org/publications/class-2012

Acknowledgements

First and foremost, I thank my Lord and Saviour, Jesus Christ, for giving me the strength, wisdom and perseverance throughout life. I am also indebted to my loving parents for motivating me to excel at everything and not to easily get discouraged by the negative sources of life.

I would especially like to thank my supervisor Professor Karen Bradshaw for her outstanding guidance and belief in my abilities. Moreover, thank you for being so understanding and patient with me during dire moments. I thank James Connan not just as a co-supervisor, but as an esteemed colleague that encouraged me to pursue postgraduate studies in Computer Science. The invaluable knowledge that I acquired from Reg Dodds, James Connan and Mehrdad Ghaziasgar in previous years provided the foundation of knowledge and problem solving abilities to complete this thesis.

I would like to thank the Rhodes University Computer Science department for the opportunity to study here and in particular Barry Irwin, Karen Bradshaw, James Connan and Caro Watkins for "rolling out the royal carpet", welcoming me. I cherish all the awesome moments shared with my lab mates from the "Red" room, especially Ibraheem Frieslaar that journeyed with me throughout this PhD, including previous years of study, as not just my colleague but best friend. Ibraheem Frieslaar, as my rival, you were integral in my continued improvement in several aspects of life.

The authors of the publicly available datasets used in this dissertation are acknowledged: PolyU (Zhang *et al.*, 2003), ORL (Samaria and Harter, 1994), SDUMLA-HMT (Yin *et al.*, 2011), IITD (Kumar, 2008), MCYT (Ortega-Garcia *et al.*, 2003), PUT (Kasinski *et al.*, 2008) and CASIA (Sun *et al.*, 2005).

I would like to thank the CSIR, Information Security Department, for their generous financial support. Additional financial support from Telkom SA, Coriant, Easttel and Bright Ideas 39 is also acknowledged through the Rhodes CoE. The author acknowledges that opinions, findings and conclusions or recommendations expressed here are those of

the author and that none of the above-mentioned sponsors accepts liability whatsoever in this regard.

Contents

List of Figures

List of Tables

Glossary

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AHE	Adaptive Histogram Equalization
CLAHE	Contrast Limited Adaptive Histogram Equalization
DFT	Discrete Fourier Transform
DIR	Detection and Identification Rate
DoG	Difference of Gaussians
DPI	Dots per Inch
EER	Equal Error Rate
ELBP	Extended Local Binary Pattern
ELBPV	Enhanced Local Binary Pattern Variance
FAR	False Acceptance Rate
FRR	False Rejection Rate
FNIR	False-Negative Identification-Error Rate
FPIR	False-Positive Identification-Error Rate
GAR	Genuine Acceptance Rate
HoG	Histogram of Oriented Gradients
HE	Histogram Equalization
HT	Hough Transform
IDO	Integro-Differential Operator
IR	Identification Rate
KFA	Kernel Fisher Analysis
LBP	Local Binary Pattern
LBPV	Local Binary Pattern Variance
LDA	Linear Discriminant Analysis
LBPH	Local Binary Pattern Histogram
LoG	Laplacian of Gaussian
MEC	Maximum Effective Circle

MIC	Maximum Inscribed Circle
NMS	Non-Maxima Suppression
OLOF	Orthogonal Line Ordinal Features
OF	Orientation Field
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SVM	Support Vector Machine
SIFT	Scale Invariant Feature Transform
VRA	Varied Reference Angle

Chapter 1

Introduction

1.1 Context of Research

Since very early in civilization, human recognition has been a simple but restricted task. Faces can be visually identified to ascertain familiar and unfamiliar individuals in a fairly unconscious manner (Kindt, 2013). However, the task has become increasingly challenging with the rise in populations, especially in terms of cross-border security and crime. Biometrics research has become more popular, however, it is reliant on using traits that are unique and easily distinguishable.

A biometric system (or biometrics for short) is defined as the measurement and analysis of unique biological and behavioural traits to establish the identity of a human being for a particular purpose, often related to access control or law enforcement (Jain *et al.*, 2000b, Kindt, 2013). Biometrics is thus used by both government and civilian applications offering greater convenience and several advantages over traditional authentication measures as they cannot be lost or forgotten. However, with technological advances resulting in the widespread use of electronic sensors and networks, security risks have been introduced into biometric systems by forgers (Jain *et al.*, 2000b, Kindt, 2013). Furthermore, real-world conditions such as user error and dirty sensors often result in degradation of the biometric data being modelled. This has prompted developers to bias these systems

towards either increased security at the cost of stricter or invasive requirements for users, or more convenient at the cost of reduced security.

In a bid to counteract these real-world problems, it has been shown that using more than one source of biometric information improves security, recognition accuracy and versatility of biometric systems (Iloanusi, 2014, Rattani *et al.*, 2011). Multimodal biometrics thus has the potential to solve these problems in well-planned applications.

1.2 Motivation for this Research

Existing multimodal biometric recognition studies typically make use of score-level fusion, which combines the little information remaining at a late processing step in the biometric recognition process. The less widely used alternative, known as feature-level fusion, has recently attracted more interest as it utilizes multiple sources of rich biometric information at an earlier processing step. Fusion at the feature level has been shown to outperform that at the score level in recent studies (Rattani *et al.*, 2011, Thepade *et al.*, 2015), but research is lacking with regard to substantiation of its applicability under various conditions. On the contrary, comprehensive reviews have been conducted at the score level with the results often used to construct optimal fusion schemes under a variety of conditions (Raghavendra *et al.*, 2011). The reviews have established guidelines for score-level fusion enabling the systematic implementation of multimodal biometric systems for future research and applications.

While the rich discriminatory information available at the feature level has greater potential, effectively utilizing this information such that systematic approaches can be identified is a complex problem that exemplifies the saying: "One size does not fit all." This is further exacerbated by the additional factors that influence recognition accuracy including the problem known as the "curse of dimensionality" and the vast number of feature-classifier combinations coupled with the nature of the biometric data.

As it is yet to be established which image processing and machine learning practices are applicable for feature-level fusion systems, there is not a clear starting point for creating

multimodal biometrics system using fusion at the feature level. Furthermore, various algorithms are relevant for the image segmentation procedure in order to capture the unique features of different image-based modalities. Evaluating the algorithms used for each modality requires comprehensive experimentation. However, since different image-based biometric modalities are independent yet complementary (Raghavendra *et al.*, 2011, Wang *et al.*, 2013), several feature-classifier combinations can be evaluated by applying the same feature extraction and transformation method on different modalities, enabling compatibility of the fused components to be classified (Raghavendra *et al.*, 2011, Wang *et al.*, 2013) (Ross *et al.*, 2006, p. 65–70).

1.3 Research Statement

This research aims to investigate several combinations of a limited set of feature extraction techniques and classifiers under various conditions to determine the effect on the recognition accuracy of image-based modalities, for unimodal and multimodal biometrics fused at the feature level.

We hypothesize that improved accuracy can be achieved using multimodal biometric fusion at the feature level, but the same combination of techniques may not apply optimally to all modalities.

1.4 Research Objectives

To prove our hypothesis the research focussed on achieving the following objectives:

1. Identify combinations of feature extraction and classifier algorithms that can be used to construct a number of effective unimodal biometric systems based on those found in the literature.

2. Develop novel or improved algorithms for implementing biometric systems for individual modalities that address limitations found in related systems.

3. Implement a number of unimodal biometric systems based on the combinations of existing or new algorithms for use with each of the four image-based modalities (fingerprint, face, palmprint, and iris) separately.

4. Identify strengths and weaknesses of each implemented unimodal biometric system as well as trends observed across experiments on the four modalities.

5. Implement bimodal biometric systems by applying the optimal algorithms, based on the observed trends, which are applicable to pairs of modalities fused at the feature level.

6. Identify strengths and weaknesses of the bimodal biometric systems as well as trends observed across the multimodal experiments.

7. Identify foundational guidelines based on the results of unimodal and multimodal experiments that can be used for future biometric systems that apply fusion at the feature level.

1.5 Approach

A fixed preprocessing approach is used on the four modalities based on popular biometrics in the literature. This forms the basis of four biometric systems where several feature extraction and classifier combinations algorithms are evaluated empirically to validate those that are optimal. As feature extraction and classifiers can both have different optimal parameters depending on various conditions, parameters are tuned systematically at the start of each experiment. Any subsequent tuning with significant impact on results is explicitly noted together with the reason for the change. The methodology per modality is also validated visually as appropriate to image-based biometrics.

Evaluation metrics are used for the unimodal experiments on three datasets per modality, which are all unseen to ensure a fair empirical process. These datasets are also acquired using different sensors than the validation dataset to show compatibility across sensors. Furthermore, the unimodal systems are tested on all these datasets against related systems in the literature to determine the effectiveness of the proposed methodology. Although evaluating the identification performance is the focus of this research, verification performance is included where necessary for fair comparison with related studies.

Bimodal experiments are conducted on two of the available datasets per modality, where two fusion methods known as feature averaging and feature concatenation are compared using the same metrics as the unimodal experiments. Another bimodal experiment is carried out on the omitted datasets by ranking the top 10 worst individuals (classes) per fusion method and visually inspecting the image samples to ascertain whether there are observable class-specific trends. Weighted fusion is performed on the top 10 worst ranked individuals in an attempt to further improve recognition accuracy and increase robustness to unbalanced components within bimodal datasets.

1.6 Assumptions

The datasets used in the experiments were chosen based on the following assumptions:

- The image quality ranges from ideal to poor so as to depict real-world conditions for biometric applications.
- Acquisition of images for all four of the modalities chosen could be done in a non-invasive procedure to the user. However, the datasets may include extreme conditions such as occlusion of faces, partial fingerprints, and so on.

1.7 Limitations

- Four image-based modalities (fingerprint, face, palmprint, and iris) are used for experimentation.
- Fusion is limited to bimodal combinations of the feature concatenation and feature averaging method.
- The weighted bimodal fusion experiments are conducted using precalculated weights on both fusion methods, while dynamic weighting is performed manually to demonstrate proof of concept.
- The same four classifiers per proposed biometric system are used on each modality and combination in unimodal and bimodal experiments, respectively.
- While rough indications of classifier speed are given as an extra guideline, computational complexity for training and test procedures per system is not a focus of this research.
- Due to different quality standards per biometric modality, image quality is subjectively evaluated during experimentation using visual inspection.

1.8 Thesis Outline

The remainder of this thesis is arranged as follows:

Chapter 2: *Biometric Systems*: This chapter introduces the various concepts pertaining to image-based biometrics.

Chapter 3: *Algorithms used in Biometric Systems*: The algorithms pertaining to the recognition of fingerprint, face, palmprint and iris biometric modalities are described including their strengths and weaknesses. The rest of the chapter reviews the literature and datasets relevant to the respective modalities.

Chapter 4: *Experimental Setup*: The methodology used to construct the four proposed individual biometric systems is first discussed. The conditions under which the set datasets were captured are described per modality. This chapter forms the basis of the optimization process on the algorithms used per modality in subsequent chapters.

Chapters 5 – 8: *[Modality] Recognition Algorithms and Results*: These chapters, respectively, present the proposed approach for each modality's recognition system followed by their methodology validation and results.

Chapter 9: *Bimodal Feature Fusion and Recognition Results*: This chapter describes the two fusion approaches, presents the results and discusses the differences between the unimodal and bimodal combinations.

Chapter 10: *Improving Feature-Fusion Accuracy through Weighted Modalities and Specific Guidelines*: This chapter discusses the significance of the experiments carried out, inspects the effect of image sample conditions on a per user basis, and how weighting strengthens multimodal biometric systems that fuse at the feature level under non-ideal conditions.

Chapter 11: *Conclusion*: This chapter concludes the thesis, highlights the contributions made towards the research and provides directions for future work.

Chapter 2

Biometric Systems

This chapter gives a broad overview of biometric systems, including four of the common image-based modalities and various ways of fusing such modalities.

2.1 Overview of Biometric Systems

Secure systems and locations have traditionally used passwords and smart/access cards as a form of access control (Jain *et al.*, 2011). However, as time has progressed security measures have been extensively studied, which showed that they are frequently bypassed or forged. The emergence of biometrics has to a certain degree addressed these risks to security by removing the need to "remember something", which has also meant that security can be applied with less inconvenience to the user. Some of the advantages of using biometrics according to Jain *et al.* (2011) are given below.

- Discourages fraud and enhances security.
- Detects multiple enrollments.
- Cannot be easily transferred, forgotten, lost, or copied.
- Virtually eliminates repudiation claims.

- Increases user convenience.

The underlying foundation of a biometric system is an automated process of recognizing an individual by acquiring and evaluating unique biological and behavioural patterns from individuals. Even though such systems have many advantages over smart cards and passwords, their widespread use has introduced new security risks posed by forgers (Bharadwaj *et al.*, 2015, Kindt, 2013). Their increasing susceptibility to forgery and degraded/ill-captured data has led to identity theft and system failure even with the introduction of more secure technologies. This is especially a concern in non-invasive biometric modalities, which are often external, such as the fingerprint, face, palmprint and iris. These types of biometrics are easier to acquire because they are external and are easily captured using image-based sensors, posing less inconvenience to the user. The pervasiveness of data captured from these sensors is a concern due to their transfer through sensor networks that can be intercepted and thus well-planned access control is vital in the real world.

Real-world security problems often change based on the application, but are especially evident in automatic security systems or those preferring low supervision. Challenges of secure biometric system design further include the dramatic change in samples (impressions) of the same biometric trait due to ageing, user cooperation or the environment (Jain *et al.*, 2011). This kind of change observed in a number of samples of an individual is known as intra-user or intra-class variation. On the other hand, when features extracted from the same biometric traits of different individuals have a significant overlap it is called high inter-class variation, for example, as in the case of the fingerprints or faces of identical twins or due to other cases of genetic similarities.

The majority of biometric systems are used in one of the following three focus areas (Kindt, 2013):

- Fighting crime – for example, using biometric evidence from a crime scene or ongoing pursuit such as latent fingerprints or face tracking over security camera feeds, respectively.

- Security and services – for example, authenticating an individual for banking services using a biometric sensor connected to a computer or access control to a room via a biometric activated door lock.

- Comfort – for example, unlocking smart phones or automatically adjusting the steering wheel and seat position of a car for a particular user.

While comfort is user-experience centric, the operational requirements for fighting crime and security services range from strict rejection of impostors – often resulting in rejection of genuine individuals – to accepting many candidates for later manual inspection (Jain *et al.*, 2011). These trade-offs are thus biased according to the application. A method of satisfying these requirements without (or with less) biasing has recently been discovered through the fusion of multiple sources of biometric information.

Multimodal biometrics is defined as the combination of two or more biometric modalities for the purpose of this dissertation. The advantages of multimodal biometrics are improved security, recognition accuracy and versatility of a biometric system (Ross *et al.*, 2006). However, the application to a large extent determines the effectiveness of fused biometric data. For instance, achieving universality or improving user experience may not be attainable by fusing biometrics and, in fact, may worsen the applicability. A selective approach, combining biometric data from easily acquirable modalities may be a solution, but many other factors such as fusion level, feature types and effectiveness need to be considered. The remainder of this chapter introduces concepts relevant to biometrics such as system types, modality fusion, and evaluation and performance indicators.

2.2 Types of Biometric Systems

Biometric systems can be broadly classified as verification (sometimes called authentication) or identification systems. Both systems can contain impostors, which are defined as unauthorized or non-class individuals.

The most recent definition for verification and identification was established by the joint international standards committee on biometrics, ISO/IEC JTC1/SC37 (Podio, 2017), in 2015 as:

1. Biometric verification – "Application that shows true or false a claim about the similarity of biometric reference(s) and recognition biometric sample(s) by making a comparison(s)." They also give examples of verification as: "Establishing the truth of any of the claims I am enrolled as subject X, I am enrolled in the database as an administrator, and I am not enrolled in the database may be considered verification." (Wayman, 2015, p. 266)

2. Closed-set identification – "Application that ranks the biometric references in the enrolment database in order of decreasing similarity against a recognition biometric sample. Note 1: Closed-set identification always returns a nonempty candidate list. Note 2: Closed-set identification is rarely used within practical systems, but is often used experimentally." (Wayman, 2015, p. 266)

3. Open-set identification – "Application that determines a possibly empty candidate list by collecting one or more biometric samples from a biometric capture subject and searching the enrolment database for similar biometric references." (Wayman, 2015, p. 266)

These definitions are placed into context with respect to metrics and clarified in Section 2.6.

2.3 Biometric Modalities

There are a vast number of human biometric traits, including fingervein, gait and voice. For the purposes of this research, however, four external biometric modalities are focussed on, namely fingerprint, face, palmprint and iris.

2.3.1 Fingerprint

The skin on the palm side of hands contains ridge and valley patterns unlike most of the skin on the rest of the human body. Since these patterns are unique and immutable at the tips of the fingers, they are useful for biometric authentication. However, since manual labour workers and a minority of people have skin problems at the surface, capturing their fingerprints often result in non-ideal samples. Currently, many law enforcement agencies worldwide use fingerprint biometrics as the primary means to verify or identify an individual (Jain *et al.*, 2011).

2.3.2 Face

Face recognition is an important research area as it is one of the more visible and user-friendly biometrics (Jain *et al.*, 2000a). In cases where the face is not used as an authentication biometric, it is still often bound to the primary identification device or system, such as an ID card or a criminal fingerprint database (Kindt, 2013). As a primary means of identification, the face is challenging in uncontrolled applications due to varied pose angles and occlusions. Furthermore, real-world conditions often result in degradation of the quality when acquiring face images from camera sensors.

When bound as a secondary form of identification, human face images help determine many other revealing attributes like gender, age, ethnicity and even the emotional state of a person (Kindt, 2013). Therefore, the face biometric is an important biometric identifier in law enforcement and human-computer interaction communities.

2.3.3 Palmprint

The palms of human hands also contain ridges and valleys, like the fingerprints, forming unique patterns, which are referred to as palmprints. Since the pattern containing area of the palm is much larger than that of the finger, palmprints are expected to be even

more distinctive than fingerprints (Zhang *et al.*, 2003, Kumar, 2008). However, palmprint scanners capture a larger area, and are thus bulkier and more expensive than fingerprint sensors. A higher resolution sensor (expensive) is thus required for a palmprint to capture features similar to fingerprint sensors.

Palms contain additional distinctive features known as principal lines and wrinkles that become defined after infancy, allowing palmprints to be captured with a lower resolution scanner, which is also likely to be cheaper (Ross *et al.*, 2006). Moreover, when using a high-resolution palmprint scanner, multiple features of the hand such as the geometry, ridge and valley features, principal lines and wrinkles can be combined to build a highly accurate biometric system (Ross *et al.*, 2006).

2.3.4 Iris

Iris recognition has been the focus of much research in recent years. Statistical experiments show that the iris has the most reliable and stable features of all external biometrics (Yin *et al.*, 2011). In a typical eye, the iris region includes the coloured tissue surrounding the pupil (black circular region), which is further surrounded by the sclera (white region). The muscles in the iris, namely, the dilator and sphincter, control the contraction and dilation, respectively, of the pupil region, which can be a hindrance to normalization when capturing the iris image.

2.4 Selection of an Ideal Biometric Modality

According to Wayman *et al.* (2005), many authors agree that the following characteristics must be taken into consideration when selecting a modality for use in a biometric system.

1. Universality: Every user must have this biometric trait, which implies complete population coverage for all users of the system. For example, manual labourers can damage their outer fingerprint and render it unrecognizable.

2. Collectability: Biometric data acquisition or availability should be simple without requiring extensive user cooperation or a large number of attempts to capture a single sample. The latter problem is referred to as "failure to enroll" based on an application specific limit.

3. Uniqueness: The biometric attribute should be distinctive for individuals across the target population. This means that at least a subset of its features should have large variations across individuals in the population. This is often linked to yielding a low number of false matches, which is sometimes referred to as "Type II errors" in biometric systems.

4. Permanence: The biometric attribute should remain constant over time, which includes age or alterations due to a disease not compromising robustness. This characteristic is often linked to yielding a low number of false non-matches, which is sometimes referred to as "Type I errors" in biometric systems.

5. Acceptability: The collection process should not intimidate users as they should be willing to submit their biometric samples. This characteristic can be measured by the number of system users.

6. Accessibility: This also includes ease of user interaction, which usually incorporates an electronic sensor that captures an external biometric. This is measured by the number of individuals that can be processed per time unit.

7. Circumvention: This refers to tamper resistance as the biometric should be difficult to spoof or manipulate.

8. Cost effectiveness: The hardware and software required to collect and identify the biometric should be reasonably priced and readily available to the average business.

The intuitive solution to determining the ideal biometric at a given time would be a weighted value incorporating all of the above qualities. Research however, has shown that desired qualities are highly dependent on the specifics of the application, including the target population, physiological and psychological states, and hardware/software used (Maio *et al.*, 2002, Mansfield *et al.*, 2001, Phillips *et al.*, 2000, Wayman *et al.*, 2005).

2.5 Fusion of Biometric Modalities

Early biometric systems typically used only a single modality. However, in an attempt to improve accuracy, biometric systems have recently attempted to combine two or more modalities. Such systems are referred to as multimodal systems. Multimodal biometrics can also be used to solve or mitigate problems of real-world scenarios such as achieving universality for entire population coverage in well-planned applications (Iloanusi, 2014). A practical example application is the use of joint user authentication where two or more users are required to provide their respective fingerprints, resulting in improved security of a shared interest.

Biometric data can be fused at different levels for multimodal biometric systems, namely, sensor level, feature level, score level, rank level and decision level. These are discussed in the subsections that follow.

2.5.1 Sensor-Level Fusion

Sensor-level fusion can directly combine the raw data of individual sensors (Wayman, 2015, Ross *et al.*, 2006). However, sensor-level fusion is most effective when sources of the same biometric trait are obtained from multiple different but compatible sensors, or when multiple instances of the same biometric trait are obtained from a single sensor. An example of the former is when multiple two-dimensional (2D) face images from different perspectives can be stitched together to form a three-dimensional (3D) model or a panoramic face image. An example of multiple instances from a single sensor is the mosaicking of several patches of the same fingerprint impression to form a complete fingerprint image. The challenge of sensor-level fusion is that multiple cues must be compatible and the start and end points of the raw data per sensor must be known in advance – calibrated camera systems – or reliably estimated during postprocessing.

2.5.2 Feature-Level Fusion

The combination of feature sets of two or more modalities is expected to produce significantly improved recognition performance and increased compatibility over sensor-level fusion (Wayman, 2015, Ross *et al.*, 2006). However, this is not always the case as varying degrees of improvement have been reported in the literature.

In this type of biometric fusion, feature sets per biometric trait are selected and/or extracted separately for subsequent fusion to produce a single multimodal biometric feature vector. This fused feature vector is used as the input to the matcher or classifier to make the final decision. This is expected to be the best type of fusion in terms of potential accuracy gain as the feature sets constitute the richest source of usable information.

2.5.3 Score-Level Fusion

This method of fusion involves the integration of modalities in the biometric process at a later stage, that is, after obtaining the match score (Wayman, 2015). This match score is used by a decision module usually involving various normalization techniques, and rules such as the max, min, sum, mean or median rule. This system is flexible in terms of compatibility such that it is fairly simple to implement. On the other hand, this widely used fusion level does not utilize the rich discriminatory information available at the feature level.

Comprehensive studies have been conducted at the score level across most biometric modalities. These studies often compare the best fusion methods, forming fusion guidelines that can be used in future applications of a similar nature (Raghavendra *et al.*, 2011, Ribaric and Fratric, 2006). Ross *et al.* (2006, p. 91–141) provide the numerous interactions of score normalization and rules by several authors including resulting accuracies and best practices in a dedicated chapter, unlike the other (lesser studied) levels. As this fusion level is well researched, the best suited rule for an application is decided based on specific biometric modalities and the environmental conditions under which they operate.

An example of the guidelines for score-level fusion is the use of min-max normalization to achieve a low false acceptance rate (FAR) and high genuine acceptance rate (GAR) when it is assumed that no low quality data is dealt with (Ross *et al.*, 2006, p. 128). On the other hand, median normalization achieves lower accuracies under those conditions, but is more suited to applications known to contain low quality data. Furthermore, the weighted sum rule is preferred for general fusion and the product rule is preferred for very high quality input data fusion.

2.5.4 Other Fusion Levels

The last two fusion levels are known as rank level and decision/abstract level (Wayman, 2015, Ross *et al.*, 2006). Rank level is relevant in an identification system where a value is assigned to the top matching identities, while decision level is commonly used when propriety rights prevent the use of the previous fusion levels. This often occurs with commercial off-the-shelf biometric matchers that provide access only to the final decision per modality (and per class), limiting the decision level to 'AND' and 'OR' rules, majority voting and weighted majority voting.

2.5.5 Feature Level vs. Score level Fusion

Initially, multimodal biometric systems adopted the score-level fusion approach as it was simple to implement through score normalization. Guidelines were subsequently established for this type of fusion (Rattani *et al.*, 2011). However, this widely used fusion method does not utilize the rich discriminatory information available at the feature level. Feature-level fusion integrates feature vectors corresponding to two or more biometric modalities. However, the challenge of feature-level fusion is that it is not only more complex than match score-level fusion, but also lacks guidelines, as it is a lesser studied problem. Ross *et al.* (2006, p. 65–70) show that limited information is available on the best practices of feature-level fusion as there are no comprehensive studies. At the time of writing, this gap had not been addressed. A commonality of the score and feature levels

is that they both support a wide range of biometric applications unlike the other fusion levels.

In this research, a comprehensive set of experiments on feature-level fusion are conducted, the results of which are later used as a basis for fair comparison against the score level and other well-studied fusion levels where relevant. The results are also used to inform future guidelines for feature-level fusion. Section 1.7 specifies that two specific feature-level fusion methods are investigated: feature concatenation and averaging.

At present, the guidelines used during feature-level fusion are limited to the following generalizations, which have been verified in a study (Brown and Bradshaw, 2017a). Feature set compatibility is imperative for the decision module, which typically consists of classification or distance matching. This is data dependent and requires further analysis through experiments. Uncorrelated feature sets should be used among different modalities and correlated feature sets among multiple samples of the same modality (Wang *et al.*, 2013). Furthermore, different image-based biometric modalities are often independent and complementary. Finally, the application of the same feature extraction algorithms on different modalities can yield a very efficient multimodal biometric system (Raghavendra *et al.*, 2011). Fusion is often applied after, as opposed to before, extraction, using linear or non-linear methods, as it can result in better reduction of feature correlation (Brown and Bradshaw, 2016a).

2.6 Biometric Performance Indicators

A similarity/confidence score related to distance measure is often used as a threshold that affects the accuracy of a biometric system. The threshold can be optimized for either high security or a high user acceptance rate. If the confidence values of a classifier are higher than the given threshold, the user is rejected as an impostor.

A threshold T (Jain and Li, 2011) is used to classify a test case to be either a correct (true positive) case or an incorrect (false positive) one. If the case is below a threshold T, it is classified as false and if it is above threshold T it is classified as true.

2.6.1 Verification

In verification systems, the input (probe) biometric data are labelled with a claimed identity (Jain and Li, 2011). The system compares the probe data strictly with similarly labelled templates or trained data in the gallery or database model. The aim of this type of system is to ensure that a user's claim of identity is correct. This involves a simple form of matching referred to as 1:1 matching, as a probe is compared against a single or small number of gallery entries or database samples for only the individual under scrutiny.

Verification accuracy is measured as follows:

- GAR – Percentage of class images that were correctly predicted out of the total class images.
- FAR – Percentage of impostor images predicted (incorrectly) as class images out of the total impostor images.
- False Rejection Rate (FRR) – Percentage of class images that were incorrectly predicted as non-class images, or simply $100 - \text{GAR}$.
- Equal Error Rate (EER) – Percentage at which FAR = FRR.

While a high GAR and a low EER are generally what a state-of-the-art biometric system aims to achieve, the measure of individual error rates such as FAR and FRR are important for biasing particular biometric applications. A receiver operating characteristic (ROC) curve measures the tradeoff between errors and is often used for plotting the verification or identification accuracy of a biometric system (Wayman, 2015, p. 1234–1236).

2.6.2 Identification

The identity of an individual's biometric data within the input image is determined by comparing the features extracted from the image with every template in the gallery or

database (Jain and Li, 2011, Tabassi *et al.*, 2014, Quinn *et al.*, 2016). This type of matching operation is also referred to as $1 : N$ matching, as N comparisons are performed during the matching operation, where N is the total number of images in the gallery/database for every person.

A closed-set identification system, also known as positive identification, refers to a biometric task in which an unidentified individual is known to be in the database of biometric characteristics and the system attempts to determine his/her identity (Jain and Li, 2011, Tabassi *et al.*, 2014, Quinn *et al.*, 2016). Since, this type of system does not contain impostors, FAR is not included in the definition. Accuracy of a biometric identification system is measured by applying a similar metric to that used for verification, but it also sums every class and divides the result by the total number of classes. The metrics used in this case are redefined as follows.

(a) Identification rate (IR) is the percentage of correctly predicted class images out of the total class images.

(b) False-negative identification-error rate (FNIR), also known as the miss rate, is the percentage of incorrectly predicted class images out of the total class images.

A real-world application of closed-set identification is searching for information on a registered criminal without inputting their name. This is useful for reducing the duration of or eliminating the need for manual checking of a large number of registered criminals against a suspect.

An open-set identification system, also known as negative identification, refers to a biometric task with no guarantee that the searched individual/class is enrolled in the database (Jain and Li, 2011, Tabassi *et al.*, 2014, Quinn *et al.*, 2016). This closely resembles the majority of operational biometric system conditions requiring two tasks:

(a) Detection – determine if an individual is in a database.

(b) Identification – find the record of the individual in the database.

Open-set identification is sometimes referred to as a "watchlist" task to differentiate it from the more commonly referenced closed-set identification. The accuracy measures used in this case are as follows:

(a) Detection and identification rate (DIR) is the percentage of correctly predicted class images out of the total class images.

(b) FNIR, also known as the miss rate, is the percentage of incorrectly predicted class images out of the total class images.

(c) False-positive identification-error rate (FPIR), also known as the false alarm rate, is the percentage of non-class images that are incorrectly detected out of the total non-class images.

Instead of the separate metrics above, detection and identification rates are sometimes combined to include impostors and referred to as DIR, with the included FPIR metric. Alternatively, FPIR and FNIR can be combined and referred to as FAR, which can be represented by a ROC curve with IR on the y-axis and FAR on the x-axis. When the false alarm rate (FPIR) needs to be differentiated from the miss rate (FNIR), the concept of ROC curves can be extended to more dimensions. A high false alarm value is of great concern for security and given good data, indicates that the system has low performance in general. A high FNIR is also of concern as it shows the algorithm does not distinguish well between known classes often resulting in a high false alarm value anyway. The threshold generally influences the performance of all metrics. Real-world open-set identification has applications such as an automatic watchlist for a known criminal on CCTV. The prediction result can possibly be an impostor, and thus a supervisor (usually an expert in the field) verifies the result for critical security applications. When such a system often matches the wrong individual in the database, the algorithm or database needs to be adapted accordingly otherwise the supervisor will have to be consulted continually.

A direct approach to evaluating identification performance is simply performing verification N times, where N is the total number of classes (often individuals). Therefore,

increasing either the number of classes or impostors results in a more challenging experiment.

2.7 Image Quality Issues

The performance of the recognition system depends on the quality of the sensed image for a particular modality, which in the case of image-based modalities, includes environmental conditions and damage sustained to the trait of a particular individual. According to the fingerprint quality standard proposed by Bausinger and Tabassi (2011), good quality fingerprints should present a high ridge-valley contrast with a clear ridge structure as shown in Figure 2.1a and 2.1b – collectively referred to as well-defined ridges – whereas a poor quality fingerprint has low ridge-valley contrast with corrupted ridges as shown in Figure 2.1c and 2.1f. However, each modality has a different quality standard and thus objective quality analysis is a non-trivial problem that requires further study, which falls outside the scope of this dissertation. Quality is thus subjectively evaluated in this research using visual inspection.

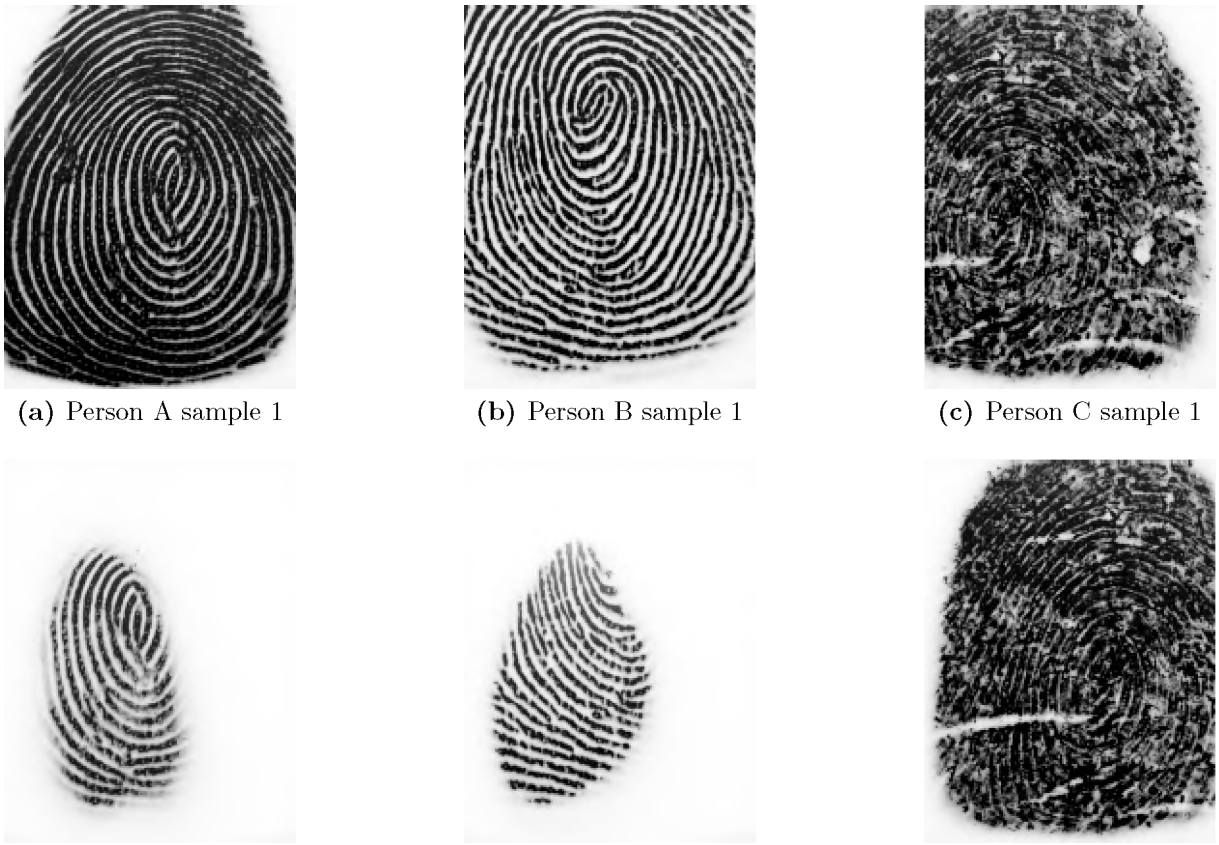
Figure 2.1d and 2.1e are referred to as partials – with and without a core, respectively, while Figure 2.1a and 2.1b show well-defined ridges.

Besides other conditions that negatively affect quality, such as low performing sensors or the incorrect use thereof, the following are examples of poor quality images of the face, palmprint and iris:

(a) Face – The use of makeup or occlusion of a face region with a scarf.

(b) Palmprint – Similar cases as the fingerprint, but including changing the pose of the hand, such as not spreading fingers or partially clenching, when using less controlled scanners.

(c) Iris – Not fully opening eyes or capturing under different lighting conditions, resulting in the stretching of the pupil.



(a) Person A sample 1 **(b)** Person B sample 1 **(c)** Person C sample 1

(d) Person A sample 2 **(e)** Person B sample 2 **(f)** Person C sample 2

Figure 2.1: Examples of fingerprint quality.

2.8 Summary

This chapter explained various concepts pertaining to biometric systems in general. An overview of biometrics and the different types of biometric systems were first presented. Four image-based biometrics modalities in the scope of our research and modality applicability were discussed next. Various aspects of the fusion of biometric modalities were introduced. Finally, metrics commonly used to evaluate biometric performance in both verification and identification systems were explained.

Chapter 3

Algorithms used in Biometric Systems

This chapter discusses algorithms pertaining to the recognition of fingerprint, face, palmprint and iris biometric modalities. Image preprocessing and enhancing algorithms common to all modalities are first explained, followed by algorithms specific to individual modalities. Furthermore, related studies of full biometric systems specific to individual modalities are introduced. In Chapters 5, 6, 7 and 8, a best effort is made to compare these related studies on a fair scale with the proposed algorithms.

3.1 Image Preprocessing and Enhancement

Various algorithms exist in the literature for enhancing images particular to the requirements of an application; these range from basic pixel-wise operations to a complicated contextual filtering. Commonly used methods in image-based biometrics are discussed in this section.

Image features often require pixel-wise enhancement that improves clarity and consistency over multiple impressions of the same class. The target of each of these algorithms is

particular to the biometric modality, for example, ridges and valleys in fingerprint and palmprint images, contours of the eye, nose and mouth region in face images, and the less visible arching ring and freckle details in iris images (Porwik and Wrobel, 2005).

Pixel-wise enhancement algorithms for biometrics are generally considered to be part of the fundamental image preprocessing tasks, which include lighting normalization and various noise filtering algorithms. These operations are designed to modify the underlying pixel while maintaining the texture pattern. They are thus, used during preprocessing to prepare input images for use in further processes such as feature segmentation.

3.1.1 Pixel Normalization

The pixel values of an image or certain image regions are set to a constant mean and variance to compensate for slight inconsistencies in lighting and contrast (Bovik, 2010). This is essential for normalizing image biometric data, which has a desired intensity range of 0 to 255. Given an image with pixel intensities ranging from 30 to 190, the normalization process is carried out by subtracting 30 from each pixel intensity followed by multiplication by $\frac{255}{160}$.

3.1.2 Histogram Equalization

Histogram equalization (HE) is effective at automatically setting the illumination across a dataset to a fixed amount (Bovik, 2010). The greyscale range is distributed uniformly by applying a non-linear transformation with a slight side effect on the histogram shape that is more pronounced in the case of high or low extremities in lighting. HE is often more effective than pixel normalization, but should be avoided in most histogram-based matching methods as it lowers the correlation between the histograms to be matched. Adaptive histogram equalization (AHE) attempts to mitigate this side effect, but can yield less desirable results than HE by over amplifying contrast, resulting in consistent

lighting at the cost of increased noise. Both pixel normalization and histogram equalization methods improve the consistency among multiple data samples and are thus useful in image-based biometric systems.

Contrast limited adaptive histogram equalization (CLAHE) differs from AHE in that it introduces contrast limiting to prevent over amplification (Sepasian *et al.*, 2008). The CLAHE method applies contrast limiting to each neighbourhood from which a transformation function is derived. This limits the amplification by clipping the histogram at a predefined value and redistributing the excess equally among all histogram bins. The application of CLAHE on image-based biometrics compared with other lighting normalization methods is not well-studied and thus requires empirical research. Common values limit the resulting amplification to between three and four.

3.1.3 Image Binarization

The image binarization algorithm designed by Otsu (1979) calculates the minimum variance between multiple peaks in the grey-level histogram image to select a threshold value automatically, instead of determining the optimal value empirically. This is effective on images containing highly contrasting regions. The steps in Otsu's algorithm are given below:

1. Compute histogram and probabilities of each intensity level resulting in a cumulative distribution function.

2. Step through all possible thresholds $t = 1, 2, 3.., 255$.

3. Compute means and variances.

4. Compute intra-class variance minimization function $\sigma_b^2(t)$.

5. Calculate the desired threshold as the lowest $\sigma_b^2(t)$ result.

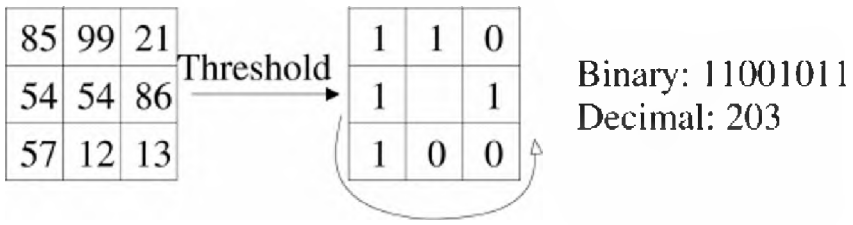


Figure 3.1: Basic 3×3 LBP operator.

3.1.4 Original Local Binary Pattern Operator

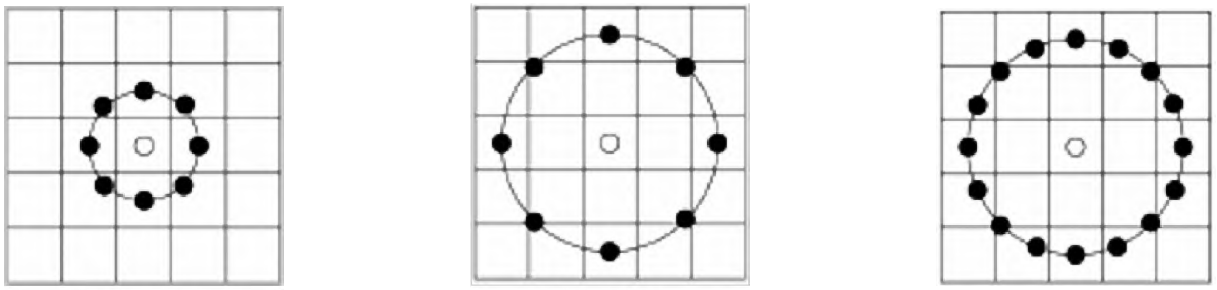
The basic local binary pattern (LBP) operator, introduced by Ojala *et al.* (1996), assumes that the most descriptive traits of image texture are two complementary aspects known as its pattern and strength. The size of the local area can be specified as a block of pixels. An example 3×3 pixel block LBP operator is shown in Figure 3.1. The pixels in this block are thresholded by the block's centre pixel value, multiplied by powers of two and summed to obtain the decimal value of the centre pixel known as the label. The neighbourhood of this centre pixel consists of 8 pixels, which can contain $2^8 = 256$ different label combinations depending on their pixel values relative to the centre. Neighbourhood values that are greater than or equal to the centre pixel are given the value one, while the remaining values are set to zero, as shown in Figure 3.1. In the example, the blank centre pixel is given the label 203, which gives LBP its vivid texture pattern. LBP thus operates as a feature extractor that is particularly aimed at enhancing texture patterns for matching. The texture pattern within the LBP mask can be extracted as a one-dimensional (1D) LBP histogram for matching.

3.1.5 Extended/Circular Local Binary Pattern Operator

Ojala *et al.* (2002) proposed a more robust LBP operator known as extended LBP (ELBP), which includes circular neighbourhoods. By enabling customization of the pixel mask, this operator can characterize the same texture at different scales, rotations and lighting. This extension is useful for texture alignment and matching, while calculating the binary result in the same clockwise manner. The circular local neighbourhood is defined as the set of sampling points evenly spaced on a circle where the number of sample points

and radius can be specified. By considering the global aspect of the image, increasing the number of sample points causes patterns to often appear brighter (depending on the neighbourhood), while increasing the radius size results in a sparser pattern with a less sharpened appearance. When a sampling point does not fall in the centre of a pixel, bilinear interpolation is used to give it a greyscale value.

Another extension is the grouping of uniform binary patterns. A pattern is defined as uniform if it contains at most two bitwise transitions from zero to one or vice versa. An example of uniformity is 00000000 and 01110000, which have zero and two transitions, respectively. An example of a non-uniform pattern is 1001001 – as it has four transitions. These features are patterns in the form of edges, corners and speckles and thus can be used for pattern matching (Ojala *et al.*, 2002, Ahonen *et al.*, 2006). These authors established that uniform patterns provide discriminative texture information unlike non-uniform patterns that are ambiguous. They ran further tests that show the effect of the radius and neighbourhood parameters by applying three different ELBP masks shown in Figure 3.2, on the FERET (Phillips *et al.*, 1998) face dataset. Using a radius of one and eight neighbours (1, 8) captured 90,6% of the pattern. The number of features are reduced when using a sparse mask such as (2, 8) as it retained 85.2% of the texture, while (2, 16) simplifies the pattern even further to 70% suggesting that a large number of neighbours can actually reduce the distinctiveness (over contrast) of the pattern. Therefore, ELBP can be used to both describe and reduce unwanted (and wanted) features surrounding pixel patterns.



(a) (1, 8) **(b)** (2, 8) **(c)** (2, 16)

Figure 3.2: Three different pixel masks defined by (radius, neighbours) [taken from (Ahonen *et al.*, 2006)].

3.1.6 Local Binary Pattern Variance Operator

The relatively new texture descriptor LBP variance (LBPV), characterizes the local contrast information into a 1D LBP histogram (Guo *et al.*, 2010). This operator differs from ELBP in that it computes the joint histogram of LBP and variance locally, but has the same custom mask that can be used as a texture descriptor for scales, rotations and lighting of an image. The differences in approach are explained as follows.

The principal orientations of the texture image are estimated and used to align LBP histograms of local regions. The aligned histograms measure the dissimilarity between regions where there is no overlap. Furthermore, LBPV is a training-free feature descriptor that requires no quantization and thus further reduces feature dimensions compared with ELBP. This is useful when a feature descriptor with low dimensionality is needed or where lighting variations should be considered both locally and globally.

3.1.7 NL-means Filter

This image denoising algorithm is described by its authors, Buades *et al.* (2005) as neither local nor global. NL-means differs from typical neighbourhood filters as it compares the geometrical configuration in an entire neighbourhood instead of a single greyscale pixel of one neighbourhood corresponding to another. NL-means preserves the edges of texture in an image. This is important in fingerprint applications as the accurate representation of ridges and valleys is key for minutiae singular point extraction[1]. However, high filter strength and large neighbourhood size can reduce the contrast between some ridges and valleys as shown in Figure 3.3a. The ridge-valley contrast is enhanced with an unsharp mask using a 15×15 Gaussian filter as shown in Figure 3.3b.

[1]NL-means was effective on fingerprints but only for improving keypoint detection (not postprocessing), in previous work (Brown and Bradshaw, 2016a,b)

(a) NL-means image **(b)** Enhanced image

Figure 3.3: Enhancing ridge-valley contrast with an unsharp mask.

3.2 Image Alignment and Segmentation

The biometric recognition process is often initiated by aligning the input image before feature extraction techniques are performed for improved feature discrimination (Bovik, 2010, Feng and Jain, 2011, Peralta *et al.*, 2014, Zou *et al.*, 2013). At the local level, there are unique points within biometric data that are different for each biometric modality. These points are especially useful as they serve as a guide during image registration for effectively normalizing image features. The points are used to align texture features such that they maximize similarities and differences between intra-class and inter-class images, respectively. Since the approaches for the different modalities are highly distinct, algorithms for fingerprint, face, palmprint, and iris are covered separately in the next four subsections.

Image segmentation is an obligatory preprocessing step in image-based biometric systems that relies on the precision of accurate image registration, to avoid overfitting or underfitting during the classification stage. It typically aims to reduce the search space by removing unwanted features from the original image, while retaining features that are highly representative of the underlying image class (Bovik, 2010). Although it depends on the biometric modality and the acquisition protocol used to capture the original image,

dimensionality can be substantially reduced by removing unwanted features.

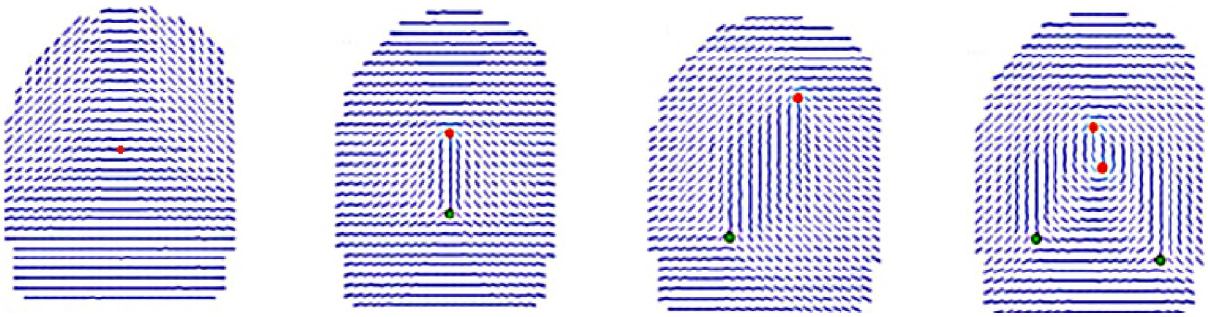
Omitting or incorrectly employing the image segmentation step typically causes the phenomenon known as "the curse of dimensionality problem" (Raghavendra *et al.*, 2011). The entire image can be considered for classification, but specific regions contain less noise and more discriminatory information on a per modality basis. The remaining features, known as the foreground, are hence referred to as the region of interest (ROI). The ROI should consist of data relevant to the identity of an individual while avoiding ambiguous data. Ambiguous data is defined as unrecoverable regions due to low quality images, or depending on the biometric, a mixture of foreground and background regions – increasing the number of similar features that images of different individuals share.

3.2.1 Fingerprint Algorithms

The fingerprint contains points, located on ridge curvatures, that are either unique or sharper than those in other areas. The core point, also known as the reference point, is often defined as the sharpest concave ridge curvature based on the orientation field of the fingerprint image (Jain *et al.*, 2000b). The following algorithms extract various features from the fingerprint; the algorithms covered are typically those used in related studies (Luo *et al.*, 2000).

Poincaré Index Core Detection

Many approaches to reference point determination critically rely on properties of the orientation field (OF), the most notable of which is the Poincaré index method (Jain *et al.*, 2000b). The OF of a fingerprint image is an important characteristic since subsequent steps in fingerprint recognition depend on its accurate estimation. One of these steps involves the determination of singular points, that is, where a ridge changes direction abruptly. The two main singular points are known as the core and delta. In Figure 3.4, the four main types of fingerprint patterns are shown with their core (red) and delta (green) points detected using the Poincaré index. The rare arch fingerprint type usually



(a) Arch fingerprint type

(b) Tented arch fingerprint type

(c) Right loop fingerprint type

(d) Whorl fingerprint type

Figure 3.4: Fingerprint pattern types.

outputs no core point for very flat cases as shown in the figure. However, Poincaré can still be used, but with a higher tolerance to detect a much smaller than typical peak.

The local ridge orientation at pixel $p(i, j)$ is defined as the angle θ that the ridge crossing forms through a $w \times w$ neighbourhood with the horizontal axis. The process is repeated for all neighbourhoods as seen in Figure 3.5. The popular approach by Jain *et al.* (2001) can be broken down into the following steps:

1. Divide the input fingerprint image into blocks of size $w \times w$.
2. Compute the gradients G_x and G_y at each pixel per block.
3. Estimate the local orientation of each block using the following set of equations:

$$V_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,\ v)\partial_y(u,\ v), \tag{3.1}$$

$$V_y(i,\ j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \partial_x^2(u,\ v)\partial_y^2(u,\ v), \tag{3.2}$$

$$\theta(i,\ j) = \frac{1}{2}\tan^{-1}\left(\frac{V_y(i,j)}{V_x(i,j)}\right), \tag{3.3}$$

The OF of all pixels yields the orientation map. With all OFs mapped, the Poincaré index of each block can be computed by summing the change in direction for each

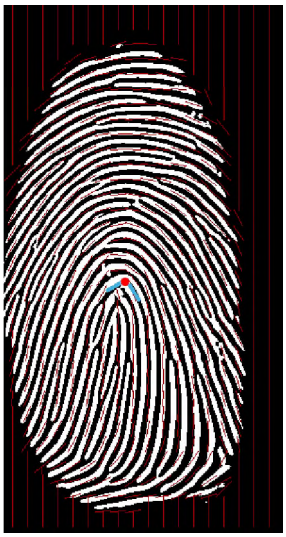


Figure 3.5: Orientation map.

pair of adjacent blocks. The largest change in direction is used as the core point as denoted by the small red dot in Figure 3.5.

Ridge Thinning

Ridge thinning is used to eliminate the redundant pixels of ridges to enable fast simple minutiae extraction (Maltoni *et al.*, 2009). The thinning is applied to a binarized fingerprint image by reducing certain pattern shapes until only 1-pixel wide lines remain, using morphological operations. It is critical that thinning algorithms are robust to aberrations of the binary-ridge boundaries that often result from noise in fingerprint images. Unwanted effects such as "spiky ridges" can result from the aberrations, which lead to the detection of spurious minutiae. Robust thinning algorithms therefore, should reduce each ridge to its central line without producing noise and the resulting thinned fingerprint image should not respond to further thinning iterations.

Zhang and Suen (1984) designed a thinning algorithm that uses a 3×3 sized block (shown in Figure 3.6), which is faster and more popular than other methods (Khanyile *et al.*, 2011). The algorithm processes the image by iteratively removing contour points until a skeleton image remains. Each iteration is divided into two subiterations. The

approach is explained in Algorithm 1, where $A(P_1)$ is the number of binary patterns in a clockwise direction from P_2 to P_9, and $B(P_1)$ is the total number of non-zero neighbours of P_1.

Algorithm 1 Ridge thinning developed by Zhang and Suen (1984).

1: **procedure** ZHANGSUEN
2: **while** points are deleted **do**
3: **for all** P(i,j) **do**
4: **if** $2 \leq B(P1) \leq 6$
5: **if** $A(P_1) = 1$, apply in order when an iteration is odd or even:
$isOdd : P_2 * P_4 * P_6 = 0$
$isEven : P_2 * P_4 * P_6 = 0$
$isOdd : P_4 * P_6 * P_8 = 0$
$isEven : P_2 * P_6 * P_8 = 0$
6: **then** delete $P(i, j)$

The disadvantage of this minimal overhead approach is that it can fail to correctly preserve ridges that have been reduced to 2×2 squares. Furthermore, it occasionally fails to preserve connectivity with diagonal lines and produces spurious line endings. Guo and Hall (1989)'s thinning algorithm improves the accuracy of Zhang and Suen (1984)'s work, but with increased complexity. It also uses a 3×3 sized block and two sub-iterations. Algorithm 1 is thus often modified to produce consistent skeleton images of both low and high quality fingerprint images (Boudaoud *et al.*, 2017, Kocharyan, 2013).

The modified Zhang-Suen thinning discussed here is based on the work of Kocharyan (2013). Algorithm 1 is modified to avoid deleting full ridges and end points, thereby preserving the topology of ridge connectivity. This is done by considering the following additional conditions.

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Figure 3.6: Pixel window for ridge thinning.

Figure 3.7: Ridge thinned fingerprint image.

During odd iterations, when $A(P1) = 2$, the following additional conditions are checked:

(a) $P_4 * P_6 = 1$ AND $P_9 = 0$ OR

(b) $P_4 * P_2 = 1$ AND $P_3 * P_7 * P_8 = 0$

During even iterations, when $A(P1) = 2$, the following additional conditions are checked:

(a) $P_2 * P_8 = 1$ AND $P_5 = 0$ OR

(b) $P_6 * P_8 = 1$ AND $P_3 * P_4 * P_7 = 0$

Minutiae Extraction

Minutiae extraction and labelling is performed using the skeleton image as input. Minutiae extraction is performed using the crossing number method and typically uses a 3×3 pixel

(a) Database fingerprint image **(b)** Input fingerprint image

Figure 3.8: Minutiae extraction.

sliding window (Bovik, 2010). This effectively creates eight-directional minutiae masks, where white pixels p, are denoted as the ridge. The crossing number equation is given as:

$$CN = 0.5 \sum_{i=1}^{8} |P_i - P_{i+1}| \tag{3.4}$$

The crossing number $CN(p)$ is verified based on the non-ridge pixel (black) count per 3×3 block. An ending minutia is defined as the block with $CN = 1$, while a bifurcation minutia is defined to have $CN = 3$. Each minutia is labelled based on its x- and y-coordinates and the direction θ of the ridge associated with it, which is obtained from the OF. Figure 3.8 shows corresponding database and input fingerprint images, where each image has the core point as a small red dot, ridge endings as green circles and bifurcations as cyan circles.

Fingerprint Segmentation

Fingerprint texture patterns are known to contain richer information than singular points and minutiae (Maltoni *et al.*, 2009). The texture patterns consist of ridges and valleys on the surface of the finger, which are collectively considered to be the foreground. The keypoints typically consist of a combination of detected minutiae/singular points for a consistent segmented region.

3.2.2 Face Algorithms

The keypoints on the face are the eyes, nose and mouth, and these are used for alignment. Each outer keypoint is connected, resulting in a border around the face, which helps to avoid typical changes to the features on and around the face such as hairstyle, occluded ears and neck scarves. However, an initial detection window around the face is often used to reduce the search space.

Constructing a consistent face recognition system under these conditions is an ongoing research area and requires a new approach. Recent advances include an automatic face alignment system requiring milliseconds per face (Kazemi and Sullivan, 2014) and the frontalization of all face images up to 60° pose angles (Sagonas *et al.*, 2015, Haghighat *et al.*, 2016). Face systems that utilize 3D modelling generally produce high recognition rates and versatility, but require greater computational power and more training data (Kafai *et al.*, 2016).

Haar Object Detection

Viola and Jones (2001) use AdaBoost to train progressively complex region rejection rules based on Haar-like wavelets and space-time differences. A large number of these Haar-like features are organized to create a classifier cascade. Haar cascading is a popular method for selecting the detection window on the face as it has low prediction time, but requires a large amount of training data and time.

Figure 3.9: Face detection using HoG features, designed by Felzenszwalb *et al.* (2010).

HoG Object Detection

Dalal and Triggs (2005) presented an alternative face detection method using Histogram of Oriented Gradients (HoG) features as input for training a classifier. This robust generic object detector is shown to perform well when trained and tested on face images, and outperforms the Haar cascading method.

Felzenszwalb *et al.* (2010) also proposed a face detection method, but with improved HoG features. This HoG descriptor is a popular method used in face related literature, where the set of features that match the description are scored – using distance matching or a classifier – to find a rectangular region known as a window. Both Dalal and Triggs (2005) and Felzenszwalb *et al.* (2010) use a sliding window approach where the highest scoring window is selected using Non-Maxima Suppression (NMS). The goal of NMS is to retain only one window by eliminating those with low scores by thresholding and those that overlap. However, this NMS method is described as greedy in its discarding process and thus does not always find the best solution.

King's Improved Window Scoring of HoG Features

King (2015) uses an improved peak sorting algorithm that solves a typical situation shown in Figure 3.10 by training a model using the novel approach known as Max-Margin Object Detection (MMOD), on a less greedy (lower threshold) NMS result that can keep multiple

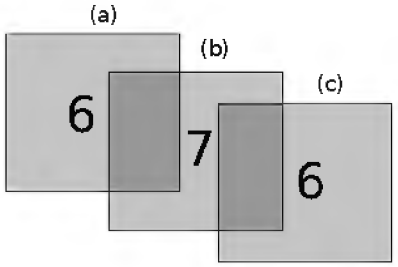


Figure 3.10: Suboptimal window (b) scores 7, while optimal windows (a, c) have a combined score of 12 [taken from (King, 2015)].

sub windows, thereby limiting the average loss of detections per training image. This leads to improved face detection that requires less positive training data and no negative training data, which can also result in faster computation. Considering the example in Figure 3.10, window scoring using the suboptimal method selects window (b) with a score of 7, and rejects the two windows (a, c), whereas King's method selects both windows resulting in a noticeably improved score ($6 + 6 = 12$).

MMOD's aim is similar to the popular support vector machine (SVM) classifier as it finds the maximum margin that satisfies some constraints with similar (SVM) parameters, but enables grouping of multiple subwindows with the assumption that entire HoG windows of training and test images do not always yield the best match. The HoG detector operates by creating an image pyramid and scanning the HoG detector over each pyramid level in a sliding window fashion. The image pyramid downsamples at a ratio of $\frac{5}{6}$. The window that passes the scoring function is part of the face detection result. Given image x and the window scoring function f, the ideal object detection procedure is defined by King (2015) as:

$$y^* = \underset{y \in v}{\operatorname{argmax}} \sum_{r \in y} f(x, r) \tag{3.5}$$

Face Landmarks Prediction

Face keypoints are often referred to as landmarks in the literature. The breakthrough research by Kazemi and Sullivan (2014) enabled significantly faster prediction of face

landmark points. The authors use an ensemble of regression trees to train and predict a set of keypoints that can be used to accurately align and segment the face before the classification stage. Moreover, face landmarks are automatically estimated, directly from a sparse subset of pixel intensities, allowing execution performance to surpass most realtime requirements at < 1 ms per image.

The landmark training stage uses an initial guess of the shape based on a set of randomly generated candidate splits of a given face, and selects the candidate that minimizes the sum of the squared error. A decision is made based on the difference between two pixel intensities at each split node in the regression tree. This defines coordinates of the mean shape of the face. Shrinkage and averaging are used as regularization methods in the gradient boosting learning procedure. In the case of shrinkage, each regression tree is multiplied by a learning rate parameter $0 < v < 1$, while averaging involves individually training multiple trees.

During the landmark prediction stage, the mean shape is warped according to adjacent triplet keypoints of the learned model by calculating a similarity transform between the corresponding coordinates. The time complexity is reduced by calculating the transform and warping only once at each level of the cascade. Missed landmarks are catered for by using the weighted average of surrounding targets, and fitting regression trees to the weighted residuals. The resulting keypoints can be used to directly segment the face or first perform further processing such as pose alignment.

3.2.3 Palmprint Algorithms

Palmprints contain many features in common with fingerprints, but they also contain three flexion creases called principal lines and secondary creases called wrinkles (Ross *et al.*, 2006). Only the principal lines are genetically dependent, while the wrinkles are unique even in identical twins. Furthermore, both types of creases can be effectively classified in low or high resolution images. A low resolution palmprint image is shown in Figure 3.11. On the other hand, minutiae-based classification in palmprints requires higher resolution images of at least 500 dots per inch (DPI) (Jain *et al.*, 2011).

Finger Valley Approach

A popular method for segmenting the palmprint, used extensively throughout the literature (Anitha and Rao, 2015, Li *et al.*, 2012, Methani and Namboodiri, 2009, Wang *et al.*, 2007, Zhang *et al.*, 2003), first requires the detection of valleys between the fingers, which can be used as keypoints. These keypoints are centred at the index-middle and ring-little finger valleys as depicted in Figure 3.12. The index-thumb valley is generally not used for palm detection, but rather to eliminate the thumb for easier palmprint segmentation (Khan *et al.*, 2014).

The finger valley keypoints are used to form the coordinate system for the segmentation proposed by Zhang *et al.* (2003). The system operates as follows:

1. The original image is binarized resulting in Figure 3.13a.
2. The boundary search algorithm starts at the bottom leftmost pixel and searches vertically for white pixels perpendicularly to the right with respect to the intermediate gradient until a black pixel is found within the defined threshold (typically 50 pixels), as seen in Figure 3.13b. If a black pixel is not found, the image is rotated 90° and repeated until a black pixel is found within the threshold.

Figure 3.11: Principal lines and wrinkles are visible in a 75 DPI palmprint image from PolyU palmprint database.

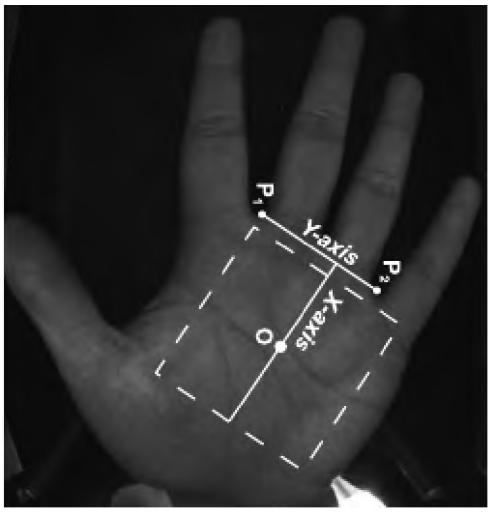


Figure 3.12: Main valley points (P_1 and P_2) [taken from (Zhang *et al.*, 2003)].

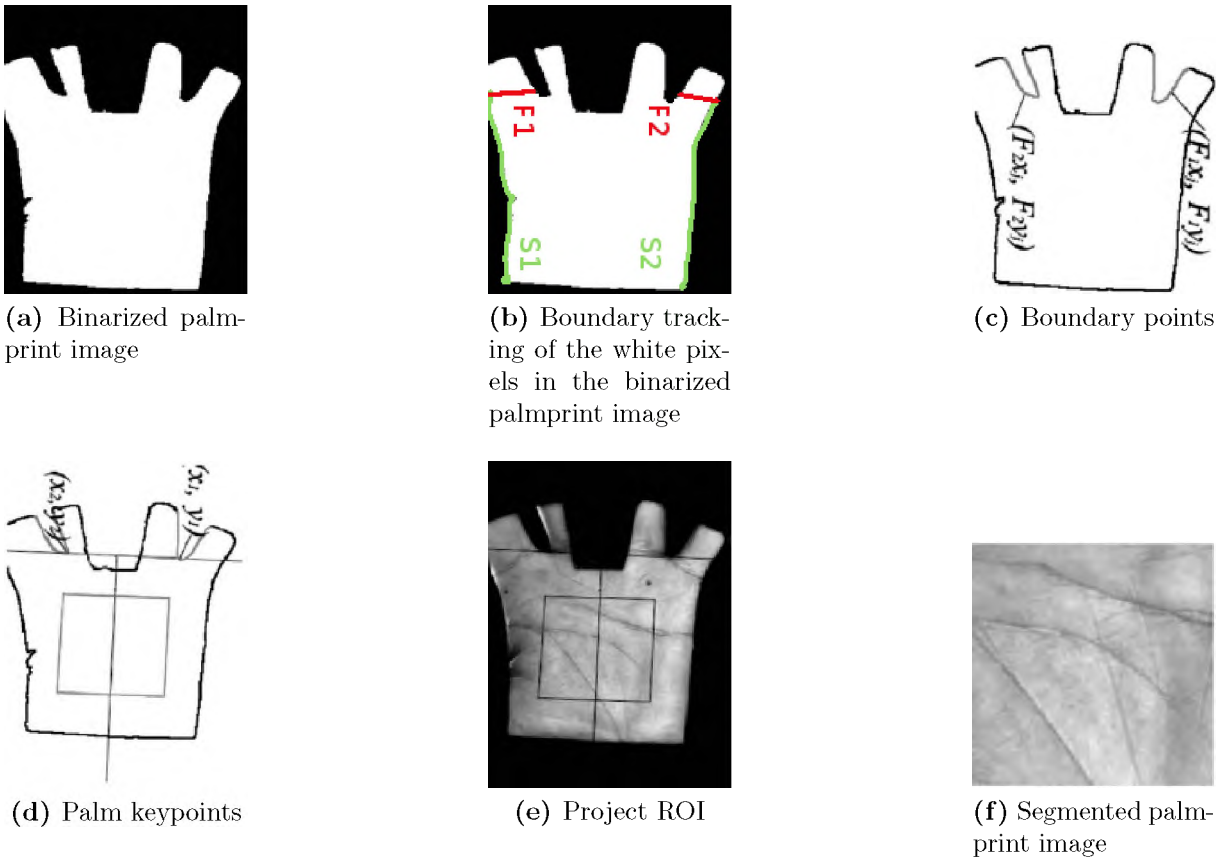
3. The square ROI is thus bounded by $F_i x_j, F_i y_j$ $(i = 1, 2)$ valleys, as shown in Figure 3.13c.

4. The tangent of the valleys is computed by satisfying the inequality $F_i y_j \leq m F_i x_j$ for all i and j.

5. The line perpendicular to the midpoint of the tangent is the centre of the square ROI, as shown in Figure 3.13e.

Maximum Inscribed Circle Approach

A maximum inscribed circle (MIC) makes contact with three or more captured data points and contains no data points within the solution (Zhang, 2004). Calculating the MIC of the white binary pixels of the hand, that is, hand pixels, finds the largest circle that can fit within the white pixels without discontinuity. This is useful for aligning and segmenting the palmprint from hand images without fingers or with a vague finger region. However, it can also be combined with the finger valley keypoints for improved accuracy.
Zhang increased the radius of a circle with the centre starting at the centre-most hand pixel and shifting to adjacent pixels in four directions, iteratively for every zero pixel reached. The resulting MIC is shown in Figure 3.14b. The following two works build on Zhang (2004)'s coordinate system.



(a) Binarized palmprint image

(b) Boundary tracking of the white pixels in the binarized palmprint image

(c) Boundary points

(d) Palm keypoints

(e) Project ROI

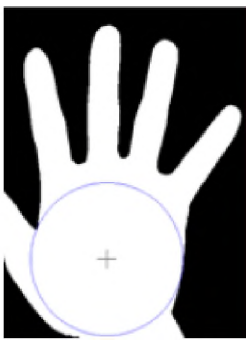
(f) Segmented palmprint image

Figure 3.13: Palmprint segmentation using valley points [taken from (Zhang *et al.*, 2003).

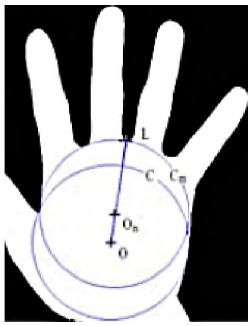
Ding and Ruan (2006) reported rotation, translation, and scale invariance by shifting the MIC toward the middle-ring finger valley known as the origin, as shown in Figure 3.14c. They argue that the MIC is often found close to the heel of the hand, which contains redundant information. The shifted MIC is called the maximum effective circle (MEC).

Choge *et al.* (2009) used the same coordinate system proposed by Zhang *et al.* (2003), but found the centre of the palmprint such that the radius passes through the index-middle and ring-little finger valleys. The circular ROI obtained is unwrapped in the form of a rectangular ROI with a fixed size. These approaches, however, do not cater for changes in either the circle centre or radius when the pose of the palmprint is changed and are thus, not optimized for palmprint images obtained via contactless sensors.

(a) Binarized palmprint image



(b) Zhang (2004) MIC result



(c) Ding and Ruan (2006) MIC result with MEC

Figure 3.14: Palmprint segmentation using maximum inscribed circle.

3.2.4 Iris Algorithms

Iris sensors often capture the periocular region if the hardware does not automatically extract the iris (Ross *et al.*, 2006). The periocular region consists of the eye and other features in close proximity, such as the eyebrows. While features other than the iris also contain discriminatory information. they can be dynamic, and are therefore, undesired.

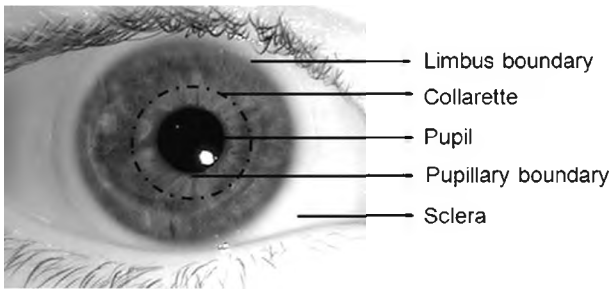


Figure 3.15: Periocular region [taken from (Jillela and Ross, 2016)].

Iris Segmentation

Iris segmentation is the process of pruning the image of everything but the region lying between the inner and outer boundary keypoints of the iris, known as the pupillary and limbus, in an image usually consisting of the periocular region (Jillela and Ross, 2016).

Two popular techniques used to determine keypoints for segmentation of iris images are the integro-differential operator and the Hough transform. Figure 3.15 shows the periocular region – typically captured by "iris sensors" – containing the iris, which lies between the pupillary and limbus boundaries.

Integro-differential operator

The integro-differential operator (IDO) proposed by Daugman (2004) performs a separate brute-force search for the centre and radius of the iris. This is achieved by calculating the maximum in the blurred derivative with respect to the circular trajectory formed by the increasing radius of normalized contour integrals. Parameters are specified for the minimum radii for the pupillary and limbus boundaries. The resulting common centre coordinates and radii coordinates are used as keypoints for the segmentation process. The obtained keypoints allow the iris to be segmented in image $I(x, y)$ by first applying IDO on the limbic boundary of the iris, which can be expressed as:

$$\mathbf{max}(r, x_0, y_0) = \left| G_\sigma(r) * \frac{\partial}{\partial t} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \tag{3.6}$$

where

$$G_\sigma(r) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-} \left(\frac{(r - r_0)^2}{2\sigma^2} \right) \tag{3.7}$$

represents the radial Gaussian smooth operation with scale σ and centre r_0. The convolution operation is denoted by $*$, with radius r of the circular arc ds, which is centred

at (x_0, y_0). Lastly, dividing by $2\pi r$ normalizes the circular integral with respect to its perimeter.

The same process is used for the pupillary boundary, except that a coarser scale of convolution σ is used to differentiate between the limbic and pupillary boundaries.

HT-based iris segmentation

The Hough Transform (HT), proposed by Wildes (1997), calculates the pupil and iris boundaries using binary edge maps from the Canny edge detector algorithm (Canny, 1986). Votes are accumulated to estimate the parameters of the boundary concentric circles. The voting procedure is carried out in parameter space, from which circular contour candidates are obtained as local maxima in an accumulator space. Keypoints are obtained similarly to the IDO approach.

The use of HT for iris segmentation is made possible by fitting the obtained circle x_c, y_c, r, where x_c and y_c denote the centre coordinates and r the radius. The approach is performed as follows. Given an iris image I(x, y), the edge map is highlighted by thresholding the magnitude of the image intensity gradient:

$$|\nabla G(x, y) * I(x, y)| \geq th, \tag{3.8}$$

where $\nabla \equiv (\partial/\partial x, \partial/\partial y)$, and th denotes an empirically chosen intensity threshold. For a set of edge points, $(x_i, y_i), i = 1, 2, ..., n$ a Hough transform detects circular contours using accumulator array $H(x_c, y_c, r)$, which is computed as (Jillela and Ross, 2016):

$$H(x_c, y_c, r) = \sum_{i=1}^{N} h(x_i, y_i, x_c, y_c, r) \tag{3.9}$$

where

$$h(x_i, y_i, x_c, y_c, r) = \begin{cases} 1, & \text{if } g(x_i, y_i, x_c, y_c, r) = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

and

$$g(x_i, y_i, x_c, y_c, r) = (x_i - x_c)^2 + (y_i - y_c^2) - r^2 \tag{3.11}$$

For each edge point (x_i, y_i), $g(x_i, y_i, x_c, y_c, r)$ is set to zero if the parameter triplet (x_c, y_c, r) represents a circle through that point. The parameter triplet that maximizes H is selected to represent the contour of interest. The maximizing parameter triplet is determined by first building $H(x_c, y_c, r)$ as an array and scanning for its largest value.

The same thresholding and voting process is used to determine the pupillary boundary, but with the following changes:

- The edge detector is not directionally tuned because the pupillary boundary is less prone to occlusion from the eyelids.
- The parameter values (x_c, y_c, r) are constrained within the limbus bound circle instead of within the entire image.

Jillela and Ross (2016) show that IDO and HT methods share similar accuracies and are both susceptible to over or under segmentation when presented with non-ideal iris images. While not by a large margin, the resulting offset affects the recognition accuracy. They use the HT method and combine it with a gradient-based edge detector tuned for horizontal edges to find the upper and lower eyelids to cater for their occlusion. The final boundary is cropped as shown in Figure 3.16.

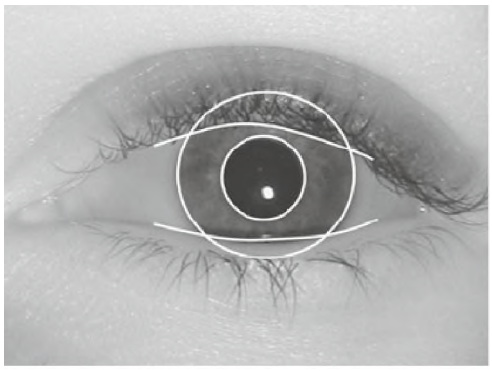


Figure 3.16: Keypoints for iris segmentation.

3.3 Feature Normalization and Extraction Algorithms

3.3.1 Feature Normalization within the ROI

Feature normalization algorithms, described in Section 3.1, are typically used before aligning keypoints of biometric images for a more accurate segmentation result. This lowers the intra-class variation of entire images (global), but the lighting is often dependent on local regions and thus a secondary pass can be of benefit. Applying lighting normalization after segmentation can further reduce intra-class variations as the segmented image is the ROI – local regions used for the classification step. However, greyscale pixels within the ROI often contain redundant information that reduces efficiency of data separation performed by classifiers. This is especially the case when scaling a biometric system to contain a large number of classes, that is, the curse of dimensionality. Feature extraction is a means of reducing the dimensionality by pruning features on the extremities of the spectrum, which often contain redundant information. This increases the potential for high inter-class separation for better discrimination between different classes.

3.3.2 Feature Extraction using Frequency Filters

The quality of the input image plays an important role in the performance of the feature extraction and matching algorithm (Chikkerur *et al.*, 2007). Bandpass filters that can be

tailored to remove low and high frequencies, increase the amplitude of the DC[2] component of the image. This has the effect of increasing the dominant spectral components while attenuating the weak components.

The Gabor filter[3] is often used to filter frequencies based on the texture patterns of images such as ridges of the fingerprint. It is constructed using a special short-time Discrete Fourier transform (DFT) by modulating a 2D sine wave at a particular frequency and orientation with a Gaussian envelope. A Gabor filter requires much tuning for specific orientations and frequencies to isolate the undesired noise while preserving the structure of a particular biometric (Budhi *et al.*, 2010).

The Laplacian of Gaussian (LoG) filter can remove unwanted features on the high frequency spectrum before enhancing the remaining features, effectively increasing the DC component (Li *et al.*, 2005). However, a side effect can occur when applying the filter to badly aligned or poor quality images. This side effect is also prevalent in images with inconsistent lighting. The increased feature discrimination of LoG further highlights differences among multiple samples, leading to bad training and testing sets and consequently a lower recognition accuracy. Appropriate ROI selection during the segmentation of biometric images and subsequent feature normalization is thus imperative to the success of an image-based feature-fusion biometric system.

3.4 Feature Transformation and Classification

Image classifiers aim to express the most relevant image properties obtained from feature extraction, and typically produce a decision based on them (Chapelle *et al.*, 1999). This often requires feature transformation, which expresses a feature vector in an alternate space – this is a subspace representation of the features that improves discrimination

[2]Here DC is simply analogous to direct current. It is the part of a signal where the average positive and negative half cycles do not equal zero, and often carries the richest information.

[3]This was tested in our previous work and shown to be inconsistent, while requiring much tuning to be effective (Brown and Bradshaw, 2016a,b).

and allows for further feature extraction. As a result, this representation can reduce dimensionality and intra-class variation, allowing for high inter-class separation by better categorizing a particular class against other classes in the same dataset. The implementations of the following image texture classification methods considered are biased more to the overfitting of data in a model than underfitting, which means these classifiers benefit more from the feature reduction techniques described in Section 3.3.2 (Brown and Bradshaw, 2017a). However, after feature reduction, these texture-based features still have a relatively high dimensionality compared with point/minutiae-based features, which may require kernel-based classifiers that shift data to a higher dimension. The classifiers discussed in this section are thus all linear and cope well with texture-based features. The three classifiers discussed in this section are Eigen, Fisher, and LBPH. These classifiers were popular in traditional face recognition applications. In the case of Eigen (similarly for Fisher), this was known as the classic "Eigenface" approach (Belhumeur *et al.*, 1997). For simplicity purposes of this thesis, Eigen, Fisher, LBPH and SVM are all referred to as classifiers or methods that determine the class of an individual.

3.4.1 Eigen

To achieve ideal inter-class separation, the unique features relevant to a biometric modality needs to be isolated effectively. Statistical classifiers that operate in the Eigen space aim to maximize total scatter of the data to discriminate effectively between different individuals (Belhumeur *et al.*, 1997).

This Eigen classifier relies on principal component analysis (PCA) to transform statistically significant features that define a given set of feature (Eigen) vectors. These feature vectors are known as principal components, which can be used to construct an efficient model from samples to represent a given class. In other words, PCA performs feature extraction in Eigen space which allows the Eigen method to accept unprocessed feature vectors by simply selecting fewer principal components. The classification decision is made with preference to the nearest neighbour among eigenvalues between the trained model and the model to be tested during matching. The nearest neighbour is calculated as the

Euclidean distance among eigenvalues. The k value of the nearest-neighbour classifier is set at $k = 1$.

Given N sample images x, the total scatter matrix is defined as (Belhumeur *et al.*, 1997):

$$S_t = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T \tag{3.12}$$

where $\mu \in \mathbb{R}^n$ is the mean image obtained from the samples.

3.4.2 Fisher

Although PCA is effective at maximizing the total scatter of data and reducing dimensionality, it does not consider the rest of the classes when characterizing a particular class. Under low noise or ideal conditions, this loses discriminative information on an inter-class level (Belhumeur *et al.*, 1997). Linear discriminant analysis (LDA) aims to solve this by performing additional class-specific dimensionality reduction. This Fisher classifier is thus defined when LDA is used instead of PCA in the Eigen classifier procedure of the previous subsection. When the data is not noisy, Fisher is known to cluster same class data tightly and to better separate different classes in a lower-dimensional representation.

Given N sample images x and C classes, the inter-class scatter matrix is defined as:

$$S_b = \sum_{i=1}^{C} N_k(\mu_k - \mu)(x_k - \mu)^T \tag{3.13}$$

and the intra-class scatter matrix is defined as:

$$S_w = \sum_{i=1}^{C} \sum_{x_k \in \mathbb{X}_i} (x_k - \mu)(x_k - \mu)^T \tag{3.14}$$

The equations show that the k value of the nearest-neighbour classifier is dependant on the number of samples per class. The dimensionality is guaranteed to be lower than

the Eigen method as $C - 1$ is the maximum number of non-zero generalized eigenvalues (Belhumeur *et al.*, 1997). The disadvantage of the Fisher method is that inconsistent data within classes such as inconsistent illumination or segmentation have a greater negative impact on classification performance than the Eigen method. Additionally, Fisher is more likely to underfit data than Eigen, but the reduced dimensionality that LDA offers allows Fisher to have lower training and substantially lower testing times. This is especially noticeable when using datasets with high numbers (> 5) of samples per class.

3.4.3 LBPH

Regular histogram matching is one of the simplest image matching methods. Local Binary Pattern Histograms (LBPH) is another texture feature descriptor for images that aims to capture a richer set of features and has been shown to achieve better performance than other histogram matching methods (Ahonen *et al.*, 2006). As explained in Section 3.1.5, a local binary pattern operator transforms the neighbour pixel to a one if it is greater than the centre pixel and a zero if it is smaller than the centre pixel. LBPH uses circular neighbourhoods (ELBP) that are uniform such that the histogram has a separate bin for every pattern neighbourhood, and can group the same texture with a different lighting, scale and rotation. This results in the distinct advantage of lighting, scale and rotation invariant texture classification compared with Eigen and Fisher. LBPH uses a weighted Euclidean distance measure, known as Chi-squared distance, when performing nearest-neighbour classification, similarly to the Eigen method.

Given m circular neighbourhoods, their corresponding spatially enhanced histograms are determined, with a size of $m \times n$, where n is the length of a single histogram. Training the advanced histogram model is also substantially faster that Eigen and Fisher. Furthermore, the training time is not significantly affected by the image resolution and it produces the smallest model size of the classifiers discussed in this section. However, testing time is significantly higher than Eigen and Fisher and is directly proportional to the number of circular neighbourhoods.

3.4.4 Support Vector Machines

The SVM is a supervised statistical learning model that has been used extensively in pattern recognition problems (Brown and Bradshaw, 2017a, Crammer and Singer, 2001, Chapelle *et al.*, 1999). Its advantage over many other classifiers is that the training time is not significantly affected by the high dimensionality of feature vectors from images. Although the SVM was purposed to solve binary class problems, it was extended to support multi-class problems (Crammer and Singer, 2001). In a binary class problem, SVMs aim to maximize a mathematical function in higher-dimensional space, given a collection of data points that consist of two classes that can be separated by a decision boundary.

Consider the two classes $S^+ = \{x_i \mid y_i = 1\}$ and $S^- = \{x_i \mid y_i = -1\}$ that are linearly separable (Crammer and Singer, 2001). This means that there is at least one boundary that can be formed between them. The data points of sets S^+ and S^- that are located on the boundaries of the margin are known as the support vectors. A simple rescale of $\boldsymbol{w}$ for all x_i that are support vectors holds that:

$$\boldsymbol{w} \cdot x_i + b = 1 \tag{3.15a}$$

$$\boldsymbol{w} \cdot x_i + b = -1 \tag{3.15b}$$

The distance d between the decision boundary and the margin can be expressed as:

$$d = \frac{2}{|| \boldsymbol{w} ||} \tag{3.16}$$

In higher-dimensional space, the decision boundary that achieves the maximum margin between sets S^+ and S^- is known as the optimal hyperplane, which allows an SVM to model new data points efficiently.

3.5 Existing Biometric Systems

This section discusses well known related biometric systems as well as those using specific publicly available datasets that are of interest in this research.

3.5.1 Related Fingerprint Recognition Systems

Jain *et al.* (2000b) proposed FingerCode, which uses a tessellation of filtered fingerprint images centred at the reference point as a circular ROI containing 80 sectors. The ROI is further processed to generate eight-dimensional feature maps by computing their average absolute deviation features. Multiple FingerCode templates are created by rotating the resulting ROI by up to 45° in both directions to handle unaligned images. The matching performance of this method is directly proportional to the accuracy of determining the reference point, which is based on the quality of the fingerprint images. However, this FingerCode method cannot guarantee that a reference point will be found on every type of fingerprint image, such as the arch-type, and on very poor quality fingerprint images.

Xu *et al.* (2009) used a novel method referred to as spectral minutiae representation, which represents a minutiae set as a fixed-length feature vector that is invariant to translation and scaling. The experiments show that applying PCA and the DFT feature reduction algorithms on spectral minutiae effectively decrease feature dimensionality by 94%, without degrading the recognition performance. Subjects 101 – 144 of the MCYT dataset were tested and resulted in a 99% GAR (at 0% FAR) verification accuracy. Data for these particular test subjects are, however, not freely available.

The study by Simon-Zorita *et al.* (2003) evaluates the large MCYT fingerprint dataset focussing on the effect of position variability. This dataset was categorized as low, medium and high control – referring to the restriction placed on positioning the finger during capturing. The authors employ a generic fingerprint feature extraction process using image enhancement, thinning and minutiae extraction as explained in earlier sections of this chapter. Their fingerprint matching approach is a multiple-reference strategy that

uses the highest similarity score when using multiple training samples. They show that the samples with the most similarly positioned fingers achieve the highest similarity score while samples with low levels of control show significantly reduced performance.

The approach adopted by Chen *et al.* (2006b) focuses on improving matching by respecting non-linear distortions on the challenging FVC2004 dataset. The feature extraction process is generic, but only minutiae greater than 5-pixels distant from other pixels or from the border are captured. They use the normalized fuzzy similarity measure, which takes into account that the further minutiae pairs are apart the greater is the coordinate variance between matching (the same) minutiae of the database and input images. They compared their approach on the FVC2004 dataset to the traditional approach of Luo *et al.* (2000), which does not include the normalized fuzzy similarity measure.

Zuobin *et al.* (2017) use a subspace approach similar to PCA and LDA, but which retains more components that can be used as discriminative information during nearest-neighbour matching. This differs from traditional fingerprint recognition algorithms because it focuses on multimodal biometric fusion at the feature level by testing the SDUMLA-HMT dataset. The feature extraction methods are not specified. The authors propose an approach called intra-class and extra-class discriminative correlation analysis with the aim of eliminating intra-class correlation, while retaining sufficient dimensions of features for correlation analysis of inter-class differences. Moreover, their approach is said to outperform LDA at reducing intra-class correlation.

3.5.2 Related Face Recognition Systems

A well-known recent deep learning study by Facebook, referred to as DeepFace (Taigman *et al.*, 2014), uses a nine-layer neural network with over 120 million parameters. This enables it to achieve high recognition accuracy at various pose angles by training with over 4 million labelled faces. Disadvantages of this system are the requirement of large amounts of training data that are not necessarily readily available and substantial processing power. Systems tested on popular datasets are described below.

Yi *et al.* (2013) proposed a robust face recognition algorithm using a 3D deformable model aimed at estimating the pose of face images with large pose variations. A set of Gabor filters is transformed according to generated parameters of the model into a feature vector. PCA is applied to capture what is deemed most relevant data from the feature vector. The "half-face" method, which uses the least occluded side of the face under occlusion conditions, is the key component to their approach as it improved system accuracy by 46%. The dot product is used to compute the similarity between resulting feature vectors of the training and test face images for matching. This method offers a significant improvement on face data only with large pose variations. The authors deduced that "traditional" approaches are only sufficient for low pose variation face data, which is often limited to frontal images.

A recent approach by Haghighat *et al.* (2016) automatically segments the face, performs frontalization and classifies the result without 3D modelling or deep learning. This is similar to another face recognition approach by Sagonas *et al.* (2015). The former approach however, includes the half-face method for viewing angles that are greater than 45°, which is not catered for in the method by Sagonas *et al.* Haghighat *et al.* aim to frontalize all faces (input and database) using a base mesh per individual without face detection preprocessing. The image features include a fusion of HoG and a vast number of Gabor features using canonical correlation analysis, which involves finding linear combinations of variables with the highest correlation. The resulting feature vectors are matched using minimum distance classification (nearest neighbour) of the discrete cosine transform, which is similar to DFT but uses a function with data independent bases that capture a different pattern in the frequency spectrum. The disadvantage of this approach over other feature extraction methods is the substantial increase in complexity of 40 Gabor filters in five scales and eight orientations. Furthermore, near-frontal training samples are highly favoured over profile faces for achieving high accuracy, although performance is significantly better than traditional approaches.

The half-face frontalization method has been shown to improve the results for systems by both Yi *et al.* and Haghighat *et al.*, outperforming nine similar systems (Haghighat *et al.*, 2016). Moreover, the study showed that Haghighat *et al.*'s system outperforms the 3D

deformable model of Yi *et al.* on the FERET b-series (Phillips *et al.*, 1998) face dataset. Table 3.1 illustrates the improved performance of Haghighat *et al.*'s system compared to the second and third best systems.

Table 3.1: Face recognition rates of different approaches at eight viewing angles on the FERET b-series dataset (in %).

Method	**+60°**	**+45°**	**+25°**	**+15°**	**−15°**	**−25°**	**−45°**	**−60°**
Haghighat *et al.* (2016)	91.5	96	100	100	100	100	99	93.5
Sagonas *et al.* (2015)	–	96	100	100	100	99	96.5	–
Yi *et al.* (2013)	93.75	98	98.5	99.25	99.25	98.5	98	93.75

Samet *et al.* (2016) proposed a face recognition system that uses a different method for dealing with pose variations; it assumes that the input and database face images of different poses can be represented by a linear combination of corresponding vectors. First it extracts eigenvectors using a 2D PCA. Linear regression is applied to reduce the Eigen coefficients of low variance principal components of the input face image. The mean squared error of the transformation result for the input image is compared with the database image. The process is repeated until the transformation matrix yields the lowest error, and is used to align the input image according to the database image. The Mahalanobis distance is used for matching images of the Fei dataset, which consists of images with large pose variations.

Shams *et al.* (2016) provide limited results on the face as part of a larger multimodal biometric study. Segmentation is performed on the face using Canny edge detection and a binary rectangular mask using unspecified parameters to produce a 64×64 fixed-size ROI. The feature extraction is performed on the ROI using three LBPV operators. The parameters for each are given as (1, 8), (2, 16) and (3, 16). The results of three LBPV feature vectors are weighted and classified using a combined learning vector quantization classifier, which is a neural network that uses the winner-take-all Hebbian learning-based approach.

3.5.3 Related Palmprint Recognition Systems

Traditional palmprint recognition systems are more recent than those using the fingerprint and face (Ross *et al.*, 2006). Subspace and principal line approaches are popular as they do not require high DPI sensors, which are more suited to extracting minutiae. While principal line methods rely on preprocessing and edge detectors for their extraction, subspace approaches such as PCA and LDA are often applied directly to the segmented ROI followed by a classifier.

Wu *et al.* (2002) designed two masks in the form of a vertical first-order derivative and a second-order derivative of palmprint images. Zero crossings of the first-order derivatives identify the edge points and corresponding directions, while the first-order derivatives give the corresponding magnitude. The weighted sum of the local directional magnitude is regarded as an element in the feature vector. These feature vectors are processed with min-max normalization and subsequently matched using the Euclidean distance.

Chen *et al.* (2006a) perform a 2D dual-tree complex transform on the preprocessed palmprints to decompose the images. Dual-tree complex transforms are proposed to allow alignment invariance compared to the traditional wavelet transform. The transformed palmprints are further extracted using a DFT such that subbands are used as feature vectors for the SVM classifier.

Subha and Mariammal (2017) present a comparative analysis of line, scale invariant feature transform (SIFT) and LBP as features for palmprint recognition systems. The line-based method considers principal lines and uses SOBEL edge detection and unspecified morphological operations for feature extraction thereof. The SIFT method searches over all scales and locations of the input and database images by using a difference of Gaussians (DoG) filter to highlight SIFT keypoints. A 3×3 window is used in the simple LBP method. The left and right palmprints are fused at the score level using the weighted sum rule. The Euclidean distance is used for matching the result for all three systems. The results for each method are compared on the IITD-Palmprint dataset and show 75%, 95.65%, and 98.5% accuracy, respectively. Information on the dataset such as sample size, is not provided by the authors.

Zhang *et al.* (2016) also produced a texture-based palmprint recognition system, which uses the difference between adjacent orientation pairs instead of Palmcode. Zhang *et al.* (2003)'s method is used to extract the ROI as described in Section 3.2.3. Six Gabor filters are used to extract features from the ROI to be used for matching based on the Hamming distance between adjacent orientation pairs of the Gabor filters. The system is tested on the CASIA-Palmprint dataset.

Badrinath and Gupta (2011) produced a palmprint recognition system using the Palmcode method – similar to Fingercode, but based on palmprint texture. Palmprint segmentation is performed using Zhang *et al.* (2003)'s method for a square ROI. This system uses the nearest-neighbour approach on the texture within the ROI (Palmcode) for matching.

Morales *et al.* (2011) created a palmprint recognition system that performs matching on orthogonal line ordinal features (OLOF) and SIFT features. The proposed method, which is designed to be robust, uses the IITD-Palmprint to verify the performance. The adapted SIFT algorithm first performs Gaussian blur and Gabor filtering to highlight palmprint keypoints, while the OLOF approach uses the DoG applied thrice to the image at different angles: 0°; 30°; and 60°. The authors note that the min-max normalized matching scores of OLOF and SIFT are uncorrelated and are thus, effective for weighted fusion.

3.5.4 Related Iris Recognition Systems

The system of Wildes (1997) uses the obtained iris segmentation result and considers the pupil and iris boundaries as approximated circles for the localization process. Feature extraction is performed on the ROI using an isotropic bandpass filter, while normalized correlation is applied for the verification matching process. The study concluded that each iris is unique across subjects of widely differing ages (including those of twins and an individual's left and right iris).

Daugman (2004) also used the resulting segmentation for iris verification. This system relies on Iriscode, which uses the same approach as Fingercode, but applies Hamming distance for matching instead of Euclidean distance.

Podder *et al.* (2015) apply radial suppression that uses a non-separable wavelet transform to remove local high frequency information from the iris region to improve the recognition rate. Initial segmentation is performed using IDO. The authors consider eyelashes and eyelids as noisy information, and segment these as background data, while keeping only iris texture. They normalize the ROI size to be the same for all data and surround this region by a black border.

Umer *et al.* (2014) introduced a modified HT technique to reduce the original search space resulting in both faster execution time and improved accuracy. The technique is called restricted circular Hough transformation and can search specifically for circles bounded by the upper and lower eyelids. The algorithm achieves this by randomly selecting an initial position centre point and distributing all other points symmetrically around it at an iterated distance in search of the outer edge of the iris. Textures within two concentric circles are extracted, constituting the feature vectors of the iris. The individual feature vectors of each eye are combined into a larger vector, which is classified with a kernel SVM using a radial basis function with unspecified parameters. The dataset used by the authors is called IITD-Iris. They also tested their algorithm on three other datasets, achieving near-perfect verification rates.

Vishi and Yayilgan (2013) do not provide information on their particular biometric recognition algorithm, but state that VeriEye software was used to produce scores for the SDUMLA dataset.

3.5.5 Related Fusion Systems

Yao *et al.* (2007) proposed an accurate multimodal biometric system that fused the face and palmprint at feature level. Four Eigen-based face and palmprint feature-fusion algorithms were compared. The proposed method filters the EigenFaces and EigenPalms with a Gabor filter followed by weighted concatenation of the resulting feature vectors. The AR Face dataset was organized as a component to a fused dataset, with a resolution of 60×60, and limited to the first 20 images per 119 individuals. Images in the palmprint

dataset, provided by Hong Kong Polytechnic University, were matched to those used in the face dataset with the same number of samples per individuals. The proposed fusion method obtained its highest accuracy of 95% with six training samples, but achieved 91% with only one training sample.

Ahmad *et al.* (2010) proposed a similar system tested on ORL face and PolyU palmprint datasets. The resulting accuracy was a 99.5% GAR, with an unspecified FAR. The near-perfect accuracy is attributed to the use of the relatively small-scale ORL face dataset, limiting the fused dataset to 40 individuals and 10 samples in total per individual.

Rattani *et al.* (2011) created a multimodal biometric system by fusing face, fingerprint and palmprint images. They showed that good dimension reduction and normalization algorithms allow feature-level fusion to produce a lower EER than score-level fusion. Their feature-level fusion approach emphasises the selection of image regions of both modalities. This study concluded that fusing information from uncorrelated traits such as the face, fingerprint and palmprint at the feature level increases human identification accuracy. Significantly better FAR and FRR (and ERR) were achieved at the feature level compared to the score level, but experiments were performed on datasets that are not publicly available.

Other face, fingerprint and palmprint studies include (Karki and Selvi, 2013) and (Sharma and Kaur, 2013), both of which use a Curvelet transform followed by SVM classification. Unfortunately, both of these studies were tested using datasets that are not publicly available.

Paul and Gavrilova (2014) proposed a biometric system that performs feature-level fusion. They focus on producing an effective cancelable biometric system for template protection, while sustaining high accuracy.

A resolution of 75×50 is used on a pseudo multimodal dataset consisting of 858 images. The pseudo multimodal dataset consists of three face datasets and two ear datasets, respectively, that were obtained under different conditions. A nearest-neighbour classifier is used directly on the texture features with $k = 7$.

At 0% FAR on the fused dataset, the authors' proposed system achieves a verification accuracy of 96% compared with the 93% when using LDA feature extraction with Euclidean distance matching.

A decision-level biometric fusion system by Paul *et al.* (2014) tested the same pseudo multimodal dataset with the aim of overcoming the issue of varying (image) data quality. They show that fusing data at the score level, and including a final classification result at the decision level helps mitigate the varying data quality issue. The decision level uses the Fisher method, which is discussed in Section 3.4.

Eskandari and Toygar (2015) designed a robust fusion scheme for the face and iris at the feature level. They emphasise that feature alignment plays a pivotal role during the preprocessing step. A comprehensive set of experiments show the effect alignment has on recognition accuracy relative to the biometric data. The face is segmented and aligned using the detected eye position as reference. The segmented iris is rotated for alignment based on texture patterns. Furthermore, this system is made robust to noise by applying a backtracking search algorithm. The CASIA-Iris-Distance dataset, containing close-up near-infrared face images, was used for testing the importance of feature alignment.

The first 90 subjects and 10 random samples (50:50 split) were selected from the CASIA-Iris-Distance dataset, which consists of images of both the face and iris. The iris benefitted significantly from alignment with a 44% improvement to verification accuracy. Aligning the face improved verification accuracy by 7%. The face, both irises and face-iris fusion verification results were 87.33%, 71.55% and 94.44%, respectively at 0.1% FAR. Multimodal biometric fusion at the feature level achieves similar accuracies from different authors in the literature and they often use subspace algorithms such as Eigen, Fisher or similar frequency domain approaches, where they rely solely on reducing features in the subspace, instead of using feature extractors such as LoG, Gabor or LBP-based methods that can be used in the spatial domain for additional feature reduction purposes.

However, the more fundamental concerns of multimodal biometric studies at the feature level are the following:

1. Availability of multimodal biometric datasets is relatively low, resulting in most experiments being carried out on pseudo versions of multimodal datasets.

2. Private unimodal datasets are used extensively in the literature limiting comparisons to other systems in some way.

3. Many multimodal feature-fusion systems perform human verification only or perform identification on a small subset of a large dataset.

4. The lack of EER, FAR and FRR metrics in most related studies limits a direct comparison to only system accuracy.

Many other fusion studies are available that clearly explain their method, testing procedure and results, but not at the feature level. Tran and Le (2015) proposed a biometric system that performs score-level fusion on the ORL face and FVC2004 fingerprint datasets, which are publicly available. They use a texture-based approach with an SVM classifier.

Zernike Moment features are extracted by using the centre of the face and the core of the fingerprint as respective reference points. The extracted features per modality are separately trained using an SVM. Probability estimates produced by the SVM are used as matching score values for the face and fingerprint. The simple sum rule is used to fuse the face and fingerprint scores obtained from the SVM.

At 0.01% FAR, verification accuracies of 92.71%, 94.48% and 98.82% are reported for the face, fingerprint and fused dataset respectively. The high accuracy is again attributed to the use of the relatively small-scale ORL face dataset, and the authors limited the fused dataset to 40 individuals and 8 samples in total per individual, similar to the study by Ahmad *et al.* (2010). The authors also show that the relevance vector machine and SVM achieve similar accuracies on those datasets.

The following two studies use the publicly available SDUMLA-HMT multimodal biometric database and record their results as follows.

Vishi and Yayilgan (2013) use images from two modalities in SDUMLA-HMT to perform verification on the fingerprint and iris. The individual scores acquired from these modalities are first normalized using TanH, and subsequently fused at the score level using Maximum Score. Unfortunately, further details on their approach are not revealed, but this study is included due to the use of a multimodal dataset (SDUMLA-HMT) of interest. Their test procedure used 100 individuals and the first five samples, but the training and test sample split is not provided. The fingerprint images are selected from one out of the four sensors – FPR620 – which they state has the best quality and performance, while FT-2BU achieved the worst performance, but these explicit results are not given. Their fingerprint, iris and fused "accuracy values" are given as GAR values of 1.01%, 3.30% and 0.004%, respectively.

Khellat-Kihel *et al.* (2016) combine the fingerprint, fingervein and fingerknuckle using feature-level fusion by concatenation resulting in 633 components. A 2D Gabor filter is used for feature extraction but with unspecified parameters. They enhance the result by further reducing dimensionality using kernel Fisher analysis (KFA) resulting in only 105 components. KFA is an extension of LDA that allows for non-linear mappings where the dot product is replaced by a higher-dimensional kernel function. The fingerprint and fingervein datasets are obtained from the SDUMLA-HMT multimodal database with 106 subjects. Verification results are calculated using the atypical cosine distance[4] instead of Euclidean or Mahalanobis. The research shows promising results when using KFA for feature reduction. However, the training and test sample ratio is not provided nor are the sensors specified. This study shows that feature level fusion can result in worse accuracy as denoted by the higher EERs for the PCA and LDA approaches as shown in Table 3.2.

[4]Similarity measure is calculated as the smallest difference between the cosine angles of two vectors.

Table 3.2: Comparing the verification performance of KFA, PCA and LDA on individual and fused modalities [taken from (Khellat-Kihel *et al.*, 2016)].

Approach	**Modality**	EER(%)
KFA	Fingerprint	13.6
KFA	Fingervein	8.4
KFA	Fingerknuckle	8.4
PCA	Fused	35.6
LDA	Fused	38.0
KFA	Fused	0.04

3.6 Summary

This chapter presented the algorithms used in implementing fingerprint, face, palmprint and iris biometric systems. Related studies particular to each biometric modality were discussed, including some using specific datasets that are later compared with the results of the proposed biometric systems implemented by combining the presented algorithms in a variety of ways. The next chapter gives an overview of the proposed methodology for implementing these algorithms to create a variety of biometric systems and sets out the experimental conditions.

Chapter 4

Experimental Setup

Before attempting to create new multimodal biometric systems, it was necessary to ascertain the best methods for implementing highly accurate, unimodal biometric systems. Hence, much experimentation was done on each of the four modalities chosen for this research. The next four chapters document the experimental work on unimodal biometric systems using fingerprint, face, palmprint and iris images, respectively. Since only image-based modalities are used in this research, commonalities observed during the creation of the unimodal biometric systems were used to assist in finding optimal features that could be combined in the fusion experiments discussed in Chapter 9.

This chapter describes the approach used in all experiments on both unimodal and multimodal biometric systems. First, in Section 4.1 an overview of the generalized approach, relevant to all modalities, is given. The structure of the datasets used for open-set identification for the four modalities is explained in Sections 4.3 to 4.6, while the evaluation procedure, common to all biometric systems, is discussed in Section 4.7. Finally, Section 4.8 details the hardware and software used in implementing the various test biometric systems.

4.1 Overview of the Experimental Approach

The biometric systems tested were each created using the generalized approach presented in this section and illustrated in Figure 4.1. A variety of algorithms were used for the experimental fingerprint, face, palmprint and iris biometric systems. Algorithms common to all modalities are mentioned in this chapter, while those specific to a particular modality are explained in the relevant chapter.

For each biometric system (created per modality), training and test images are processed in a similar way from a high level perspective as described in Section 4.1.2. A detailed approach specific to each modality is documented in the relevant chapters.

4.1.1 Preprocessing

The proposed systems begin processing by converting the colour space of all images to the greyscale range 0 – 255. As discussed in Section 3.4, well-aligned texture and consistent illumination levels of a biometric image are key factors in minimizing intra-class variations. However, to obtain accurate keypoints that are required in the texture alignment process, consistent illumination levels are important. The standard pixel normalization method, explained in Section 3.1.1, is used to normalize lighting as it has few to no known side effects. HE and CLAHE are known to be more effective for extreme cases but can cause side effects, and are thus explored per modality in the relevant chapters.

4.1.2 Modality Specific Algorithms

This is a generic section that describes the computer vision techniques used to produce a well-aligned ROI for a particular modality. The "specific enhancement" step refers to lighting normalization and enhancement algorithms that improve keypoint detection when applied to that particular modality. The resulting keypoints are used as anchoring points for cropping, with respect to the images of a particular modality, to obtain appropriately

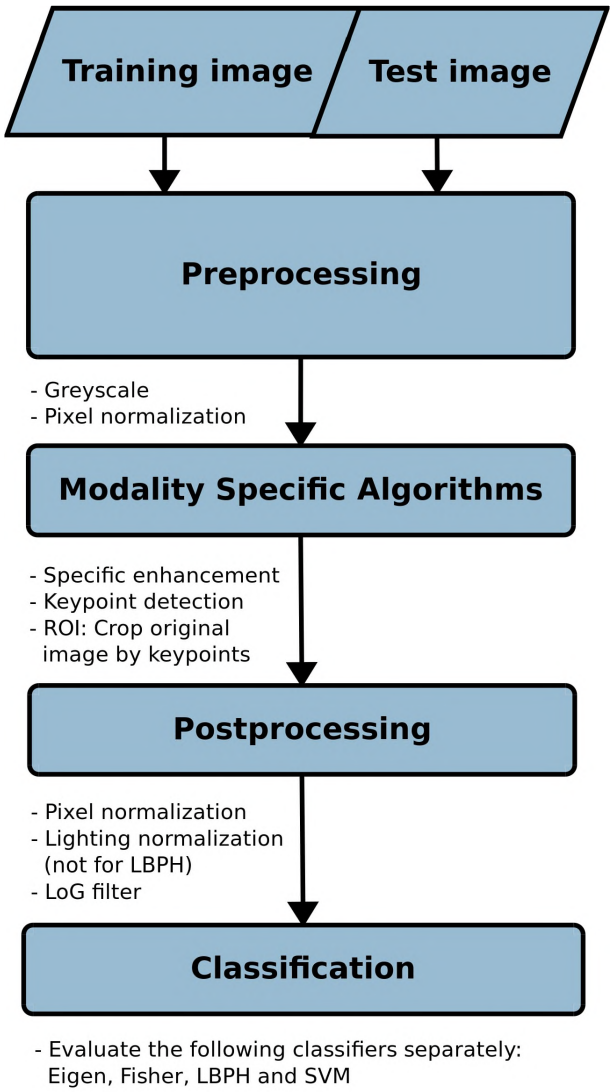


Figure 4.1: Overview of generalized experimental approach.

segmented images, that is their ROIs. However, cropping is performed on the original images instead of the preprocessed images, resulting in an ROI that retains the original lighting. The reason for this is explained in the next section. Additionally, the detailed process per modality is discussed with respect to Chapters 5 to 8.

4.1.3 Postprocessing

During this stage, feature extraction takes place on the ROIs produced by respective modality specific algorithms. However, the lighting normalization methods used during preprocessing could also be applied to the segmented images (containing original lighting), since histogram measures are more localized at this processing step. The second application of pixel normalization is thus more effective on the features within the ROI as it excludes background noise that interferes with the process. However, normalizing images with larger fluctuations of lighting in terms of intra-class samples can introduce noise when using histogram-based methods such as HE and CLAHE as discussed in Section 3.1.2.

As was established in Sections 3.1.5 and 3.1.6, the circular LBP methods, ELBP and LBPV, are effective at characterizing lighting, but are texture feature extractors. Since these features include some sporadic textures resulting from bilinear interpolation, sparse versions of the ELBP and LBPV operators are thus proposed in this research to act as alternatives to traditional lighting normalization algorithms. Furthermore, the LBPV operator is modified by using the result of the original image subtracted from the LBPV processed image. This is theorized to further enhance lighting normalization by mitigating variances in neighbourhood contrast, and is henceforth known as ELBPV. These operators are hence explored in the following experimental chapters as an alternative to histogram-based lighting normalization approaches. The parameters used and the comparison with other methods are shown in Section 5.3.2. It is to be noted that these methods (including LBPV) are not applied to LBPH during classification as they invoke similar operations already performed by LBPH.

It is theorized that using modified ELBP and ELBPV operators for lighting normalization can mitigate the unwanted effects of the LoG filter that are described in Section 3.3.2,

and thus a LoG filter that favours bandpass frequencies, is applied to the result of these modified operators. The LoG parameters were tuned, during the validation section of the modality specific algorithms, starting at a base kernel size of 3×3 and using an increasing increment of three up till 19×19. Large Gaussian and smaller Laplacian kernels gave favourable results akin to well-tuned Gabor filters and were also more consistent and had faster processing speeds. The resulting combinations of ELBP and ELBPV with the LoG filter tend to reduce the number of redundant features in image space. The synergy of these resulting combinations are thus promising for efficient biometric fusion. To the best of our knowledge, no existing biometric system performs lighting normalization and feature reduction in this way.

The suffix HE, L, LBPL or LBPVL is added to the classifier name to specify HE, LoG, or combinations of ELBP and ELBPV as a postprocessing method, respectively. LBPVL is designed to make modalities as uncorrelated as possible before fusion, without jeopardizing the integrity of features of standalone modalities.

The width and height of image ROIs were resized to their nearest 50^{th} pixel, with some exceptions, to minimize distortion and to simplify the resize process when fusing and subsequently classifying. This simplification refers to the compatibility across feature vectors when fusing modalities using either the feature vector concatenation method or feature vector averaging method, as discussed in Chapter 9. The resulting postprocessed images are outputted to a text file as 1D vectors for classification.

4.1.4 Classification

Eigen, Fisher, LBPH and linear SVM are among several methods that can solve pattern recognition problems. These four methods all solve linear problems and are well suited to characterizing texture features as discussed in Section 3.4. The purpose of this research was to investigate the effects of biometric fusion, and not to optimize at the classification stage. Therefore, additional classifiers such as logistic regression, deep learning and several others were considered to be beyond the scope of this research.

The Eigen, Fisher, LBPH and linear SVM methods are applied using the training and test images acquired after postprocessing. These classifiers are thus used on the resized and postprocessed segmented texture features in all experiments for comparison. Parameter tuning of all the classifiers are performed during the validation section of the modality specific algorithms. The classifiers use one or more training data samples to predict the correct identity of a test sample (or reject an imposter for open-set identification). These training samples are separated from the test samples thereby ensuring that test data is unseen. As it is impractical to fit the large number of combinations of classifiers and features in the experimental chapters, comparisons are often limited to the best performers with unexpected results of alternative combinations also mentioned. All parameters including resolution are unchanged per dataset unless otherwise mentioned.

The Eigen, Fisher and LBPH methods, originally designed for face recognition, are expected to work comparably on other biometrics. The LBPH method does not require retraining and is specifically designed for texture-based matching. All three methods use the nearest-neighbour distance to make a decision on the correct class when no threshold is applied. When a threshold is applied (open-set identification, where FPIR $\neq 100$), and the nearest-neighbour distance falls above this requirement, the input image is rejected as an imposter.

The linear SVM, popular as a general classifier, is used for its scalability over kernel-based versions and since it removes more data points that do not adhere to the maximum margin without requiring substantial parameter tuning (Tang, 2013). Furthermore, kernel-based methods prefer lower dimensional data than that used in the test systems. Even the linear SVM requires relatively little training data compared to the other three classifiers and thus is of interest to see the interactions with texture-based features used in the proposed approach. This linear SVM uses the one-versus-all method to handle the multi-class problems of identification and probability estimates to reject imposters.

The SVM uses a different measure to accept/reject a class. A logistic sigmoid is used to convert the deterministic decision weights into probability estimates using the formula of Platt (1999):

$$P(y|X) = 1/(1 + \exp(S^{+} * f(X) + S^{-})) \tag{4.1}$$

Overfitting the model is prevented by using an internal five-fold cross validation. The disadvantage of using the SVM to reject imposters in this way is the increased computational complexity compared to not using probability to make the final decision. Finally, multi-class support is enabled for identification by using the one-versus-rest method proposed by Hsu and Lin (2002). The Liblinear library allows multi-threading when using the above method.

4.2 Organized Open-set Identification Classes

The datasets together with their corresponding acquisition techniques are described in the sections that follow. Organizing data for open-set identification is generally the most challenging. The following sections discuss the structure of the datasets used to measure open-set identification performance.

For the purpose of this research, verification is considered to perform a 1-to-1 match with a single training sample unless stated otherwise. Imposters in verification tests are regarded as all classes minus the class to be verified – the standard method used in biometric literature. Training and test data are classed differently for identification according to each dataset and are described in the following sections where information is available. Unless otherwise stated the classes and corresponding image samples are sorted in alphanumerical order with no other changes to the datasets. Non-class data samples are not taken from different datasets as to avoid acquiring them under vastly different conditions, which can be differentiated by the algorithm. Instead, the experiments are made challenging by splitting the datasets into classes and non-classes such that both are acquired under near-identical conditions to reduce discrimination.

The total number (t) of N samples are arranged into d training samples for generating the database model, and p test samples, of which each class is predicted. The number of

training samples used is denoted by prefixing the number before the classifier name for e.g. "1-Eigen" denotes one training sample used with the Eigen classifier. Due to the varying and often limited number of samples per individual, the test samples are calculated as follows unless noted otherwise:

$$N(p) = \sum_{i} N(t)_i - N(d)_i, \tag{4.2}$$

where i is the sum of classes and non-classes. The individual class to non-class ratios are given at the end of each dataset description in the sections that follow.

4.3 Fingerprint Datasets

4.3.1 MCYT

The entire fingerprint capturing process was performed under supervision of an operator (Ortega-Garcia *et al.*, 2003). The two different fingerprint sensors used in this process were:

- CMOS-based capacitive capture device, model 100SC by Precise Biometrics with a resolution of 500 DPI (300×300). The lighting was less consistent on the data collected with this sensor compared with the optical sensor.
- Optical capture device, model UareU by Digital Persona with a resolution of 500 DPI (256×400).

A total of 12 samples of each of 10 fingerprints per 100 individuals were acquired with these two devices, resulting in 240 images per individual. Variability in fingerprint positioning on the sensor was included to simulate real-world conditions. The dataset captured with the optical sensor is used for open-set identification. The 12 samples per fingerprint were grouped by the authors as follows (Figure 4.2):

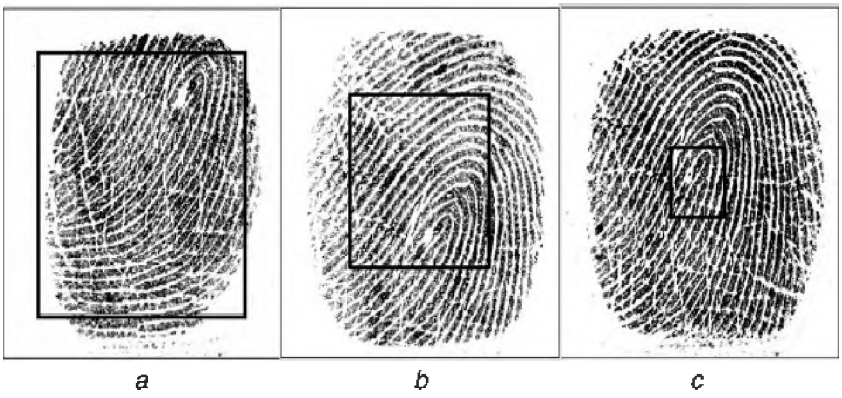


Figure 4.2: Low, medium and high levels of control in fingerprint positioning/constraints.

(a) Low level of control – three samples were arbitrarily positioned without looking at the sensor, but with the operator ensuring that at least one core/delta fell within the image region before capture.

(b) Medium level of control – three samples were captured while positioned within a designated rectangle on the sensor shown to the user on the screen.

(c) High level of control – the remaining six samples were captured similarly to the medium level, but confined to a smaller rectangle on the screen.

Prior to experimentation, class and non-class (imposter) data were distinguished as discussed below.

Class data

All fingerprints except from fingers 4 and 9 were used, resulting in 800 classes. The order of samples for each individual was reversed such that the first six samples (fingers 0–5) were of the high control group. Twelve data samples were shared between training and testing.

Non-class data

Fingerprints from fingers 4 and 9 were not used for training, and hence were regarded as the non-class set of 200.

4.3.2 FVC2004

FVC2004 is a popular challenging collection of fingerprint datasets because deliberate noise was introduced such as wet fingers and exaggerated distortion thereby dramatically increasing intra-class variation. The dataset acquired by the CrossMatch V300 optical sensor, known as DB1, comprises eight samples for each of 100 different fingers resulting in 800 fingerprints in total (Maltoni *et al.*, 2009). This is known to be the most challenging dataset as it generally suffers significantly higher EERs. These 100 different fingers were chosen from four different fingers of 30 subjects and made publicly available. These challenges resulted in groups of fingerprints distinguished as follows:

(a) Atypical fingerprint placement at different vertical positions

(b) Low or high pressure applied against the sensor

(c) Exaggerated skin distortion and rotation by combining (a) and (b)

(d) Dry or wet fingers

Furthermore, a minimum quality was not enforced and the sensors were not cleaned between sessions. Lastly, the acquisition of different fingers was interleaved in an effort to maximize placement differences of intra-class fingerprints.

Class data

The DB1 dataset of fingerprints obtained from the index finger were set into 50 classes with eight data samples shared between training and testing.

Non-class data

The fingerprint impressions of the remaining 50 classes were used as non-class sets. The genuine-imposter ratio is thus 50:50.

4.3.3 SDUMLA Fingerprint

The fingerprint data are taken from the multimodal dataset known as SDUMLA-HMT (Yin *et al.*, 2011), which is referred to as a true multimodal dataset because the same 106 individuals are used to capture images for each modality.

The fingerprint database contains fixed 152×200 resolution images captured eight times per thumb, index and middle finger of both hands, for 106 individuals using the 500 DPI FT-2BU capacitive fingerprint scanner by Miaxis Biometrics. The individuals were 61 males and 45 females aged between 17 and 31. This dataset contains the most partials and varies from good to bad quality, but are not grouped accordingly. Furthermore, the left and right hands are not specifically labelled.

Class data

Fingerprints obtained from the left index finger were set into 106 classes.

Non-class data

The fingerprint impressions of the left middle fingers were used as non-class sets. The class to non-class ratio is thus 50:50.

4.4 Face Datasets

4.4.1 Feret

The FERET b-series (Phillips *et al.*, 1998) dataset is a subset of FERET face that consists of 200 individuals with samples arranged as eight pose angles ranging from $-60°$ (subject's left) to $+60°$ (subject's right), and three frontal poses at approximately $0°$. The FERET-b subset was used because the original dataset had a varying number of samples, often limited to only one or two per person, which is not sufficient for identification. This subset was used for validation only.

Class data

Eleven samples of the first 100 individuals described above were arranged for each of the 100 classes such that the frontal poses were first, followed by $-60°$ to $+60°$.

Non-class data

Images from the second 100 individuals were organized identically to the class data.

4.4.2 Fei

The FEI face dataset (Thomaz and Giraldi, 2010), consisting of 14 samples for each of 200 individuals, was used in the experimentation. Details about this dataset were acquired from file metadata. The images were captured at a resolution of 640×480, f/4.5 aperture, 18.7 focal length and using a Sony DSC-W1. The first ten samples were captured at $90°$, $75°$, $60°$, $45°$ and $15°$ in both horizontal directions. The remaining four frontal samples contained varied facial expressions and in addition the last sample for each individual was significantly darker and not fully visible without brightness compensation. The data were split equally such that the first 100 individuals were male and the second 100 individuals were female.

Class data

The class data comprised the first 50 males and first 50 females. The order of the samples was retained.

Non-class data

Images of the remaining 50 male and female individuals were used as non-class sets.

4.4.3 PUT Face

The PUT face dataset (Kasinski *et al.*, 2008), consisting of 100 individuals, was also used in the experimentation. The images were captured at a relatively high resolution of 2048×1536 fine quality using a Fujifilm FinePix 6900 Zoom camera. Each sample was taken at a different pose angle, ranging up to 60°, in all directions permissible by the neck, starting with a left pose through the neutral position to a right pose followed by up and down and so forth. In other words, ten transitions horizontally, vertically and diagonally and five duplicates in the neutral position were obtained, resulting in 85 samples per individual.

Class data

The data were classed to contain images of the first 50 individuals. The order of the samples was retained.

Non-class data

Images of the remaining 50 individuals were used as non-class sets.

4.4.4 SDUMLA Face

This dataset was acquired from the faces of the same 106 individuals used in the SDUMLA fingerprint dataset (see Section 4.3.3). The samples covered 21 pose variations, four facial expressions, two illumination levels, and two accessories. The subject was 50 cm from the camera. The pose variations were omitted and the following eight samples were used in order:

(a) Two accessories – glasses and hat

(b) Four varied expressions – smile, frown, surprise, and closed eyes

(c) Two varied illumination levels – one bright illumination level using a lamp and one without the lamp

Class data

The eight samples, in the order described above, of the first 31 males and 23 females were set into 54 classes. In other words, the first sample of each class shows the subject wearing glasses.

Non-class data

The rest of the individuals consisting of 30 males and 22 females were used as non-class data.

4.5 Palmprint Datasets

The devices used in creating the palmprint datasets described in this section, captured the entire hand.

4.5.1 CASIA-Palmprint

For this palmprint dataset, eight samples per hand of 312 individuals at a resolution of 480×640 were captured with an in-house sensor device that required placement of the hand inside the device without the use of pegs to constrain finger positions as seen in Figure 4.3 (Zhenan *et al.*, 2016). The inside of the device contains a uniform background and evenly distributes illumination while capturing with a CMOS sensor, resulting in 8-bit gray-level JPEG images.

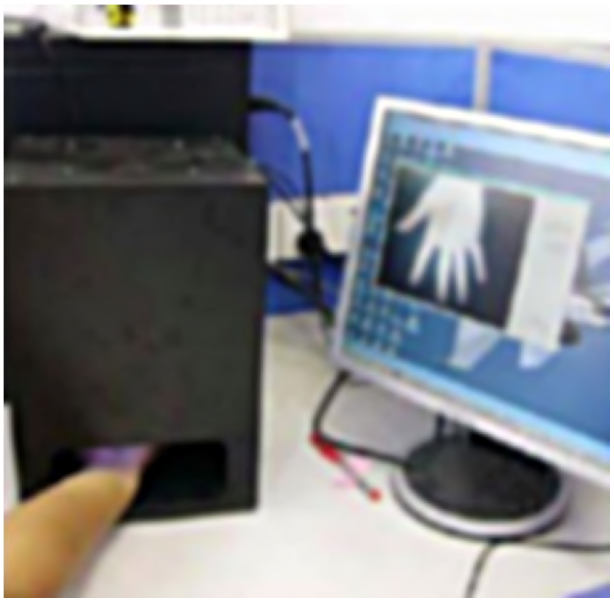


Figure 4.3: Enclosure of the CASIA-Palmprint in-house sensor device [taken from (Zhenan *et al.*, 2016)].

Class data

Left palmprints were placed into classes.

Non-class data

Right palmprints were mirrored and similarly placed into non-class sets.

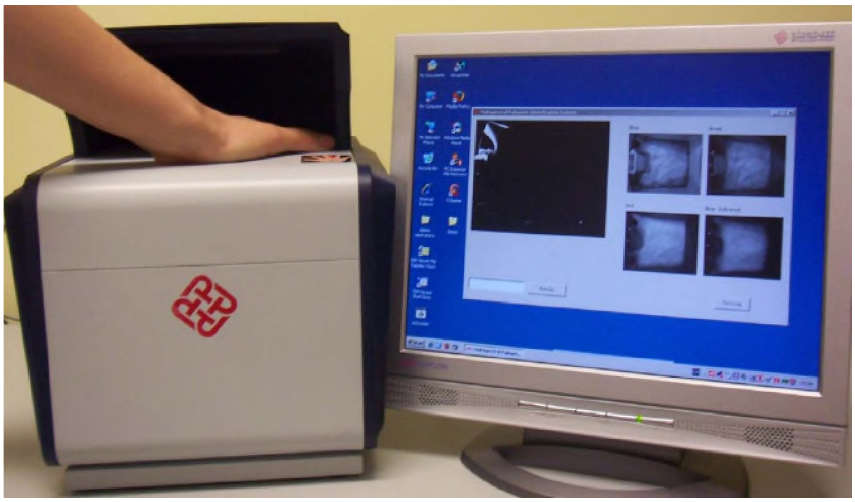


Figure 4.4: Device used to capture hand images for the PolyU dataset [taken from (Zhang *et al.*, 2010)].

4.5.2 PolyU Palm

Palmprint images were collected from 250 volunteers, including 195 males and 55 females with an age distribution between 20 to 60 years (Zhang *et al.*, 2010). Six images for each hand were collected in each of two separate sessions, at a resolution of 352×288 and four colour spectrums using the device shown in Figure 4.4. The volunteers were asked to place their palm on the platform with their fingers constrained by pegs.

Class data

A total of 12 samples per left palmprint of the red colour spectrum were placed into 250 classes.

Non-class data

The corresponding 12 samples of the right palmprints were mirrored and similarly placed into non-class sets.

4.5.3 IITD-Palmprint

The IITD Delhi palmprint image dataset (Kumar, 2008) used a simple touchless setup to create a challenging palmprint dataset for biometric systems to recognize, but simulated conditions allowing for easy image acquisition. Seven images were acquired per hand from 235 individuals in an indoor setting using a circular fluorescent illumination around the camera lens at a resolution of 800 × 600 pixels in bitmap format. Since the touchless acquisition method only requires the hand to be within the camera's viewfinder, thus allowing significant tolerance, intra-class variation is increased due to the addition of random hand pose variations and creases.

Class data

Left palmprints were placed into 235 classes.

Non-class data

Right palmprints were mirrored and similarly placed into non-class sets.

4.6 Iris Datasets

The devices used in the iris datasets captured the whole periocular region.

4.6.1 CASIA-Iris-Lamp

The CASIA-Iris-Lamp dataset (Tan, 2016) is a challenging iris dataset, captured using a non-invasive OKI IRISPASS-h handheld sensor with a light alternating between on and off while capturing the samples. This caused non-linear deformation of the iris due to the expansion and contraction of the pupil. A total of 20 samples per individual were captured.

Class data

The left periocular images of 411 individuals were used for class data.

Non-class data

The non-class samples were similarly gathered from the right periocular images.

4.6.2 IITD Iris

The IIT Delhi Iris database consists of periocular images with a resolution of 320×240, captured using a JPC1000 Iris Recognition Camera sensor held at arm's length (Kumar and Passi, 2010). In total, five samples per iris from 224 individuals, 176 males and 48 females, were captured.

Class data

The left periocular images of the 224 individuals were used as class data.

Non-class data

The non-class samples were similarly gathered from the right periocular images.

4.6.3 SDUMLA Iris

The iris dataset for this multimodal database (same individuals as SDUMLA fingerprint and face) was acquired with an in-house NIR sensor with resolution 768×576 resolution and stored in bitmap format. Although reflection was avoided by removing spectacles, eyelash occlusion was a significant factor in the majority of the images, making this a challenging dataset. The low number of samples limits those available for training.

Class data

The left periocular images of 106 individuals were used for class data.

Non-class data

The non-class samples were similarly gathered from the right periocular images.

4.7 Evaluating Closed-set and Open-set Identification

The metrics discussed in Section 2.6.2 were used to evaluate the performance of each of the unimodal and bimodal biometric systems that were created. Closed-set identification accuracies are generally given for validations tests as they do not have impostors. On the other hand, the more comprehensive experiments are on open-set identification and are displayed in ROC curves with DIR on the y-axis and FPIR on the x-axis. However, an indication of the closed-set identification results (IR) can still be read from these graphs by observing the DIR at FPIR = 100%. The earlier a curve reaches its peak the better the FPIR and the worse the FNIR per peak DIR. As the FPIR increases, the DIR yields diminishing returns, thus FPIR values on ROC curves are not according to scale to emphasize differences in the performance of classifiers. Furthermore, instead of providing a value for the area under each ROC curve, this overall accuracy measure (across FPIR values) together with individual DIR values for specific FPIR values, are stated where necessary.

The FNIR and FPIR can be combined for open-set identification to fit it into a single graph, but to accurately observe the effects of impostors, the FPIR and FNIR are explicitly stated. When no threshold is used, FNIR = 100 − DIR, which is the same as closed-set identification, but at a 100% FPIR. Both the FNIR and FPIR are affected by the threshold value for biasing true and false matches resulting in some classes mistakenly

being rejected as impostors. Therefore, the FNIR cannot reliably be based on the DIR. The significance of separating the metrics lies in real-world applications such as requiring strong bias against allowing low-level security clearance individuals access to higher security due to misclassification. The FNIR results are thus included in a table by specifying the corresponding FPIR value following the closed-set identification results. Finally, the number of individuals that are falsely rejected as impostors (essentially FRR) can simply be calculated as $100 - (\text{DIR} + \text{FNIR})$. This enables the comparison of the four classifiers in one graph, while providing both FNIR and FPIR values.

4.8 Hardware and Software

All experiments were run on an Intel i7 2700k 4.4 GHz quad-core CPU with 8 GB of DDR3 2133MHz RAM, running the Peppermint 8 Linux operating system. The software tools used to implement the proposed systems for all modalities are:

- C++ for programming logic
- OpenCV C++ computer vision library for general image processing and for Eigen, Fisher and LBPH classification
- Dlib library for additional image processing and machine learning of face segmentation algorithms
- Liblinear machine learning library for SVM classification
- Python for organizing the training and test scripts
- Bash for batching (in parallel where applicable) the tools mentioned above to cover all parameters (only for validation) and feature-classifier combinations during experimentation

4.9 Summary

This chapter discussed the methodology used for creating a variety of test fingerprint, face, palmprint and iris recognition systems, including the datasets and algorithms used, and the evaluation methods. The datasets for open-set identification were organized where possible such that the sequence of samples and number of class and non-class data were made known to establish their possible real-world setting.

Open-set identification biometric system types have several applications and to extract maximum information about the effects of various features and combinations, the evaluation procedure includes the metrics DIR, FPIR and FNIR separately in the form of a ROC curve derived from a table of results. The values for the area under each ROC curve are not explicitly given during experimentation. However, results of individual metrics are stated where necessary to provide a more relevant analysis. For the applicable datasets, the factors affecting image quality were discussed in Section 4.2. However, visual inspection was carried out during experimentation as an indicator of image quality due to differing standards per biometric modality.

The general approach followed in implementing the test biometric systems was outlined and details were provided of common algorithms used for all modalities. The tuning of optimal classifier parameters and the results of classifier and feature combinations for each of the biometric systems are compared in the subsequent chapters.

Chapter 5

Fingerprint Recognition Algorithms and Results

This chapter discusses the algorithms used to detect and align keypoints of a fingerprint in order to segment the fingerprint region, given an image captured using different sensors and backgrounds. The fingerprint recognition process begins by presegmenting (silhouette detection) the image around the fingerprint silhouette. The alignment proceeds by finding the core point within the presegmented image. The final segmented fingerprint is centred at the core position by cropping approximately 50% of its silhouette. The segmented image is further processed using the techniques described in Section 4.1.3, before classification of the results takes place. The first set of experiments validate the methodology and tune the parameters. Thereafter, open-set identification is evaluated on three experimental fingerprint datasets. This process is discussed in detail in this chapter together with the relevant results.

5.1 Aligning Fingerprint Features

The fingerprint alignment process is initiated by finding the core point and using it to allocate an ROI that minimizes the discrepancy of stretch and alignment differences among

(a) Original image **(b)** NL-means image **(c)** Unsharp image **(d)** Binary image **(e)** Presegmented image

Figure 5.1: Ridge enhancement and presegmentation before Poincaré reference point detection.

intra-class fingerprints.

5.1.1 Core Detection using Enhanced Ridges

The Poincaré algorithm (Jain *et al.*, 2000b) works well on good quality fingerprint images, but fails to correctly localize reference points in poor quality fingerprints such as partials or fingerprints with poor ridge and valley contrast. Thus, contrast must be enhanced, while removing noise at the same time. This is done using an NL-means filter at a 7×7 kernel and 21×21 neighbourhood size. An unsharp mask is applied by subtracting the NL-means result from a 11×11 Gaussian filter using a 3:2 weighting in favour of a sharpened image. This produces an image with high ridge-valley contrast, which allows for reliable presegmentation by searching for the outer white pixels in every row of a binary image to produce the result shown in Figure 5.1e. Poincaré is applied to the presegmented image in 16×16 neighbourhoods.

The core point is detected based on the fingerprint type as discussed in Chapter 3 and illustrated in Figure 3.4. The difference in angle between the singular point and core of the input image and database image is calculated and used for a shear transformation of the input image. In the case of more than one singular point or core, the one with the

smallest y coordinate is used.

5.1.2 Minutiae Detection for Texture Alignment

The modified Zhang and Suen (1984) ridge thinning algorithm described in Section 3.2.1 is used to detect minutiae – ridge endings and bifurcations – in this section. Differentiating between ridge endings and bifurcations provides useful information for improving minutiae-based matching, but is unreliable unless high quality input data are acquired. The minutiae extraction accuracy depends on the quality of the input image and the reliability of each processing step. However, this is a sequential process and thus encourages the propagation of errors. This causes some genuine minutiae to be missed and spurious minutiae to be detected as genuine. While false minutiae are reduced by considering ridge endings and bifurcations as a single type, this does not take advantage of the extra information. The use of a pure minutiae-based method lowers processing time and storage requirements. However, with modern hardware, the proposed method uses texture patterns as key features and the special minutiae, called singular points including the core, primarily for alignment and segmentation.

LBPH is used with a sliding window, searching from the centre towards the image borders, to find the best match between the input and database image texture – when lowest confidence value is returned. Assuming the database image core point is centred, the closer the detected core point is, the fewer iterations are required by the sliding window. The arch type often requires more iterations for alignment due to a lack of singular points. Partial fingerprints are considerably less trivial to segment and are dealt with using the LBPH sliding window. An example partial fingerprint is shown in Figure 5.5a. The core point, depicted as a dot and minutiae/singular points depicted as circles, are illustrated in Figure 5.2.

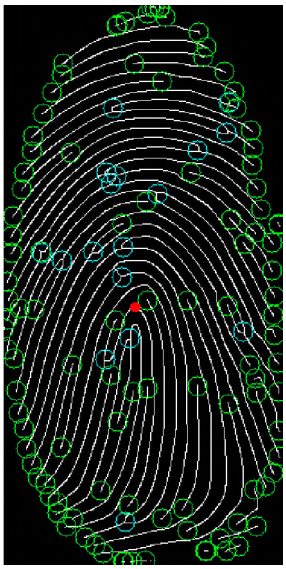


Figure 5.2: Ridge-thinned fingerprint image.

5.2 Fingerprint Segmentation

Although, detecting the boundaries based on the capturing resolution and the surface area of the sensor can also serve as a helping mechanism, the presegmented image was found to always contain only the fingerprint silhouette in validation tests. The final segmented images are cropped versions of the original images based on the anchor points obtained in Section 5.1.1 and 5.1.2. Furthermore, three different square ROI sizes of 75%, 50% and 25% are used by cropping by each percentage with respect to the outer region on the x-axis of the aligned original fingerprint image as shown in Figure 5.3. However, since the outer region can have minor variations across intra-class samples, partial fingerprints are detected based on the difference between the database and input image outer ridge boundaries being within a 50% threshold. These ROI sizes are normalized by rounding down to the nearest 50 pixels. The LBPH sliding window method is used to align and scale the ROI of an input image according to a partial database image. However, if the database image is a partial, the image is only aligned and not scaled to preserve information for test samples.

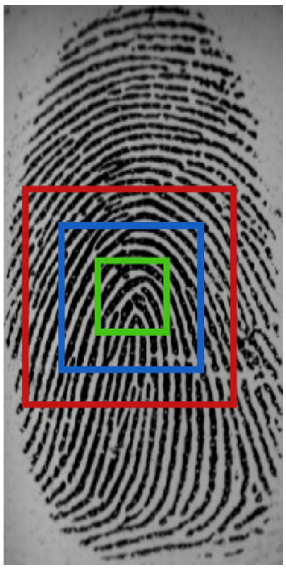


Figure 5.3: ROI sizes of 75%, 50% and 25% centred at the core of the aligned original fingerprint image.

5.3 Fingerprint Recognition Results

The fingerprint recognition methodology was first validated using visual inspection of the image data, followed by parameter tuning, and the main experiments, which perform open-set identification using the three experimental datasets.

5.3.1 Methodology Validation by Visual Inspection

The MCYT dataset captured with the capacitive sensor was used for both visual inspection and parameter tuning, discussed in the next subsection. The presegmentation method successfully eliminated the background with zero errors across all datasets. An ideal process, where HE and CLAHE both fix inconsistent ridge contrast is shown in Figure 5.4. Figure 5.5b shows the poor result when HE is used for severely damaged fingerprints during preprocessing on the MCYT images. This fingerprint core is identifiable in the case of using CLAHE, but does not appear to normalize different lighting as consistently as HE unless the clip limit is automatically tuned for various image conditions. While this is assumed to potentially yield optimal results for core identification, HE is used for preprocessing for all tests because it requires no tuning.

(a) Original image **(b)** HE image **(c)** CLAHE image (clip limit = 3) **(d)** Enhancing the HE image with nl-means, unsharp and presegmentation **(e)** Enhancing the CLAHE image with nl-means, unsharp and presegmentation

Figure 5.4: Good quality ridge enhancement before Poincaré core point detection.

(a) Original image **(b)** HE image **(c)** CLAHE image (clip limit = 3) **(d)** Enhancing the HE image with nl-means, unsharp and presegmentation **(e)** Enhancing the CLAHE image with nl-means, unsharp and presegmentation

Figure 5.5: Bad quality ridge enhancement before Poincaré core point detection.

5.3.2 Tuning of Experimental Parameters

Due to the large number of combinations in terms of metrics, modalities and algorithms, parameter tuning was constrained to a closed-set identification system measured without thresholding, resulting in a maximum DIR, that is referred to as IR as explained in Section 2.6.2. Moreover, in subsequent tests, the optimal parameters are stated only if different from those determined in this section. The identification performance on the MCYT dataset was tested first as the validation set for tuning the ROI size, as well as the various feature-classifier parameters.

To ensure that the resulting parameters do not favour the MCYT optical sensor dataset, only the left and right little fingers of the capacitive sensor are used for this purpose.

Since training samples are not used as test data, all test data are unseen. Furthermore, three different ROI sizes of 75%, 50% and 25% are tested separately to estimate how close to the core the most significant features lie on the fingerprint. The corresponding resolutions are 150×150, 100×100 and 50×50 for this dataset.

By default, all principal components are used for Eigen and Fisher classification. Moreover, the linear SVM uses $C = 1$ and LBPH uses radius and neighbourhood values of one and eight, respectively. The neighbourhood size and the number of neighbours are equal for LBPH. The default (recommended) parameters are used unless stated otherwise, which in the case of image resolution is the segmented resolution to the closest 50^{th} pixel as previously stated.

Referring to Table 5.1, the best performer is LBPH highlighted in blue. LBPH also appears to be least affected by ROI size. On average, the four classifiers achieve their best accuracies using a 50% ROI. The classic "Eigenface" approach appears to be insufficient for extracting "good features" for at least the fingerprint modality. This is of concern, as many of the feature-level fusion related studies are based on this approach and include the fingerprint modality as one of the fused components. The system was retested using CLAHE with a best result at clip limit = 3, but it underperformed compared to HE as seen in Table 5.2. CLAHE is thus excluded from subsequent tests.

Table 5.1: IR for different ROI sizes using HE on a single training sample.

Classifier	**75%**	**50%**	**25%**
Eigen	39.2	47.1	47.5
Fisher	39.2	47.1	47.4
LBPH	71.5	78.3	69.6
SVM	39.1	47.3	47.5

Table 5.2: IR for different ROI sizes using CLAHE on a single training sample.

Classifier	**75%**	**50%**	**25%**
Eigen	27.7	36.8	27.8
Fisher	27.8	36.7	27.7
LBPH	66.1	72.3	65.9
SVM	27.6	37.7	27.8

The LoG filter results are given in Table 5.3, where LBPH appears to achieve a lower

accuracy, which may mean that different radius and neighbourhood parameters should be used than the default ones.

Table 5.3: System accuracy when using HE followed by 17×17 Gaussian and 7×7 Laplacian filter.

Classifier	**HEL**
Eigen	64.0
Fisher	64.0
LBPH	63.7
SVM	64.0

Sparsely spaced ELBP and ELBPV texture descriptors are designed to be well-suited to the LoG filter's bandpass spectrum. Since the neighbours are interpolated, any number of neighbours can be set. The number of neighbours is adaptively sized to be equal to half the length of the square ROI to match the spatial information to an arbitrary sized image (for future datasets). Thus, 25 neighbours are used for a 50×50 resolution image. For a smoother image, a larger radius than the default-sized one pixel can be used. A 4-pixel radius was found to provide optimal local information without further separation of data at the intra-class level. The same LoG filter size of 17×17 and 7×7 is applied to ELBP and ELBPV. The results of using new feature extractors LBPL and LBPVL are thus listed in Table 5.4. The system accuracy effectively increased for the three classifiers when using either method but LBPL outperforms LBPVL.

Table 5.4: IR when using the novel LBPL and LBPVL feature extractors at segmented resolution.

Classifier	**LBPL**	**LBPVL**
Eigen	83.6	74.9
Fisher	83.5	75.0
SVM	83.9	74.9

Results for the different feature extractor methods are given in Table 5.5. After testing a number of resizing resolutions between 30×30 and 200×200 in both width and height increments of 10, 50×50 was found to be optimal for both decreasing the dimensionality, while also improving classifier performance for all but the LBPH classifier and the LBPL feature extractor, which both preferred the segmented resolution for this validation

dataset. This is because the LBPH uses the same ELBP operator, although with different parameters. Of note, the performance is increasingly sensitive to resolutions above 65×65. This is important because dimensionality increases when fusing two modalities, as discussed in Chapter 9. Decreasing the original resolution appears to affect LBPH (or more directly ELBP) owing to the reduced separation between texture patterns causing ambiguity with patterns of similar global appearance. The segmented resolution of 100×100 is thus used in later tests for LBPH. Furthermore, training and test time appears to be only slightly affected by the resolution and is instead, largely determined by the neighbours parameter. A confusion matrix of correct matches per label revealed that the smaller ROIs, that is, 50% and 25%, are more robust to partials and severely distorted fingerprints than images with 75% ROI. LBPVL feature extraction results greatly outperform the HE lighting normalization variant and slightly outperform HEL, which includes LoG for feature reduction. LBPL achieves noticeably lower IR values at lower resolutions ($< 75 \times 75$). However, LBPVL also outperforms LBPL at segmented resolution and by increasingly greater margins at lower resolutions. All feature extractor variants are used in subsequent tests, but only the best performer, which is explicitly stated, is included in the results in the following sections.

Table 5.5: Comparing HE, HEL, LBPL and LBPVL identification accuracies when resizing to 50×50.

Classifier	**HE**	**HEL**	**LBPL**	**LBPVL**
Eigen	54.1	82.1	79.9	85.1
Fisher	54.1	82.5	75.5	84.9
LBPH	47.5	48.7	–	–
SVM	54.2	82.3	78.2	85.1

A lower number of principal components enables faster training and testing owing to the lower dimensionality of feature vectors. It is determined whether this reduction prevents overfitting when fusing modalities in Chapter 9. The optimal number of principal components in terms of average accuracy across training samples is 200 for Eigen and Fisher, as shown in Figure 5.6. An outlier occurs when using 100 principal components for 5-Eigen. These minor outliers are expected when reducing principal components since there is a chance that bad data can be removed within the first few hundred principal components

and have an impact on the results. The extra number of principal components for Fisher is expected due to Fisher's extra reduction in dimensionality. The principal components for Eigen and Fisher are set to 200 for the remaining tests. Parameter tuning results for the other feature extractors considered for Eigen, Fisher and LBPH, such as LBPL are not given, as LBPVL produced the best result in all cases.

Table 5.6: Selecting the optimal number of principal components for Eigen and Fisher LBPVL.

Classifier	50	100	150	200	all
1-EigenLBPVL	78.5	84.1	85.1	85.1	85.1
1-FisherLBPVL	39.5	84.1	84.9	84.9	84.9
5-EigenLBPVL	97.8	98.3	98.3	98.3	98.3
5-FisherLBPVL	77.1	85.6	88.1	90.9	90.9

Table 5.7 shows the identification accuracies of LBPH, with the same LoG filter as applied in the tests reported in Table 5.3, but varying the radius from one to six, showing that radius values larger than three are preferred. A marked improvement is achieved over the 63% accuracy obtained using the default parameters with one training sample. Moreover, when using five training samples an even greater improvement is recorded except when using a 1-pixel radius which yields only 51.3% accuracy due to the abundance of fine textures causing overfitting. The best result is 86% when using a 6-pixel radius. The results for LBPH without the LoG filter have been omitted as the best accuracy obtained was 78.3% and this was at the recommended parameter values. It is noted that LBPH appears to be sensitive to tuning.

Table 5.7: Selecting the optimal radius (1–6) for LBPHL.

Classifier	1	2	3	4	5	6
1-LBPHL	63	68.9	80.4	81.6	81.6	81.7
5-LBPHL	51.3	81.3	81.1	83.3	84.8	86

Table 5.8 shows the identification accuracies for varying radii between one and six. More combinations of radius and neighbour parameters were tested, but the best result did not improve the 95% and 99.8% accuracies obtained with a 6-pixel radius and four neighbours for one and five training samples, respectively.

Table 5.9 shows that $C = 10^4$ yields the best results for the SVM, which are 85% and 97.8%

Table 5.8: Selecting the optimal neighbour count (3–8) for LBPHL with a radius of six.

Classifier	3	4	5	6	7	8
1-LBPHL	90.9	95	94	90	87.6	68.9
5-LBPHL	99.8	99.8	99.7	99.4	99	94.6

for one and five training samples, respectively. A substantial decrease in performance is noticed when $C = 10^8$ as a result of overfitting.

Table 5.9: Selecting the optimal C value for the SVM.

Classifier	$C = 10^{-8}$	$C = 10^{-4}$	$C = 1$	$C = 10^4$	$C = 10^8$
1-SVMLBPVL	83.0	83.0	83.1	85.0	76.5
5-SVMLBPVL	97.8	97.8	97.8	97.8	94.6

The results of the parameter optimization for each classifier given in Tables 5.6, 5.8 and 5.9, show that the best performing feature-classifier combination is LBPH with the LoG filter at a near-perfect accuracy of 99.8% when using five training samples and 95% when only using a single training sample. The experiments using the first dataset, consisting of varying conditions such as lighting, and fingerprint placement and pressure, achieved promising results. Hence, the optimal parameters determined above were used when running tests on various other fingerprint datasets. The parameters and feature-classifier combinations were kept constant when testing these other datasets to simplify the classification of fused datasets (as reported in Chapter 9). Therefore, the optimal range was specified as being within 5% of a different method. The difference in performance is explicitly stated when an alternative approach shows a better accuracy above the optimal range.

The next three subsections document open-set identification experiments on three different fingerprint datasets, that is, MCYT, FVC2004 and SDUMLA, organized as explained in Section 4.3 unless otherwise stated.

5.3.3 MCYT Identification Performance

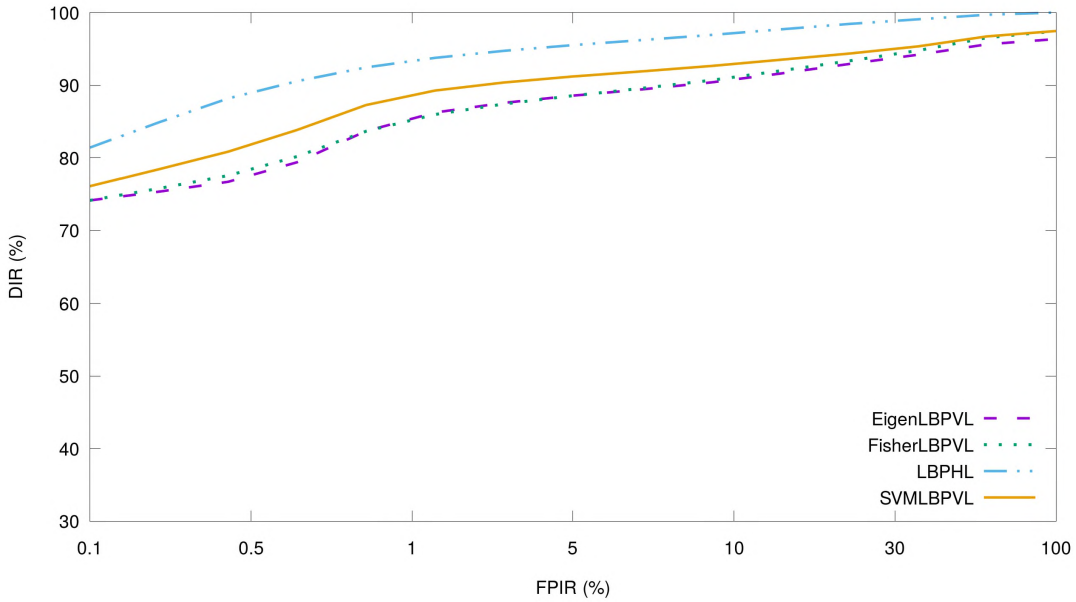
The MCYT fingerprint data were evaluated according to the organized classes explained in Section 4.3.1. Fingerprint samples one to three with a high level of control were trained

to form the model, while the rest (samples four to eleven) were used for testing. The results are presented as ROC curves.

Effect of fingerprint sample quality

This subset of the MCYT identification experiment refers to Figures 5.6, 5.7 and 5.8. The three tests used the same high control level training sample to isolate the comparison to test sample quality.

Fisher narrowly outperforms Eigen on the MCYT high control dataset. The margin by which LBPH outperforms the rest increases as the level of control (quality) decreases.



(a) DIR vs. FPIR

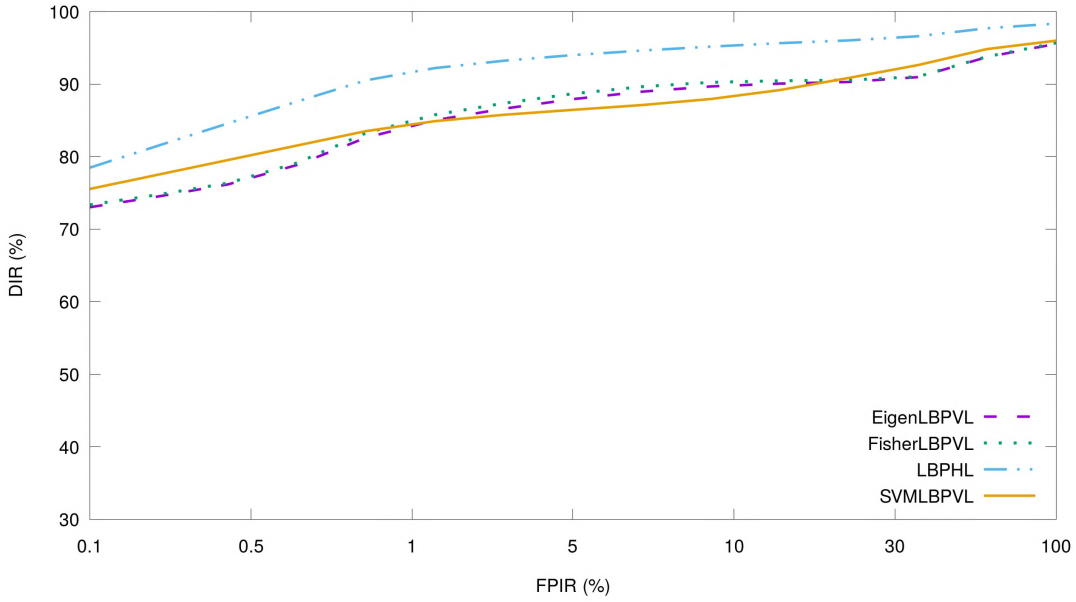
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0.3	0.5	1.3	1.5	2.7
1-FisherLBPVL	0	0	0	0	1	1.3	2.6
1-LBPHL	0	0	0	0	0	0	0
1-SVMLBPVL	0	0	0.7	0.7	1.3	1.7	2.6

(b) FNIR values

Figure 5.6: Single high control MCYT training sample on three high control test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The MCYT medium control results show that performance of Eigen is similar to that of

Fisher for known classes.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0.3	0.3	1	1.3	4.5
1-FisherLBPVL	0	0	0	0	1	1	4.4
1-LBPHL	0	0	0	0	0	0.3	1.7
1-SVMLBPVL	0	0	0	0.3	1.7	1.7	4.1

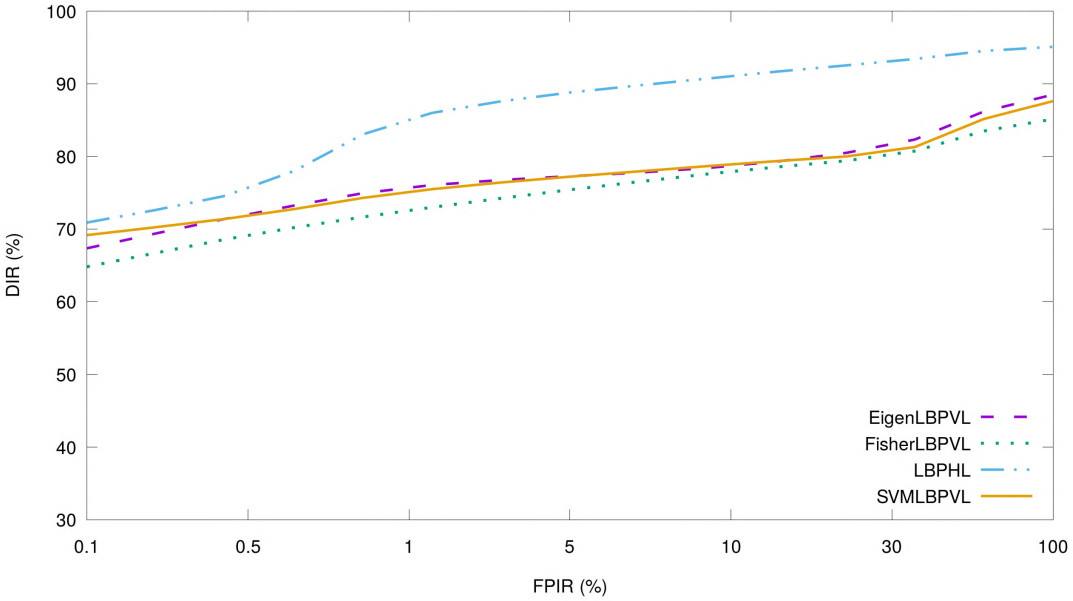
(b) FNIR values

Figure 5.7: Single high control MCYT training sample on three medium control test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The MCYT low control results show that Eigen is less affected by the intra-class variance of data than Fisher even when using a single training sample. Moreover, Fisher performs poorly with certain types of intra-class variance even with good training data. SVM appears to better distinguish impostors from class data than either Eigen or Fisher.

All data samples

All of the MCYT samples are used for further identification testing as described in Section 4.3.1.



(a) DIR vs. FPIR

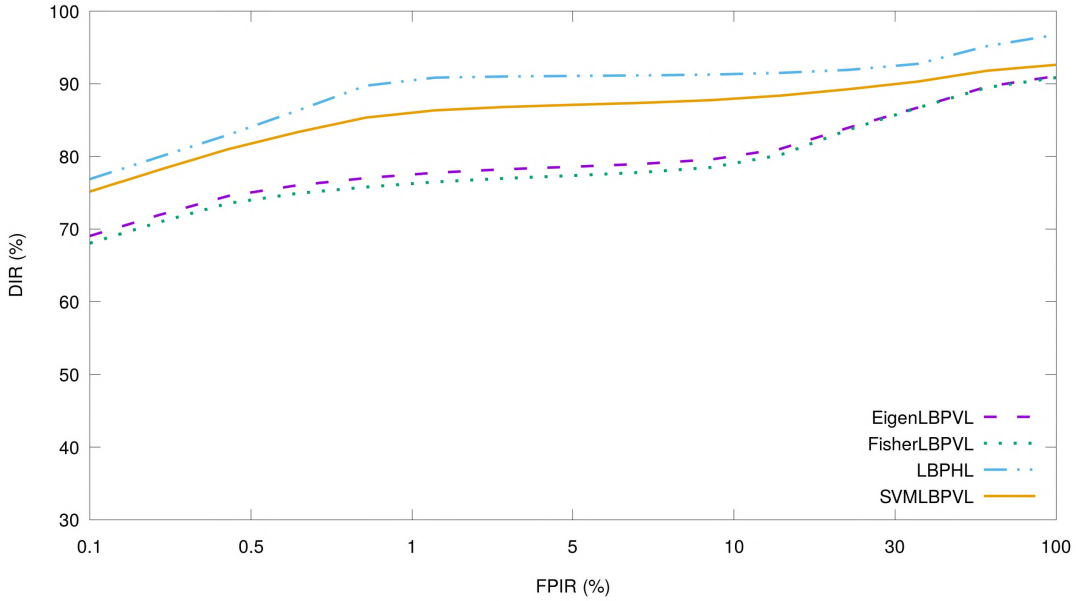
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.7	5.5	11.3
1-FisherLBPVL	0	0	0	0.3	0.7	6.7	14.7
1-LBPHL	0	0	0.3	0.3	0.5	1.7	3.7
1-SVMLBPVL	0	0	0	0.3	0.7	6.9	12.4

(b) FNIR values

Figure 5.8: Single high control MCYT training sample on three low control test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

Eigen outperforming Fisher when increasing the training samples to three (Figure 5.10b) indicates inconsistent segmentation due to inaccurate core detection or texture warping of fingerprints obtained with medium and low control.

Visual inspection revealed that fingerprint cores were consistently detected for high and medium controlled data, but noticeably less consistent for low controlled data. All three controlled data groups contained samples with mildly varied pressure resulting in warped texture in a minority of cases. The cases where the system segmented the low control fingerprints inconsistently are thus attributed to bad core detection. Furthermore, the SVM classifier appears to outperform the rest in detecting impostors. On the other hand, SVM achieves higher miss rates when FPIR < 100.



(a) DIR vs. FPIR

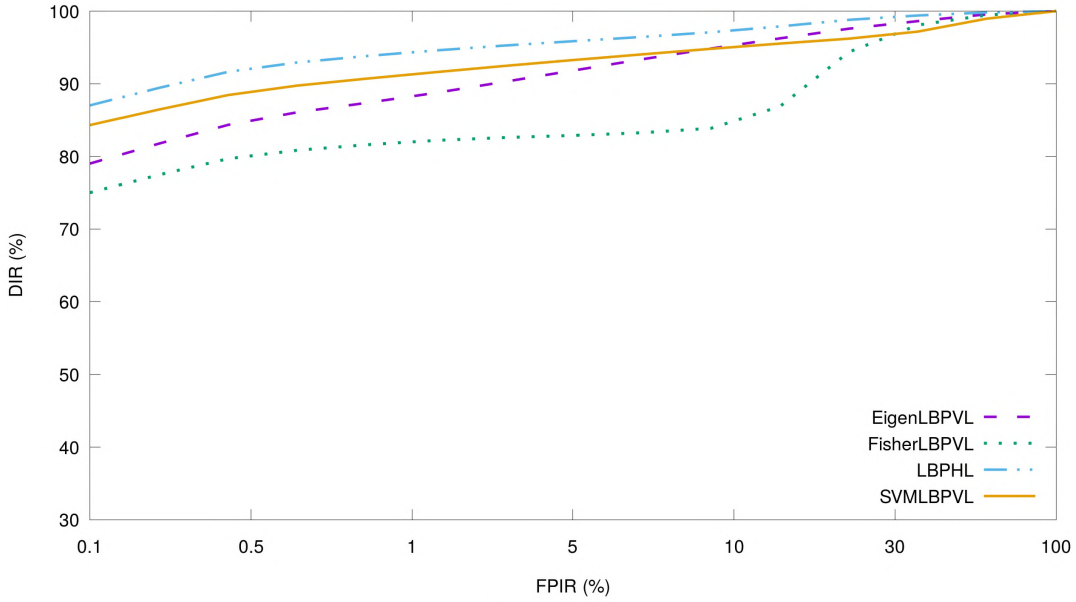
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.5	1.6	8.9
1-FisherLBPVL	0	0	0	0	0.5	1.9	9.2
1-LBPHL	0	0	0	0	0	0.3	3.3
1-SVMLBPVL	0	0	0	0	1.0	2.5	7.4

(b) FNIR values

Figure 5.9: One MCYT training sample on all test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

Referring to Figures 5.9 and 5.10, increasing the number of training samples from one to three does not necessarily improve FNIR significantly (when FPIR < 100), but generally improves overall accuracy by either increasing the DIR and/or decreasing the FPIR. Furthermore, FNIR generally decreases or remains the same when the threshold is decreased. When using three training samples, LBPH, Eigen and SVM results are more comparable, with Eigen showing the highest improvement.

The same experimental procedure conducted by Simon-Zorita *et al.* (2003) is used for comparison with LBPHL – the best performing proposed method on this dataset. The MCYT dataset subcorpus consisting of 120 fingerprints per the first 75 test subjects was used. The standard method for selecting impostors in verification systems is used, which verifies a subject and uses the remaining subjects as impostors. In this case, the remaining



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0.2	0.9	0
3-FisherLBPVL	0	0	0.1	0.1	0.3	1.2	0
3-LBPHL	0	0	0	0	0	0	0
3-SVMLBPVL	0	0	0	0.5	0.9	1.9	0

(b) FNIR values

Figure 5.10: Three MCYT training samples on all test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

75 subjects are thus impostors. However, only the first high control sample is used as an impostor, resulting in a total of 55000 impostor attempts.

The experimental procedure is separated into three tests where three training samples per low, medium or high control are used and the rest become test samples. Their results in Table 5.10 show that the proposed LBPHL method outperforms Simon-Zorita *et al.*'s minutiae-based system and achieves similar accuracies when training on high or medium samples. The better result suggests that high quality training samples can model low quality test samples better than vice versa.

Table 5.10: Comparative performance of the proposed approach for fingerprint verification on the MCYT dataset.

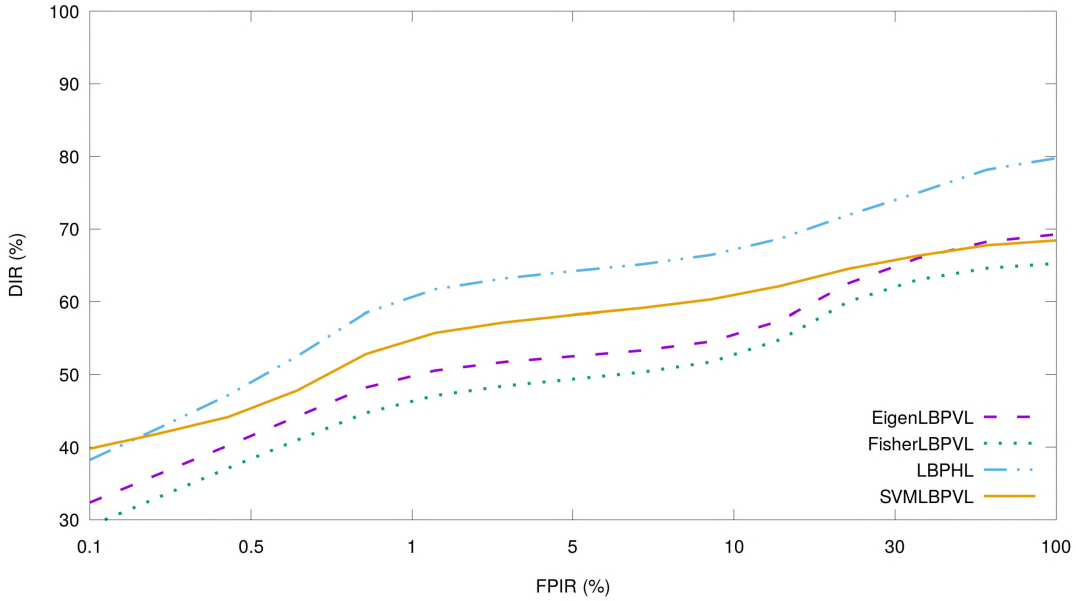
Approach	Low (%)	Medium (%)	High (%)
3-LBPHL	1.9	1.3	1.2
Simon-Zorita *et al.* (2003)	8.0	6.1	5.6

5.3.4 FVC2004 Identification Performance

The FVC2004 fingerprint dataset was evaluated according to the organized classes given in Section 4.3.2. The previously tuned parameters were confirmed to be optimal on this dataset even though it was captured using a different sensor type under challenging conditions. The results are discussed below.

Referring to Figures 5.11 and 5.12, increasing the training samples benefits LBPH the least whereas SVM slightly surpasses LBPH's performance. Furthermore, it is evident that Eigen and SVM make better use of the extra training data even with relatively high intra-class variance. Fisher continues to underperform most noticeably on the challenging data (high intra-class variance). Of note, LBPVL and LBPL perform similarly (poorly) on this dataset, but HEL achieves a 3% better accuracy on average (not shown in the figures).

The FVC2004 DB1 fingerprint dataset was evaluated using the experimental procedure of Chen *et al.* (2006b), which included results of the fingerprint verification system of Luo *et al.* (2000). The LBPHL proposed approach is thus compared with these two studies on fingerprint verification as no related studies were found for identification on this dataset. The verification procedure was on 8 samples per 100 classes, where only the first sample of each non-matching class was considered an impostor resulting in 4950 impostor matches. The results of the comparison are shown in Table 5.11, where the LBPHL texture-based approach outperforms both minutiae-based methods by obtaining a lower EER.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	1.2	4.5	8.4	30.7
1-FisherLBPVL	0	0	0	1.5	5.4	11.8	35.8
1-LBPHL	0	0	0	0	5.4	11.8	21.2
1-SVMLBPVL	0	0	0	0.1	5.4	12.9	31.6

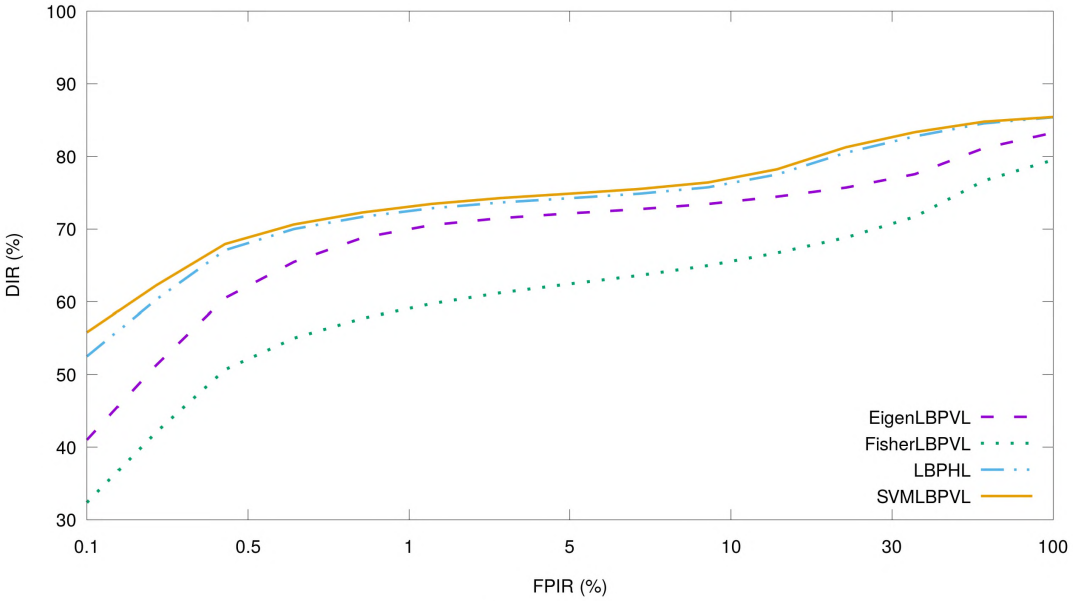
(b) FNIR values

Figure 5.11: One FVC2004 training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

5.3.5 SDUMLA Fingerprint Identification Performance

Referring to Figures 5.13b and 5.14b, the SDUMLA fingerprint datasets also favour LBPH, which achieves the highest DIR. SVM performs well at lower FPIR values and for this dataset more training samples improve performance of all classifiers similarly. In fact, they achieve similar accuracies at FPIR = 100%. LBPVL and HEL perform similarly for this SDUMLA fingerprint dataset. Poor performance by LBPH for one training sample when FPIR < 1 is noticeable.

In Zuobin *et al.* (2017)'s verification approach, five-fold cross-validation is performed on the left index fingerprint for ten runs on the 106 subjects and the recognition results are averaged. Table 5.12 shows results of 88.53% and 100% for the related study and proposed LBPHL method, respectively. LBPHL also achieves zero error when using only



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	1	2.9	5.8	16.7
3-FisherLBPVL	0	0	0	1.5	3.9	8.8	20.5
3-LBPHL	0	0	0	1	1	5.9	14.7
3-SVMLBPVL	0	0	0	1	1	5.6	14.5

(b) FNIR values

Figure 5.12: Three FVC2004 training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

three training samples and testing two test samples, from the first five samples per subject in the dataset.

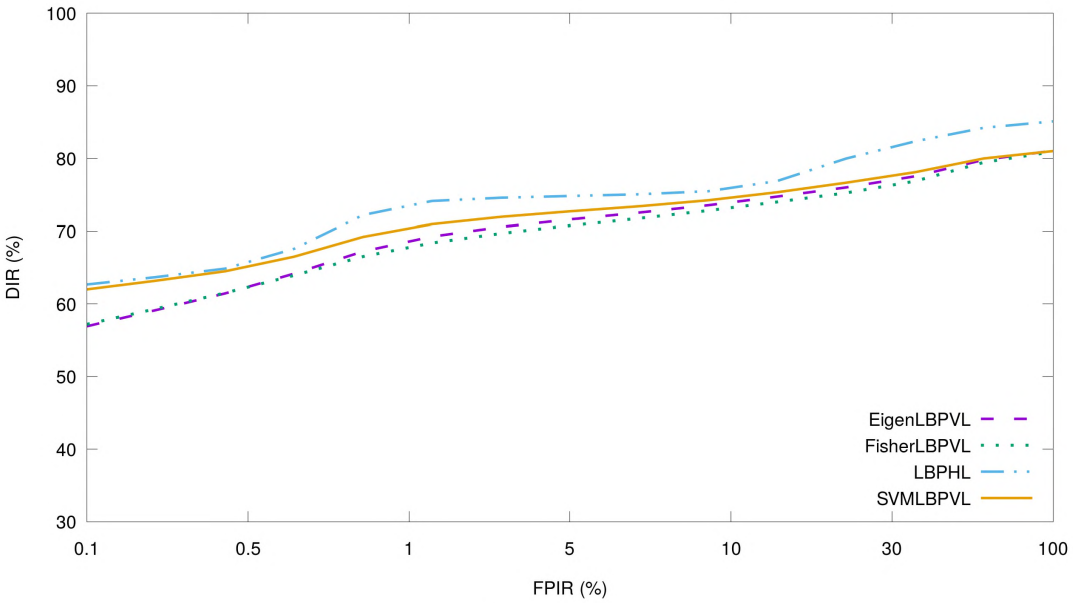
5.3.6 Discussion of Fingerprint Results

The Eigen, Fisher and SVM results improved greatly with the use of HEL or LBPVL. Although lighting normalization did not benefit LBPH, the addition of the LoG filter did improve its results. Parameter tuning provided the greatest improvement to LBPH and is an issue that will be further explored when testing the other modalities.

The MCYT experiments on levels of control indicate that Fisher's performance is significantly affected by the quality of the dataset even when using a single training sample.

Table 5.11: Comparative performance of the proposed approach for fingerprint verification on the FVC2004 dataset.

Approach	EER (%)
1-LBPHL	3.6
Chen *et al.* (2006b)	4.4
Luo *et al.* (2000)	9.1



(a) DIR vs. FPIR

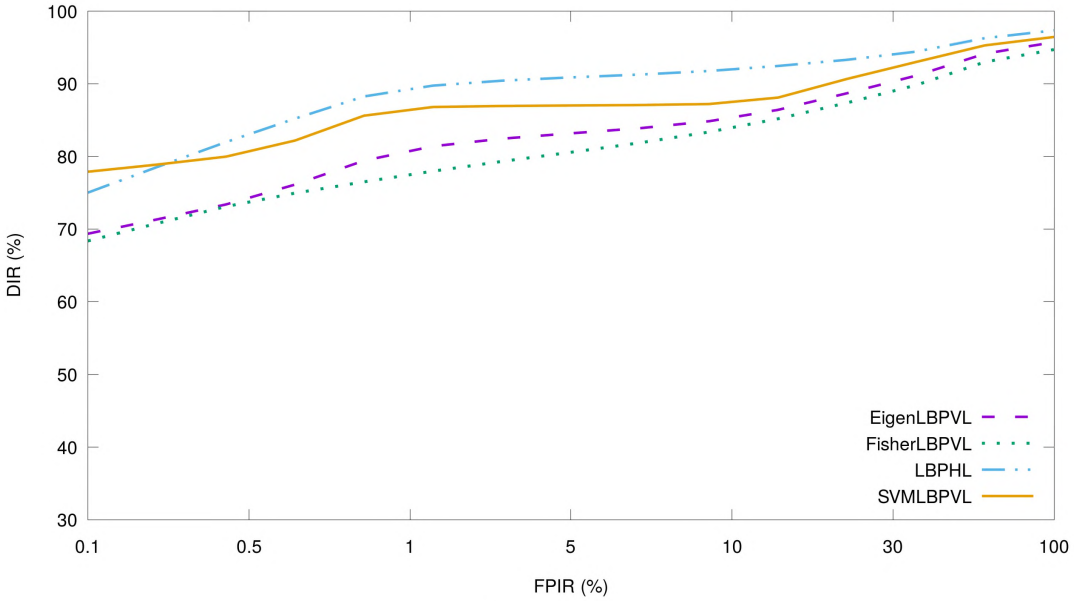
Classifier	**0.1**	**0.5**	**1**	**5**	**10**	**30**	**100**
1-EigenLBPVL	0	0	0	0	0.3	7.8	18.9
1-FisherLBPVL	0	0	0	0	0.3	8.3	19.1
1-LBPHL	0	0	0	0	0.8	3.4	14.1
1-SVMLBPVL	0	0	0	0	0.9	7.5	18.9

(b) FNIR values

Figure 5.13: One SDUMLA fingerprint training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The other two datasets, FVC2004 and SDUMLA, were more challenging as confirmed by the results, even though they contain 8-fold fewer classes than MCYT. Fisher consistently performed poorly on these last two datasets especially when using multiple training samples.

SVM avoids false positives much better than other classifiers and thus appears to perform most consistently at lower FPIR values. This could be attributed to the use of probability



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0.4	4.3
3-FisherLBPVL	0	0	0	0	0.5	0.6	5.3
3-LBPHL	0	0	0	0	0.3	0.5	2.7
3-SVMLBPVL	0	0	0	0	0.4	0.9	3.6

(b) FNIR values

Figure 5.14: Three SDUMLA fingerprint training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

estimates compared to nearest-neighbour distance values for determining when to reject an impostor. Based on the overall results, a weakness of the texture-based fingerprint method is that even when detecting the core accurately, warped texture often results in bad accuracy in general regardless of the method used, albeit to a lesser extent with LBPH. Furthermore, LBPH sometimes achieves worse FNIR values than other classifiers, especially as the matching threshold is decreased.

5.4 Summary

This chapter described the algorithms implemented for constructing the proposed fingerprint recognition system. The first step after acquiring the preprocessed image was

Table 5.12: Comparative performance of the proposed approach for fingerprint verification on the SDUMLA fingerprint dataset.

Approach	(%)
LBPHL	100
Zuobin *et al.* (2017)	88.5

presegmenting the silhouette of the fingerprint. Keypoints were detected and aligned such that the core point of the fingerprint was selected as the centre of the ROI, consisting of 50% of the presegmented image. The common postprocessing method ensured well-aligned fingerprints with consistent lighting before providing the classifiers with feature vectors obtained from the pixel values of the ROI.

In this chapter we also evaluated the performance of the proposed fingerprint recognition system by first tuning classifier parameters on the MCYT dataset, which used a capacitive sensor, followed by open-set identification results on three fingerprint datasets that used different acquisition methods including sensors. Eigen and Fisher classifiers achieved similar accuracies for a single training sample. A trend was observed where SVM noticeably favoured a better FPIR than FNIR relative to the other classifiers. Based on the average, this trend suggests that SVM falsely rejects slightly more genuine individuals but correctly rejects more impostors noticeably more than the other classifiers. LBPHL outperformed the other three classifiers in general; whether this remains to be the case for the other modalities, is considered in the following chapters.

Chapter 6

Face Recognition Algorithms and Results

This chapter explains the algorithms used to detect the initial face window and align keypoints within that window, known as landmarks, for various face poses and/or occlusions. These landmarks are organized as a mesh of triangles, which are used to align an input face image according to a database face image with the use of a novel method, called varied reference angle (VRA[1]). The VRA method is validated and classifier parameters are tuned. The open-set identification accuracy is tested on three datasets and the results are discussed.

6.1 Aligning Face Features

The keypoints on the face, that are used for alignment, are the eyes, nose and mouth. These points are used to create a border around the face, which helps to avoid typical changes to the features on and around the face caused by hair, occluded ears and neck scarves.

[1]VRA: First developed in our previous work (Brown and Bradshaw, 2017b).

The face registration process is summarized as follows. Face detection is the first step in aligning input face images according to the database face images. Face landmark detection follows whereby 68 landmarks comprising the keypoints on the face are detected. The input face image is subsequently aligned by texture warping according to a Delaunay triangulation mesh (Kouzani *et al.*, 2000).

6.1.1 Face Detection

Before face landmarks can be detected, an initial bounding box around the face must be estimated. In profile poses, this initialization can fall outside the face region resulting in a bad base mesh that cannot converge to the actual face shape. The initial bounding box is thus determined via a face detection method based on the HoG detector of Felzenszwalb *et al.* (2010), but using the improved window scoring algorithm of King (2015), which uses a similar maximum margin method as a linear SVM, as described in Section 3.2.2.

A face detection model is trained using MMOD (King, 2015) on the feature vectors produced by the improved HoG detector as follows:

Faces smaller than the 80×80 HoG filter box are detected by upsampling the image using the Gaussian pyramid method (Adelson *et al.*, 1984) and is used when MMOD detects a higher score. Five HoG filters are trained using MMOD, consisting of the following samples taken from the labelled faces in the wild face cropped dataset (Huang *et al.*, 2007): one frontal, and the remaining four at left, right, up and down poses of 45°. The number of training samples per pose is 4748. The HoG filters are trained using the following MMOD parameters (Equation 3.5): $C = 700$, $\varepsilon = 0.05C$, $L_{fa} = 1$, and $L_{miss} = 1$. The HoG detector operates by creating an image pyramid and scanning the detector over each pyramid level in a sliding window fashion thus generating multiple candidate windows. The image pyramid downsamples at a ratio of $\frac{5}{6}$.

The thresholds of NMS and MMOD for window overlaps are set as follows. More than one window is considered by MMOD when their union contributes at least 75% of a 80×80 window and are used as output. Any (remaining) windows with less than 40% overlap

are rejected by NMS. This leads to a single detection window that results either from overlapped windows aggregated by MMOD (due to a higher combined score) or selecting a single window with the highest score remaining from NMS.

6.1.2 Initial Face Alignment

An initial alignment is performed by mapping the eyes of the input image to reference coordinates located at the centre of the detected face window using a similarity transformation. This caters for matching face images with different scales and large differences in the pose angle. The resulting warped input image is thus aligned such that the inter-eye distance and chin-eye distance are the same, using the following landmarks: 37 (left eye), 46 (right eye) and 9 (chin). These landmarks are denoted in Figure 6.1 and further explained in the next section. The affine transformation T performs rotation, scaling and translation to map a point $\begin{bmatrix} x & y \end{bmatrix}^T$ on the input face to a point $\begin{bmatrix} x' & y' \end{bmatrix}^T$ on the reference face as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{6.1}$$

where a, b, c, and d are the rotation and scaling parameters, and tx, ty are the translation parameters. The roughly aligned input image is accurately aligned by remapping triangle regions on the face as explained in the next section.

6.1.3 Automated Face Landmark Detection and Labelling

Face landmark detection is achieved by implementing the regression tree method by Kazemi and Sullivan (2014) explained in Section 3.2.2, using the Dlib library (King, 2009). In the proposed implementation, a cascade of regressors consisting of facial landmarks imperative for the alignment process, is built by continually updating a vector of 68 points, within the initial window detected in Subsection 6.1.1. The 68 landmarks

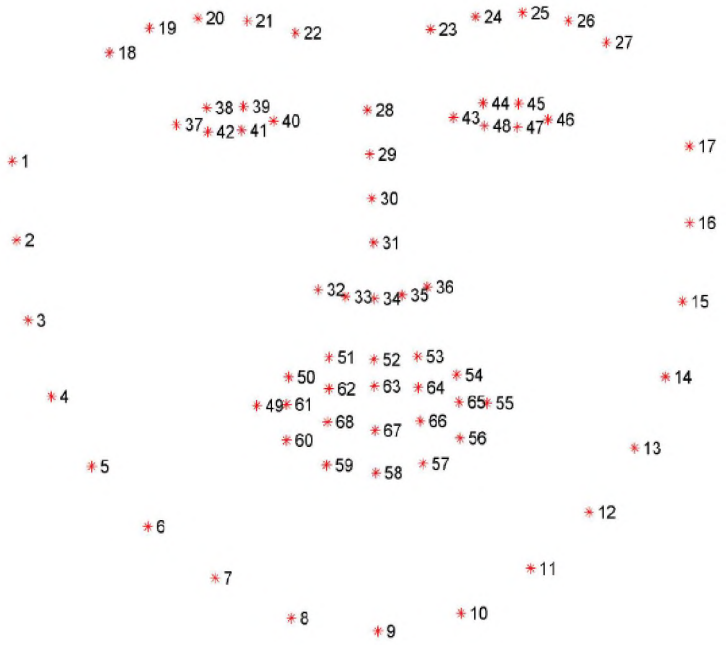


Figure 6.1: Face landmarks [taken from (Sagonas *et al.*, 2016)].

are mainly located as outline positions surrounding the eyes, nose, and mouth regions as shown in Figure 6.1.

Training data samples, acquired from the iBUG 300-W face landmark dataset (Sagonas *et al.*, 2016), include the coordinates of all 68 points per face. The learning stage is conducted by training coordinates as a set of triplets forming the input of the learning function for the next regressor in an iterative process. A decision is made based on the difference between the intensities of two pixels at each split node in the regression tree. The difference defines coordinates of the mean shape of the face.

6.1.4 Face Mesh

The landmark points enable the computation of a Delaunay triangular mesh that covers most of the face, avoiding the forehead area due to typical occlusions such as hair and various accessories. The face mesh is constructed such that no landmark is inside the circumcircle of any triangle. The pose angle can be accurately estimated based on the location of six coordinates, that is, the nose, eyes, left mouth edge, right mouth edge and

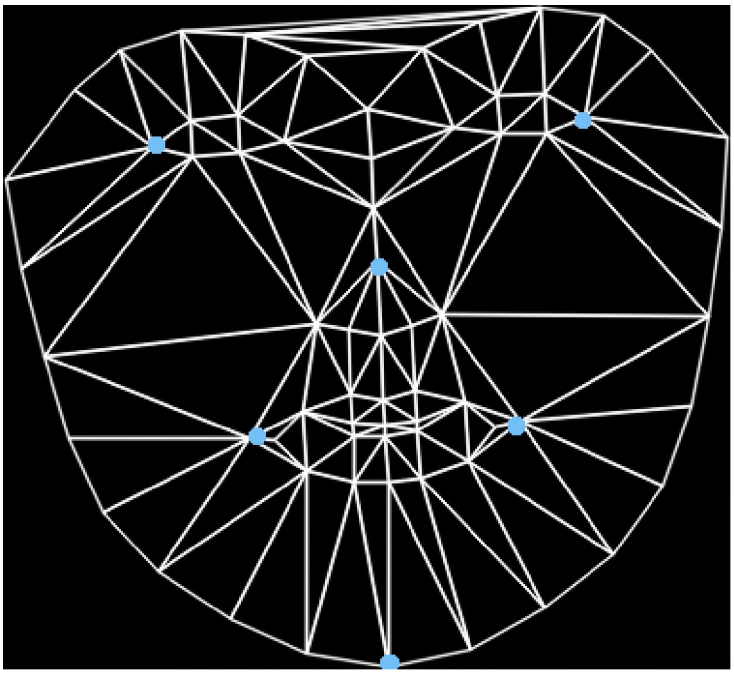


Figure 6.2: The six landmarks (blue dots) on a face mesh used for pose angle estimation.

chin as shown in Figure 6.2. The pose angle estimation is carried out using OpenCV's implementation of the Perspective-n-Point (Gao *et al.*, 2003) with the six coordinates given as input.

6.1.5 Face Frontalization

The current state-of-the-art biometric systems align texture before performing face recognition between images using frontalization as performed by Haghighat *et al.* (2016) in Section 3.5.2. The objective of this approach is to perform affine transformations of corresponding triangles on both the input and database mesh by using a frontal mesh generated from the (ideal frontal) landmark points in Figure 6.1 as reference. This method works well for smaller pose angles, but results in substantial distortion above 30° as shown in Figure 6.3. This often leads to utilizing the half-face method due to self-occlusions of large pose angles. Hence, a novel alternative to face frontalization is contributed in the next section, named VRA (Brown and Bradshaw, 2017b), which is compared in the methodology validation section with frontalization.

(a) 25° **(b)** 45° **(c)** 60°

Figure 6.3: Increasing levels of distortion with larger pose angles [taken from (Phillips *et al.*, 1998)].

6.2 VRA Segmentation

VRA is a novel fast accurate face segmentation method developed during this research. VRA improves face recognition versatility by accepting varied angle face data for either/both input and database images by calculating the smallest-difference pose angle between them without using 3D modelling. The smallest-difference angle is calculated by adding the pose angles of a database and input image, and dividing the result by two. VRA allows for both the removal of frontal training data dependence by accepting varied angle face data, which reduces the distortion for angles larger than 30° and no longer requiring the half-face method due to self-occlusions at those angles. It is theorized that the half-face method loses more information than VRA for self-occluded face images (due to large pose angles).

6.2.1 General VRA Procedure

Following this VRA procedure on identification, a scenario where $\theta_s < 30°$, allows for face biometric systems to identify an individual without using the smallest-difference angle (nor frontalization), but instead warps input images according to the pose angle of the image in the database. This enables all database images to be used as input for training samples, as distortion is relatively low or unnoticeable at these smaller angles. When $\theta_s \geq 30°$, only the database image(s) closest to the smallest-difference, with a 10° tolerance, are used yielding the best segmentation result over having multiple but distorted database images as training data. The general VRA procedure can be broken down into the following steps:

1. Given input image I and same class database image set $\{F_x\}$, containing angle set $\{\theta_x\}$. Angles are negative if they are left poses. Let θ_s be the smallest-difference angle between I and items in $\{F_x\}$, where the set of database images meeting the requirements within 10° (tolerance) of θ_s is referred to as $\{F_s\}$.

2. If $\theta_s < 30°$, perform the following:

(a) Warp I to angle θ_s

(b) Perform the postprocessing step (Section 4.1.3) on $\{F_s\}$ and I to prepare them as training and test samples, respectively.

3. If $\theta_s \geq 30°$, perform the following:

(a) Mirror I when I and pose angles $\{\theta_x\}$ of $\{F_x\}$ are in different directions

(b) Recalculate $\{F_s\}$ as $\{F'_s\}$, and if the new angle θ_s is greater (worse) than 10° away from the old value, warp both I and the original $\{F_s\}$ images to the original angle θ_s. Otherwise:

(c) Warp both I and the new (mirrored) set of images $\{F'_s\}$ to the new angle θ'_s

(d) Perform the postprocessing step (Section 4.1.3) on $\{F'_s\}$ and I to prepare for classification

An exception to the above procedure is made when warping away from the self-occluded side of a face with a pose angle larger than 60°. This limit is set as warping in this way yields the least usable (almost fully occluded) data. In other words, less data is lost when warping a face image with a smaller pose angle to a larger pose angle than vice versa. In this case either the input or database image is warped toward whichever image has the larger pose angle. The general VRA procedure is effective in verification systems as models are trained only for a single class and can thus be easily manipulated with insignificant time penalties when compared with identification systems. Therefore, an alternative VRA approach is proposed for identification in the following subsection.

6.2.2 Practical VRA Implementation for Face Identification

Operations on the database image are avoided where possible for practical considerations. Therefore, this implementation of the VRA procedure avoids modifying the model used for classification, such that continuous model retraining during (1:N) testing is not required. This VRA approach is thus ideal for applications with limited access to data but require

a real-time response, such as using an ID photo as a database image to identify a missing individual on camera feeds at arbitrary pose angles. The smallest-difference angle is limited by storing fixed database angles at 30°, 45°, 60° and 75° to avoid modifying the model after training it (except when adding a new class). This is further explained in the example below.

Given a single input and database image, if their corresponding input and database meshes have 0° and 55° pose angles, respectively, they are both warped by shifting them toward the smallest-difference angle, which is ideally 27° in this case. This example proceeds as follows:

For practical considerations of identification experiments in this chapter, all steps on the input image only take place at the classification step and database images are processed as follows. The F_x images with (right) poses of 30°, 45°, 60° and 75° are each trained as a separate model when used later in the classification step. Mirroring is thus performed on all database images that have left poses. Using these fixed angles avoids retraining a model each time an input image is tested in scenarios where $\theta_s \geq 30°$. However, if the database images do not contain these fixed angles or are simply limited (as in this example), the following warps are performed on the database image to generate the missing fixed angles. The set of database images that match the fixed angles the closest are warped according to the adjacent fixed angle in either direction by giving preference to warping smaller pose angled images to bigger ones.

During the classification step in this example, 30° is selected as the best database candidate and is thus the training image. Therefore, when the pose angle of I equals 55°, the test sample (that is I) is warped to 30° and its class is predicted against the training sample (containing a fixed 30° pose angle) within the model.

While the above identification problem can be viewed as a disadvantage, the fact that VRA does not require the training image to be frontal or near-frontal means increased application versatility. In the classification step only the best (typically one) database image, as determined by VRA, is used as a training sample for a model. This avoids the issue of the frontalization method, where classifier overfitting is likely with the use

(a) Database image **(b)** Input image

Figure 6.4: Database and input images of the same person (PUT face dataset) [taken from (Kasinski *et al.*, 2008)].

of multiple distorted training samples. However, VRA produces multiple (same class) training samples for use by the classifier if there are multiple database images with the pose angles all within 10° of each other.

The example of VRA segmentation is extended to the visual results on PUT dataset images. The database image used for the reference mesh of a particular individual (class) is shown in Figure 6.4a to be near-frontal. The corresponding mesh is shown in Figure 6.5a. On the other hand, the input image of the same class has an estimated 55° pose angle, as depicted in Figure 6.4b with the corresponding mesh shown in Figure 6.5b. Assuming this is an identification problem, the input and database images are thus warped to θ_s, which results in a pose angle of 30°, to minimize the distortion. The VRA images to be used as training and test images for classification are shown in Figure 6.6a and 6.6b, respectively.

A limitation of VRA is that it does not take advantage of multiple training samples at different angles outside the 10° tolerance, and instead, the single best angle with the lowest distortion is utilized in the above example. An adaptive boosting algorithm that can take advantage of viable additional samples is considered a good automated way to remedy this.

(a) Database mesh **(b)** Input mesh

Figure 6.5: Database and input image meshes of the same person.

6.3 Face Recognition Results

The face recognition methodology was validated both by visual inspection and empirically on the FERET b-series dataset. As with the fingerprint results section, open-set identification and verification of three datasets were used to evaluate the effectiveness of the tuned system.

6.3.1 Methodology Validation by Visual Inspection

As with the fingerprint, lighting and alignment largely affects segmentation and recognition performance. HE and CLAHE were tested on the face with HE producing more consistent results for both frontalization and VRA methods. The HE method is thus used to obtain the segmented results for frontalization and VRA. The visual difference in performance between frontalization and VRA applied to the database and input images (Figure 6.4) is shown in Figure 6.6. The frontalization method shows more distortion than the VRA method. These visual results were tested empirically as discussed at the end of the next section.

The normalized and segmented training and test images at 30° are shown in Figures 6.6a and 6.6b. There are still differences between the training and test images in certain parts, but the eyes are well normalized due to their relatively simple shape in the mesh. Facial

expression normalization is a non-trivial research area and is considered beyond the scope of VRA.

(a) VRA training image at 30°

(b) VRA test image at 30°

(c) Original training image

(d) Frontalized test image

Figure 6.6: Comparing VRA and frontalization segmented training and test images.

6.3.2 Tuning of Experimental Parameters

The VRA closed-set identification performance on the FERET b-series dataset was used as the validation set for tuning the ROI size, as well as the feature-classifier parameters. The system accuracy metric was again used as it is a closed-set system test. As with fingerprint identification, no training samples were included in the test data with the result that all test data were unseen. Furthermore, the first five training samples consisted of one frontal and four left pose angles of 60°, 45°, 25°, and 15°, respectively. Face results were optimal for an image resolution of 150×150 (segmented resolution), instead of the 50×50 used for the fingerprint, as determined empirically. The DIR values were, on average, 2% better at 150×150. A deficit of 5% is defined as the optimal range such that a low image

resolution close to 50×50 is preferred for all biometric modalities to simplify the fusion process in the later chapter. As with the fingerprint, LBPH uses 150×150 as the input resolution and processing speed is not greatly affected compared with the other classifiers. It should be noted that the face is markedly less affected by resolution reduction than the fingerprint.

Eigen and Fisher classifiers appear to require little tuning as previously recorded in Table 5.6 and now in Table 6.1. The number of principal components was thus unchanged at 200. Of note, Fisher slightly outperforms Eigen for three training samples, suggesting that training data is of good quality, but since five training samples yield a perfect accuracy for both Fisher and Eigen, further comparisons cannot be made.

Table 6.1: Selecting the optimal number of principal components for Eigen and Fisher LBPL.

Classifier	50	100	150	200	all
1-Eigen	87.6	87.8	89.3	92.7	92.7
1-Fisher	50.7	88.3	91.6	92.9	92.6
3-Eigen	97.5	97.5	97.1	97.5	97.3
3-Fisher	91.2	95.4	97.9	98.6	98.6

Table 6.2 shows the system accuracies for LBPH (in the same way as for fingerprints). A significant improvement is achieved over the 67.01% of the default parameters for one training sample. The rest of the results on LBPH without the LoG filter have been omitted as this method achieved a best accuracy of 78.3% only using the recommended parameter values. The LBPH parameters thus appear to be texture resolution dependent as well as modality dependent. Increasing the number of training samples to five appears to cause inter-class ambiguity within the model, resulting in a lower accuracy than when using only three training samples.

Table 6.2: Selecting the optimal radius (1–6) for LBPH.

Classifier	1	2	3	4	5	6
1-LBPHL	47.1	63.9	69.1	72.7	72.6	71.6
3-LBPHL	75.8	79.8	82.9	85.0	84.1	83.7
5-LBPHL	67.1	73.6	79.1	78.5	74.3	74.2

Table 6.3 shows the system accuracies when selecting a radius between one and six. The

best result of 86.5% is obtained when using a 4-pixel radius and six neighbours. This is substantially lower than the Eigen and Fisher results and was further explored with the open-set identification tests. The results on LBPH without the LoG filter have once again been omitted as the best accuracy of only 67.71% was achieved using the recommended parameter values. Increasing the radius beyond one, when not using the LoG filter, appears to have a negative effect on all datasets tested so far when not using the LoG filter. This suggests that the recommended parameters for LBPH can be a good starting point or default selection when not combining it with a LoG filter.

Table 6.3: Selecting the optimal neighbour count (3 – 8) for LBPH when using a radius of four.

Classifier	3	4	5	6	7	8
1-LBPHL	37.7	66.1	71.3	76.6	74.8	71.8
3-LBPHL	74.3	85.2	84.8	86.5	84.6	84.1
5-LBPHL	74.3	81.1	83.0	83.1	82.6	82.0

The linear SVM is unchanged as $C = 10^4$ is optimal. In fact, only changing C by a factor of more than 10^3 resulted in a slight change in accuracy ($\pm 1\%$), suggesting that the extracted features used as training and test data are balanced. The optimal parameters for frontalization were found to be the same as VRA, suggesting texture patterns (modality dependent) to be the determining factor.

As the FERET b-series dataset is popular for verification tests, the verification results of the proposed approaches are included for comparison against Haghighat *et al.* (2016)'s state-of-the-art system. Tables 6.4 and 6.5 compare the verification accuracies at different angles ranging from $-60°$ (subject's left) to $+60°$ (subject's right) using parameters particular to frontalization and VRA. All four proposed approaches are included in the tables to show their relative robustness to the range of pose angles. The images were captured at a resolution of 512×768 fine quality. Three frontal images consisting of a neutral expression, an alternative expression, and different illumination were used for training the 200 classifier models (classes) for verification. Since LDA requires at least two training samples to perform verification (due to single class modelling), three training samples were used for Fisher and the other classifiers for equal comparison.

LBPH underperforms regardless of the pose angle and only occasionally performs better when using frontalization for narrow angles. Fisher takes advantage of well-segmented training samples when using VRA to achieve the best average verification rate. Eigen and SVM perform comparably to Fisher for VRA, and better than Fisher for frontalization. This shows that Eigen and Fisher are more robust to profile poses when using frontalization while LBPH generally recognizes face texture relatively poorly. These are promising results for VRA and also provide interesting insight into the effects of texture alignment in general. The limits of VRA are further explored in the next sections.

Table 6.4: Verification accuracy (%) for eight viewing angles on the FERET b-series dataset using the frontalization method with three frontal training samples.

Method	**+60°**	**+45°**	**+25°**	**+15°**	**−15°**	**−25°**	**−45°**	**−60°**
Haghighat *et al.* (2016)	91.5	96	100	100	100	100	99	93.5
3-EigenLBPL	91.8	96	100	100	100	100	96.5	92.25
3-FisherLBPL	87.3	89.5	100	100	100	100	88.8	88.5
3-LBPHL	89.8	92.5	94.8	96.3	96.3	95.3	96.3	90.5
3-SVMLBPL	91.5	95.3	100	100	100	100	96	93.5

Table 6.5: Verification accuracy (%) for eight viewing angles on the FERET b-series dataset using VRA with three frontal training samples.

Method	**+60°**	**+45°**	**+25°**	**+15°**	**−15°**	**−25°**	**−45°**	**−60°**
Haghighat *et al.* (2016)	91.5	96	100	100	100	100	99	93.5
3-EigenLBPL	97.8	99.5	100	100	100	100	100	98.3
3-FisherLBPL	98.3	100	100	100	100	100	100	99.5
3-LBPHL	91.8	92.5	95.8	95.3	95.3	94.3	96.3	93.3
3-SVMLBPL	98.3	100	100	100	100	100	100	98.3

The next sections focus on open-set identification accuracy for three experiments on different face datasets. Training samples were removed from the available samples and the rest were used for testing, thereby ensuring that the test data were unseen as was the case for all previous experiments. The training samples were limited to a maximum of three as using three or more produced near-identical trends, and in some cases, resulted in an unchallenging open-set identification test.

6.3.3 Fei Identification Performance

Open-set identification testing on the Fei dataset uses both frontalization and VRA methods. In this experiment, the results using frontal faces as database images are referred to as *frontref*, while results using left pose at 60° as a database image are referred to as *leftref*. When using three training samples in *frontref* experiments, three frontal training samples are used. However, when using three training samples in *leftref* experiments, 0° and left pose 30° and 60° images are used. The trends of the classifiers with respect to the two methods are similar, and thus for the sake of brevity only the best classifier result is shown at FPIR = 10%. A resolution of 150×150 is used as the dataset's segmented resolution of 200×200 achieved the same performance.

The frontalization method was tested against VRA using *frontref* and *leftref* images for training, and Eigen was the best performing classifier for frontalization. LBPL results are shown for frontalization as it outperformed HE, HEL and LBPVL (to a lesser degree) on this dataset. Hence, parameters were retuned to determine whether LBPVL requires different parameters to those used in the validation dataset. However, it was found that the parameters reported in Section 6.3.2 remained optimal for both LBPL and LBPVL. Of note, when including 90° and 75° images, or when using VRA, LBPVL outperformed LBPL and the rest as was the case for the fingerprint. This suggests that texture distortion due to warping may be a possible factor for reducing LBPL's feature extraction performance (also negatively affecting the LBPH classifier). Table 6.6 shows the Eigen results for frontalization and VRA where LBPL and LBPVL feature extractors are used, respectively.

The frontalization method performed poorly compared to VRA for larger pose angles when using *frontref*. That is, both left and right poses of 75° classes were only occasionally correctly predicted due to minor overlap between the edge of *frontref* and extreme profile features. Since there is even less overlap between *frontref* and 90° features, only a couple of predictions were correct.

Applying the frontalization method to *leftref* simply did not take advantage of profile features without severe distortion. It is clear that VRA has a considerable advantage

over frontalization when using *leftref* to classify individuals posing at over 75° angles, with greater than 30% improvement overall. VRA also outperforms frontalization when using *frontref* albeit to a lesser extent due to the large amount of self-occlusion for profile features. The frontalization method has a *frontref* database image requiring no warping, but the input image requires a much larger warp than both VRA images combined, as shown in the visual validation results in Section 6.3.1.

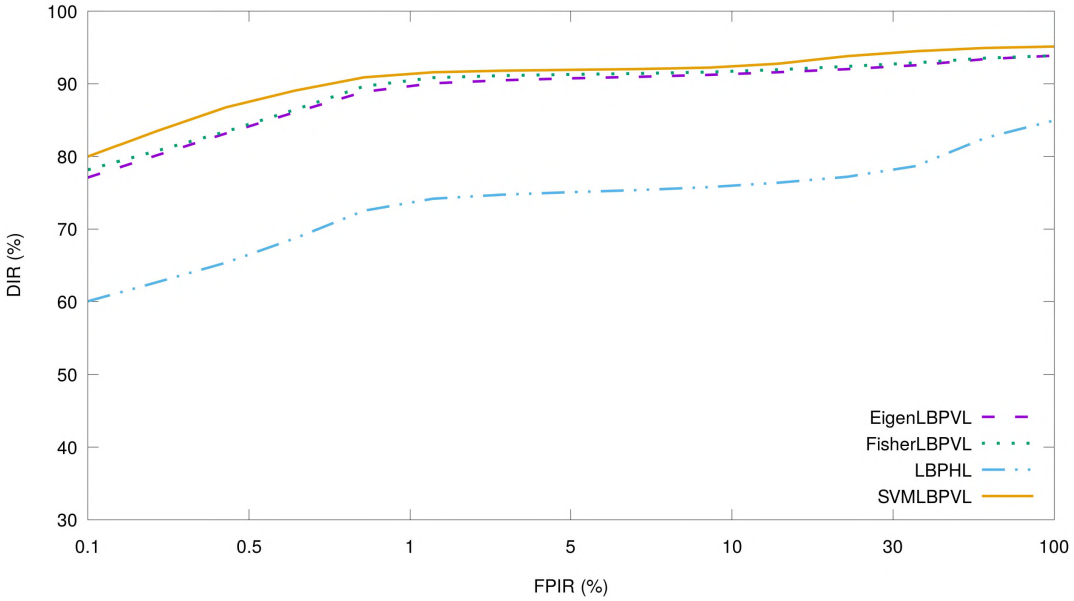
The VRA method yields promising performance for multi-angle pose datasets and is solely used in the subsequent face experiments. It should however be mentioned that LBPVL outperforms LBPL and HEL for all Fei dataset experiments when using VRA. The LBPL, HEL and other methods had a higher error rate when predicting the last sample, containing a variably lower amount of lighting than other samples, which confirms the superiority of LBPVL against various lighting conditions.

Table 6.6: Best classifier results at FPIR = 10%, for frontalization vs. VRA, when using one and three training samples.

Classifier	*frontref*	*leftref*	*frontref* VRA	*leftref* VRA
1-Eigen	48.1	45.8	60.8	78.6
3-Eigen	81.2	64.4	84.9	93.8

Referring to Figure 6.7, VRA uses the training sample(s) closest to the angle of the input image before warping. Therefore, since only one angle is used, Fisher cannot take advantage of multiple training samples at different angles. Unlike the fingerprint results, Eigen, Fisher and SVM all achieve markedly higher accuracies than LBPH. This could suggest that LBPH underperforms when small amounts of fine texture are common between the input and database images due to warping, and thus, prominent texture patterns such as the fingerprint are preferred. LBPH performs well for fingerprints due to its robustness to texture alignment, since similar texture patterns are merely distributed differently among intra classes. On the other hand, warping face images, due to different pose angles, can greatly alter fine texture patterns while aligning larger features such as eyes, nose and mouth. SVM achieves 95%, which is a slightly better result than Eigen and Fisher, but as seen in the figure LBPH is significantly outperformed.

Samet *et al.* (2016) performed closed-set identification for the first 80 subjects in the Fei



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0.2	6.2
3-FisherLBPVL	0	0	0	0	0.2	0.9	6.2
3-LBPHL	0	0	0.2	0.4	0.9	2.4	13.1
3-SVMLBPVL	0	0	0	0	0	0.4	4.9

(b) FNIR values

Figure 6.7: Three Fei training samples using VRA on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

face dataset on four poses (samples) and used the remaining six non-frontal samples for training to obtain their face model. The results illustrated in Table 6.7 show that with six training samples VRA achieves perfect closed-set identification accuracies. When training only the first frontal sample, SVMLBPVL accuracies for +45 and -45 drop to 98% and 99%, respectively.

6.3.4 PUT Identification Performance

This dataset has a segmented resolution of approximately 450×450. All classifiers achieved optimal performance between 150 × 150 and 250 × 250. Eigen, Fisher and SVM performance drops by 3% to 5% when using 100×100 and drops by approximately 7% at 50×50. On the other hand, LBPH drops are more substantial at approximately 20% and 35%,

Table 6.7: Comparative biometric closed-set identification accuracy (%) for four viewing angles selected from the Fei dataset.

Method	**+45°**	**+22.5°**	**−22.5°**	**−45°**
Samet *et al.* (2016) (Best)	79.1	94	95	81.6
Samet *et al.* (2016) (Traditional)	75	87	90	69
1-SVMLBPVL (VRA)	98	100	100	99
6-SVMLBPVL (VRA)	100	100	100	100

for respective resolutions. It is to be noted that LBPVL performed equally or slightly better (within 0.5%) than HEL. The tests were thus performed on the PUT dataset at a resolution of 150 × 150, using 0°, left angles 30° and 60°, and their combinations as training samples. The results are given in Table 6.8 at FPIR = 0%.

The trend in VRA performance on datasets up till now has been that using the smallest-difference angle yields superior results even when utilizing only a single but well-segmented database image as the training sample. Moreover, giving VRA three different pose angles for training allows it to achieve higher accuracies since it chooses the training sample with the nearest pose angle (within 10° tolerance). The trend continues for this dataset as VRA achieves a perfect accuracy when using 30° or three different pose angles for training. Of note, frontalization achieved its best result of 92.67% using Eigen. Since the nature of this dataset, that is the close proximity among pose angles, allows for VRA to produce multiple training samples, LBPH was retested using three training samples (per fixed angle) resulting in perfect accuracy. While this dataset is of significantly higher quality (resolution) than the other face datasets, this result is promising as achieving these accuracies involved performing ˜840000 matches when using a single training sample due to the large number of intra-class samples.

Table 6.8: IR at 0% FPIR and 0% FNIR when using different reference poses on the PUT face dataset.

Classifier	**0°**	**30°**	**60°**	**All 3**
1-EigenLBPVL	99.7	100	98.3	100
1-FisherLBPVL	99.7	100	98.1	100
1-LBPHL	96.9	96.7	93.2	99.9
1-SVMLBPVL	99.9	100	98.3	100

The proposed VRA approach and Malek *et al.* (2014) systems are compared on face

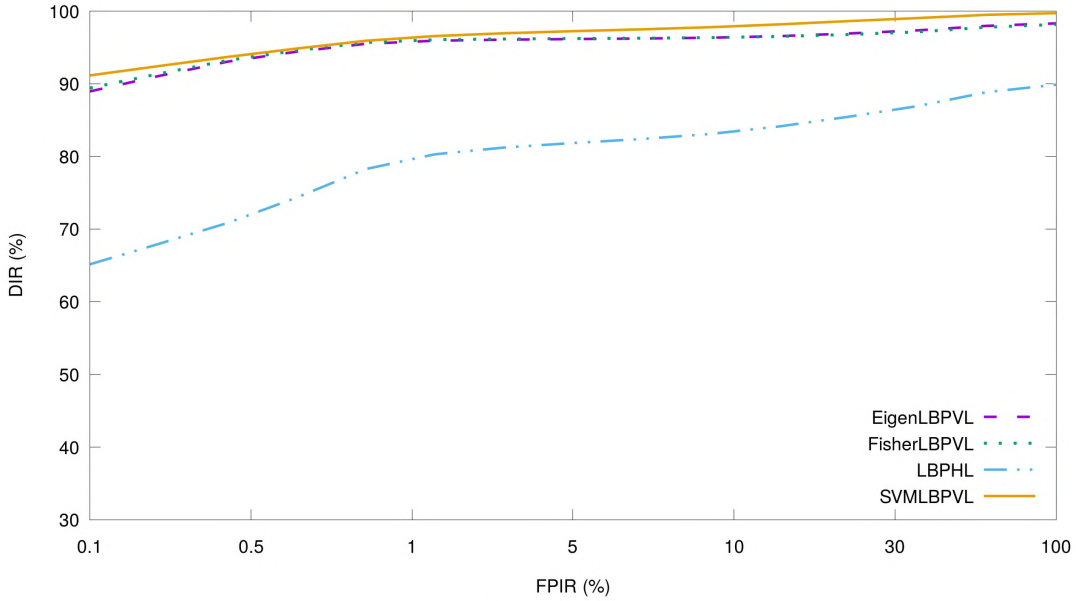
verification as no other studies were found on biometric authentication. Malek *et al.*'s experiment was replicated by using seven training samples on the first 20 individuals. The proposed VRA approach achieves zero EER in verification tests using any of the four classifiers and when using either one or up to seven training samples. This is, however, expected as it scored perfect accuracy for identification in the previous test on five times the number of classes. On the other hand, Malek *et al.*'s best system achieved 1.9% using a subspace state estimator classifier, which, for the interested reader, is explained in (Malek *et al.*, 2014). Furthermore, their Eigen classifier achieved an EER of 10.8%, highlighting the importance of accurate face segmentation.

6.3.5 SDUMLA Identification Performance

This dataset was used to evaluate the response of classifiers when face features are occluded by items such as glasses and hats or are substantially changed by facial expressions such as a gaping mouth. The pose angles used were constant at a near-frontal angle. VRA is still used, but the test images are simply warped according to the training images, instead of performing frontalization (or smallest-difference warping) on all training and test images, which means the training samples are unchanged. The segmented resolution of 200×200 yielded the best accuracies for all classifiers. A significant reduction in accuracy was observed using lower resolutions, similarly, for Eigen, Fisher and SVM – up to a 38% reduction in closed-set identification rate resulted from using a 50×50 image resolution. This reduction is, however, approximately 4% when FPIR $< 30\%$ for Eigen, Fisher and SVM. On the other hand, LBPH's DIR drops almost linearly across FPIR values at a 50×50 image resolution. The above trends are similar to those observed for the one and three training samples used in separate tests. Since only frontal faces were used, all three training samples were used as input for the classifiers.

The SVM again outperforms all the classifiers for both one and three training samples; see Figures 6.8b and 6.9b. However, the performance difference is more pronounced when using three training samples. Fisher marginally outperforms Eigen when using more

training samples, but both produce zero misclassifications as seen in Figure 6.9b. LBPH continues to be significantly outperformed by the other classifiers for face identification.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0	0	1.7
1-FisherLBPVL	0	0	0	0	0	0.2	1.9
1-LBPHL	0	0	0	0.2	0.4	0.9	4.4
1-SVMLBPVL	0	0	0	0	0	0	0.3

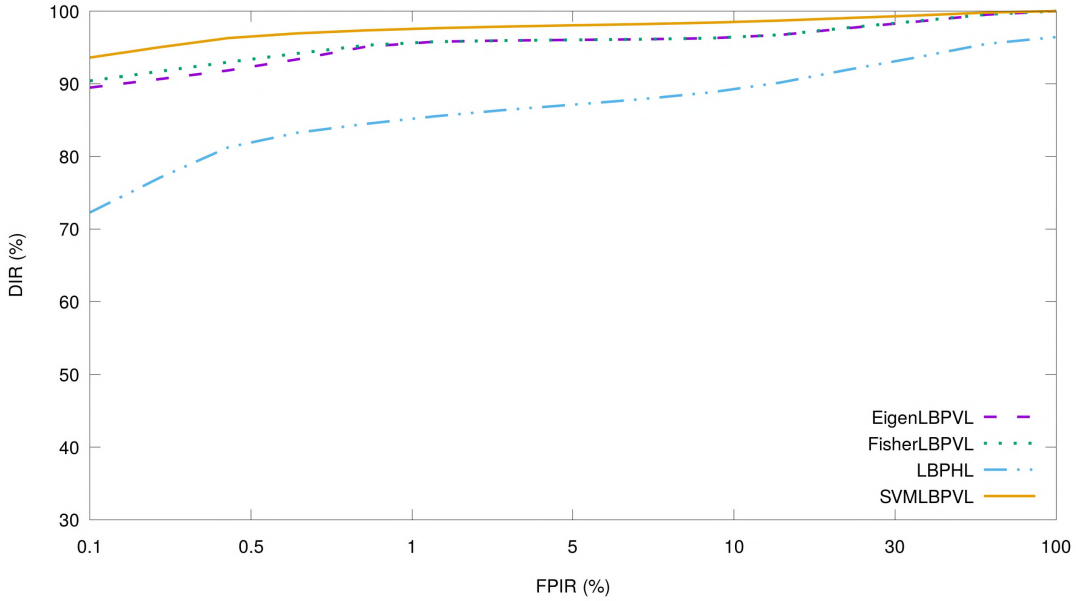
(b) FNIR values

Figure 6.8: One SDUMLA face training sample using VRA on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The closed-set identification accuracy of 106 classes from the SDUMLA face dataset was evaluated by Shams *et al.* (2016) using three frontal training samples and three frontal test samples. Their system is compared to the proposed approach in Table 6.9 where the proposed system only requires one training sample to achieve perfect accuracy, compared to their result of 89.8% when using three training samples.

6.3.6 Discussion of Face Results

LBPL marginally outperforms LBPVL on the FERET b-series dataset. The optimal parameters are identical to those for fingerprints for all methods except LBPH. In fact,



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	0
3-FisherLBPVL	0	0	0	0	0	0	0
3-LBPHL	0	0	0	0	0	0.9	3.6
3-SVMLBPVL	0	0	0	0	0	0	0

(b) FNIR values

Figure 6.9: Three SDUMLA face training samples using VRA on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

LBPH once again shows marked improvement with additional parameter tuning suggesting that it is most affected by different texture types of modalities and by the image acquisition procedure to a lesser extent.

The VRA face segmentation method outperforms frontalization in all tests and significantly when using a non-frontal image as a reference. The versatility of VRA is promising and is further assessed in Chapter 9.

SVM outperforms the other classifiers and scales well with more training samples. Fisher performs slightly better than Eigen when using VRA. Eigen is shown to be robust against the segmentation method used. The lighting normalization aspect of LBPVL appears to be superior to other methods due to its improved discrimination of local contrast. This is a promising result as it can help mitigate ill effects of dynamic lighting across multiple

Table 6.9: Comparative closed-set identification accuracy of the proposed approach using five-fold cross-validation on the SDUMLA face dataset (%).

Approach	(%)
1-SVM	100
3-SVM	100
Shams *et al.* (2016)	89.8

samples that current image-based biometric systems are often prone to.

The FNIR values (predicting the incorrect class) for SVM are zero on the SDUMLA face dataset, which means that only impostor errors occurred, contrary to the SVM results for the fingerprint. This can, however, be modality related or due to SDUMLA being a frontal dataset with facial expressions and occlusions, and thus we cannot confirm the reason at this time. Results are overall better than those for the fingerprint, which is probably due to the VRA segmentation method, especially considering the weak Fei face frontalization results, which are more similar to the results recorded for the fingerprint. However, this does not mean that the face is a better biometric modality than the fingerprint in general, as the face has a great weakness in terms of occlusions and makeup or changes over time such as facial hair and wrinkles or acute damage.

6.4 Summary

This chapter described the algorithms used to implement the proposed face recognition system. The face was initially detected in an approximate window, used as input for the facial landmark detection algorithm. The landmarks were keypoints used to obtain an accurately cropped ROI of the face. The novel VRA algorithm was developed to improve profile face segmentation such that less distortion occurred when aligning the database and input face images, thereby effectively reducing intra-class variance. The same postprocessing and classifier methods used for the fingerprint were applied and compared.

Classifier parameters were tuned on the popular FERET b-series dataset; LBPH was

found to require different parameters from those used for the proposed fingerprint identification system. Verification results were also included for the FERET b-series dataset as it is a popular dataset for verification. Eigen, Fisher and SVM slightly preferred images closer to the original resolution, but performed within the optimal range at 50×50 – the same resolution used for the fingerprint. This is important for compatibility when fusing biometric modalities for greater accuracy improvement. An exception to this was the closed-set identification accuracy on SDUMLA face, which performed poorly at lower resolutions. PUT face also dropped below optimal range at 50×50, but to a much lesser extent as results remained relatively high. However, when FPIR $< 30\%$, results were within optimal range in both cases suggesting that resolution only has a minor effect on impostor rejection. Results show that LBPH performed the weakest for open-set face identification, a result which is opposite to that obtained for the proposed fingerprint system. This noteworthy change in performance is further explored using the remaining two modalities.

Chapter 7

Palmprint Recognition Algorithms and Results

This chapter discusses the algorithms used to detect finger valleys as keypoints for palmprint biometrics. The resulting ROI obtained from the MIC method is shifted towards these keypoints to lower the inconsistency between intra-class samples of palmprints captured with low control. The evaluation of the methodology is validated and parameters are tuned, but less information is provided due to repeating results. The open-set identification accuracy is tested on three datasets and the results are discussed.

7.1 Aligning Palmprint Features

Since the orientation, scale and translation of the palm region varies more when the hand image is captured without guiding pegs, it is imperative that reliable keypoints are detected as effective input for the alignment procedure. This is particularly necessary when touchless hand sensors are used due to larger hand pose variations during acquisition. The alignment procedure thus uses the three valley points described in Section 3.2.3 to perform an affine transformation that aligns training and test images. The MIC method

in Section 3.2.3 is subsequently applied to the warped images by using the centre valley point as reference.

7.1.1 Presegmenting the Hand Image

The acquired hand image is first preprocessed by segmenting the hand from the background using Otsu thresholding (Otsu, 1979) followed by silhouette detection that finds extreme outer points on each side of the image.

7.1.2 Contour Detection

A contour is stored as a vector of points representing a curve in an image. The contour detection algorithm by Suzuki and Keiichi (1985) is adopted to store the outline of the white pixels of the binary hand image as a vector of 2D points. In the case where contour points form a straight line, only the end points are captured to reduce memory overhead. These contour points are connected by drawing lines between each of them. Additionally, the convex hull of the contour points is computed to find the extreme left, right, top and bottom contour points. The centre of these extremal points is a rough estimation of the palm centre. These extremal points are also used to segment the hand from the background in preparation for determining the ROI of the palmprint. The hand contours, convex hull and rough centre point are illustrated in Figure 7.1. This is used as the initial centre point for MIC detection in an effort to dramatically reduce the search space.

7.1.3 MIC detection

The MIC method is used so that both contact and contactless palmprints can be segmented regardless of constraining finger positions or whether fingers are even depicted within the image.

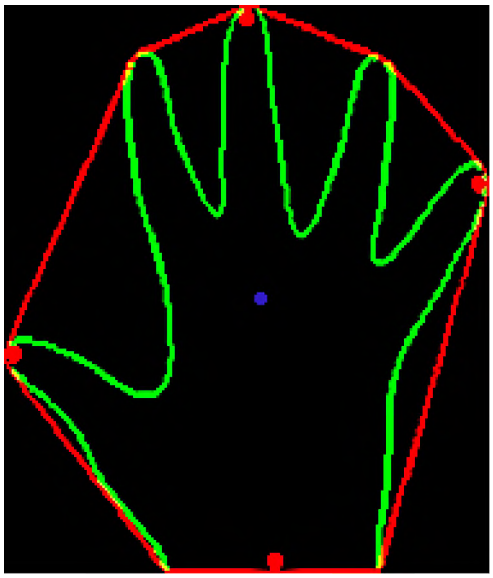


Figure 7.1: Segmentation approach of Zhang *et al.* (2003).

Shifting the MIC towards three valley points

Given the detected contour points the localization of finger valleys proceeds as follows. The starting point uses the nearest matching contour's y-coordinate with the MIC centre's y-coordinate. The contour points are sequentially traversed in a bottom-to-top manner, row by row, from left to right, while searching for an arc with a maximum end-to-end distance of $\frac{1}{8}$ of the segmented hand image, with a ± 45 orientation tolerance that caters for rotated hands. The detected arcs are assumed to be either finger tips or valleys. The three finger valley keypoints are distinguished from the finger tips by searching for those closer to the MIC centre.

The resulting MIC is shifted towards the middle valley point, similar to Ding and Ruan (2006)'s approach as explained in Section 3.2.3 and shown in Figure 3.14c.

7.2 Palmprint Segmentation

After obtaining the MIC, a minor modification is made to the MEC (Ding and Ruan, 2006) by shifting it by one pixel increment in a 10-pixel boundary until the index-middle,

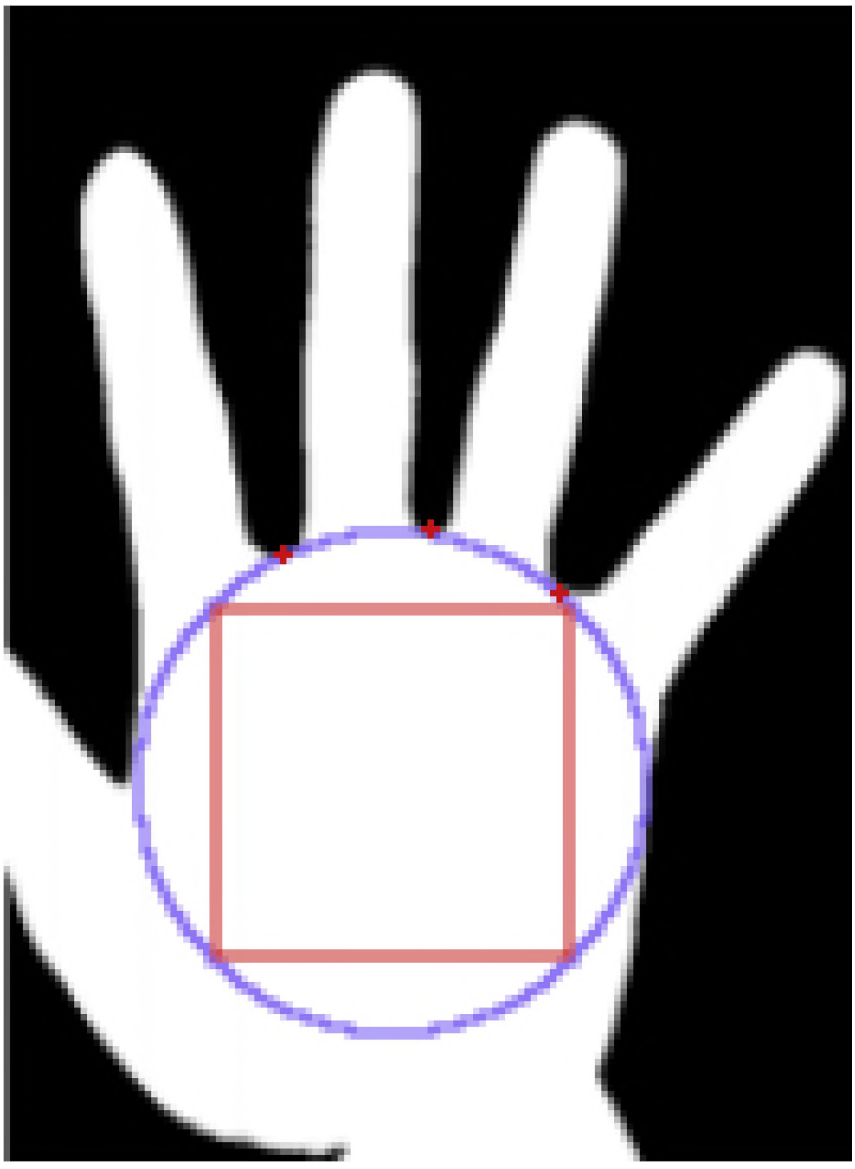
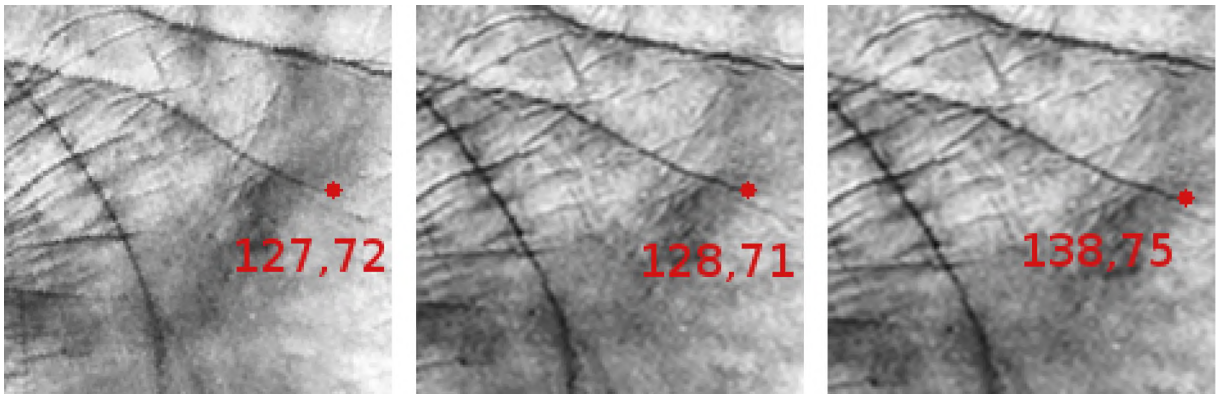


Figure 7.2: Square ROI based on three finger valley point MEC.



(a) Database palmprint image **(b)** Palmprint image using middle valley point **(c)** Palmprint image using three valley points

Figure 7.3: Improved palmprint alignment when using the modified MEC method.

middle-ring and ring-little finger valleys are equidistant from the circumference, as shown in Figure 7.2. This is in contrast to the original method (Ding and Ruan, 2006), that shifted the MIC towards the middle-ring finger valley. The final palmprint ROI is square shaped such that it circumscribes the circle with a $r\sqrt{2}$ length per side.

7.3 Palmprint Recognition Results

Methodology validation is carried out for the palmprint using visual inspection and empirical means. This is followed by open-set identification and verification experiments on three datasets.

7.3.1 Methodology Validation by Visual Inspection

Figure 7.3b shows the coordinate of the middle principal line ending of the database palmprint and two resulting input images for visual purposes. The segmentation consistency is improved when using the proposed method that consists of three valley keypoints compared with Ding and Ruan (2006)'s single keypoint approach.

7.3.2 Tuning of Experimental Parameters

The recognition performance of the proposed palmprint system was validated on the right palmprint of the CASIA-Palmprint dataset, similarly to fingerprint and face systems. A resolution of 50×50 was found to be optimal as for the fingerprint, which is perhaps attributed to their similar and sparse texture patterns. Of note, both the fingerprint and palmprint performance is increasingly sensitive to resolutions above 65×65 albeit the palmprint to a lesser degree with a 4% deficit at 100×100. This is attributed to the larger segmented resolution of 200×200 for this particular palmprint dataset. The optimal number of principal components remained 200 for Eigen and Fisher. The linear SVM at $C = 10^4$ was again found to be within optimal range. LBPH achieved an accuracy difference of only 1% when comparing LBPH against LBPHL regardless of the number of training samples used. The parameters were thus retuned, but were found to be optimal at the same values used for the face, with a 4-pixel radius and six neighbours. This means that the fingerprint prefers a sparser texture descriptor than either palmprint or face. As with the fingerprint, LBPH achieved superior accuracy compared to the other classifiers.

Table 7.1 summarizes the proposed system accuracies using the determined parameters. The LBPH classifier achieves the best results with a noticeably better closed-set identification result than the other classifiers for one and three training samples. Of note, the Eigen classifier utilizes additional training samples more effectively than the other classifiers, suggesting total variance to be relatively high. Lastly, Fisher scales the worst with additional training samples presumably due to some intra-class variance.

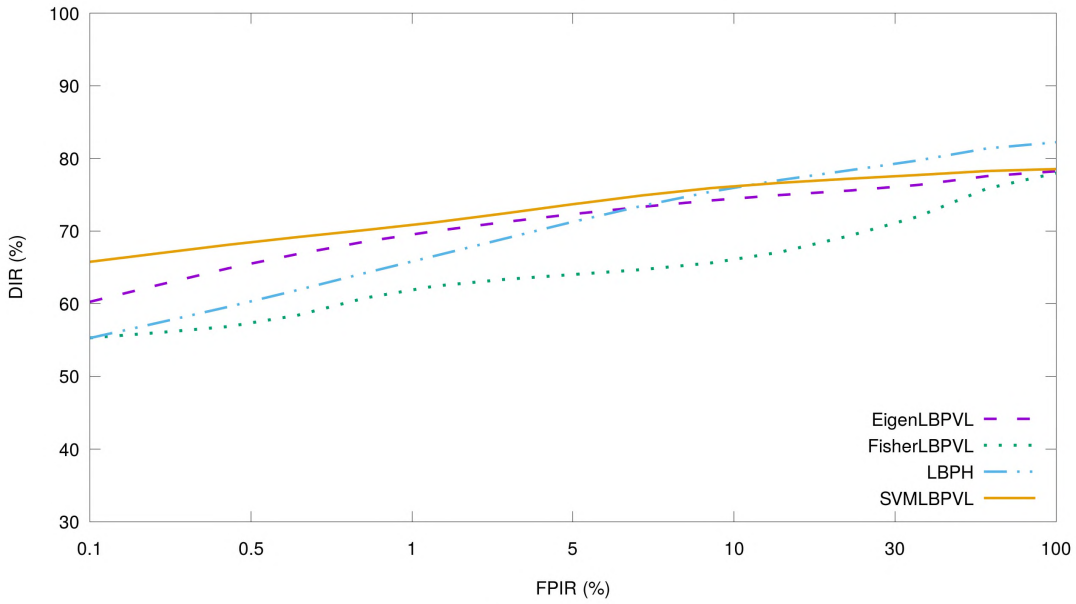
Table 7.1: IR when using 1, 3 and 5 training samples on the CASIA-Palmprint dataset (in %).

Classifier	1	3	5
EigenLBPVL	78.0	87.9	93.2
FisherLBPVL	78.3	82.5	85.6
LBPHL	85.4	92.5	95.8
SVMLBPVL	78.8	85.5	90.4

The next sections document three open-set identification experiments on different palmprint datasets.

7.3.3 CASIA-Palmprint Identification Performance

Referring to Figures 7.4 and 7.5, LBPH achieves better class identification, but at relatively poor FPIR values compared with the other classifiers. Eigen's performance scales the best on average when using three training samples and is only narrowly outperformed at FPIR = 100%. SVM also outperforms LBPH on average at FPIR < 100% and achieves maximum DIR identical to that of Eigen.



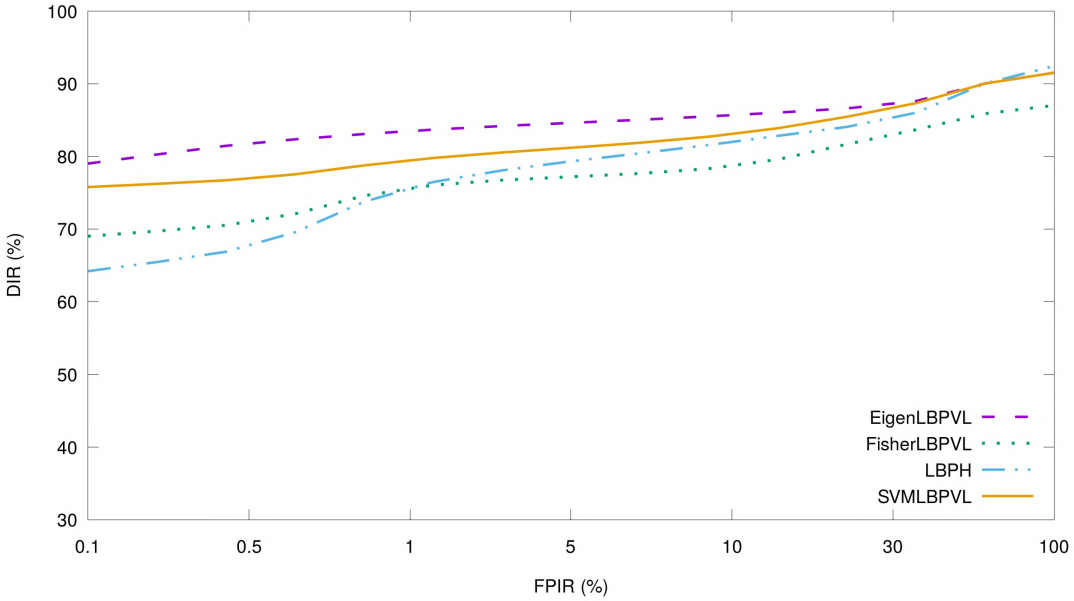
(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0.1	0.2	0.4	1.6	3.4	7.9	21.8
1-FisherLBPVL	0.1	0.3	0.4	1.6	3.6	8.0	22.1
1-LBPHL	0	0	0	0	0.5	2.3	17.8
1-SVMLBPVL	0	0	0	1.5	3.3	8.4	21.3

(b) FNIR values

Figure 7.4: One CASIA-Palmprint training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The palmprint verification results of Badrinath and Gupta (2011)'s system is compared with the proposed SVM method in a larger scale experiment. Since verification performance is tested, the left and right palmprints are combined without mirroring for 624 classes in total. Badrinath and Gupta's system and the proposed SVM method both achieve low EERs of 1.2% and 1.1%, respectively, as given in Table 7.2.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0.2	0.9	1.1	2.4	8.5
3-FisherLBPVL	0	0.1	0.2	0.9	2.9	6.2	13.0
3-LBPHL	0	0	0	0	0.1	0.9	7.6
3-SVMLBPVL	0	0	0	0.1	1.5	3.5	8.5

(b) FNIR values

Figure 7.5: Three CASIA-Palmprint training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

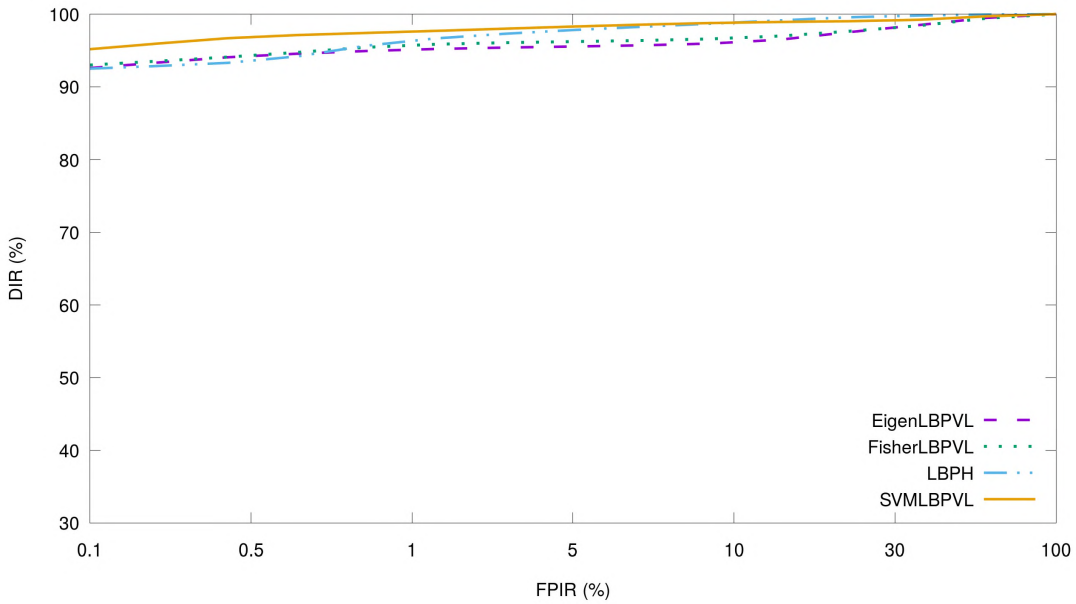
7.3.4 PolyU Identification Performance

The segmentation resolution for this dataset is approximately 130×130 and the optimal resolution remains 50×50. A 10% DIR deficit results from a 100×100 resolution. Classifiers perform similarly as the dataset samples are of consistent quality. The DIR values at 30% FPIR for Eigen, Fisher, LBPH and SVM are 98.1%, 98.1%, 99.7% and 99.1%, respectively, while FNIR (miss rate) is always zero. This is achieved using only a single training sample, with detailed results shown in Figure 7.6b.

Since all classifiers achieved a 100% DIR below 30% FPIR, DIR results at 0% FPIR and 0% FNIR are given in Table 7.3. SVM achieves the best results and it is apparent that LBPH does not scale as well as the other classifers when given more training samples.

Table 7.2: Comparative performance of the proposed approach for palmprint verification on the CASIA-Palmprint dataset.

Approach	EER (%)
2-SVM	1.1
Badrinath and Gupta (2011)	1.2



(a) DIR vs. FPIR

Classifier	**0.1**	**0.5**	**1**	**5**	**10**	**30**	**100**
1-EigenLBPVL	0	0	0	0	0	0	0
1-FisherLBPVL	0	0	0	0	0	0	0
1-LBPHL	0	0	0	0	0	0	0
1-SVMLBPVL	0	0	0	0	0	0	0

(b) FNIR values

Figure 7.6: One PolyU training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

Table 7.3: PolyU palmprint results at 0% FPIR and 0% FNIR using three training samples.

Classifier	**DIR**
3-EigenLBPVL	97.0
3-FisherLBPVL	97.7
3-LBPHL	95.2
3-SVMLBPVL	98.7

Zhang *et al.* (2016)'s palmprint system is compared with the proposed palmprint system by measuring verification performance on the 250 individuals of the PolyU palmprint

dataset. Since the number of training images are not specified in the study, only the first sample is used for this purpose. Furthermore, only the first session, consisting of six samples, are used in the test. Zhang *et al.*'s approach yields a low EER of 0.0257%, while all proposed approaches result in zero error.

7.3.5 IITD-Palmprint Identification Performance

This challenging dataset favours user convenience as it was captured with a touchless hand sensor. As shown in Figures 7.7 and 7.8, LBPH significantly outperforms the other proposed methods for one training sample. When using three training samples, LBPH and Eigen improve by 10% and 13%, respectively. Furthermore, LBPH achieves better DIR values than Eigen by at least 9% followed by SVM and lastly Fisher.

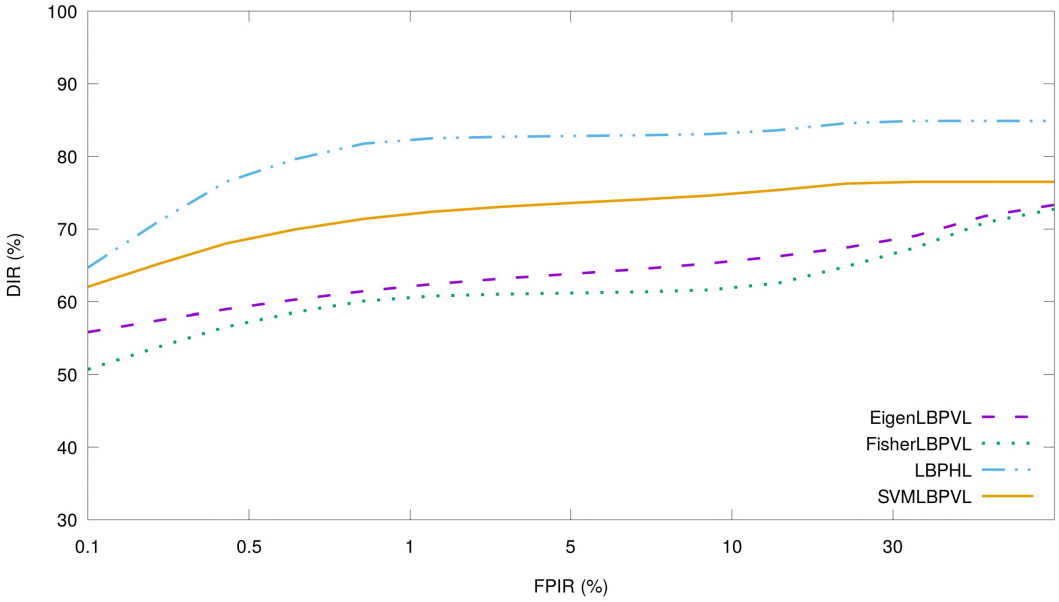
Morales *et al.* (2011) compared their system with the IITD dataset authors' (Kumar, 2008) approach using five training samples and only a single test sample, per 235 subjects. The relatively large number of training samples allow both systems to achieve low EERs. The LBPH version of the proposed system performs comparably with the non-texture-based system of Morales *et al.*, as shown in Table 7.4.

Table 7.4: Comparative performance of the proposed approach for palmprint verification on the IITD-Palmprint dataset.

Approach	EER (%)
Proposed 5-LBPHL	0.23
Morales *et al.* (2011)	0.21
Kumar (2008)	1.31

7.3.6 Discussion of Palmprint Results

LBPH achieves the best DIR when using a single training sample, but at a relatively high FPIR. The SVM also achieves a high DIR for a single training sample and often achieves the lowest FPIR. When specifically considering datasets with high intra-class variations, Fisher achieves bad results for the fingerprint, face and palmprint. The SVM does not



(a) DIR vs. FPIR

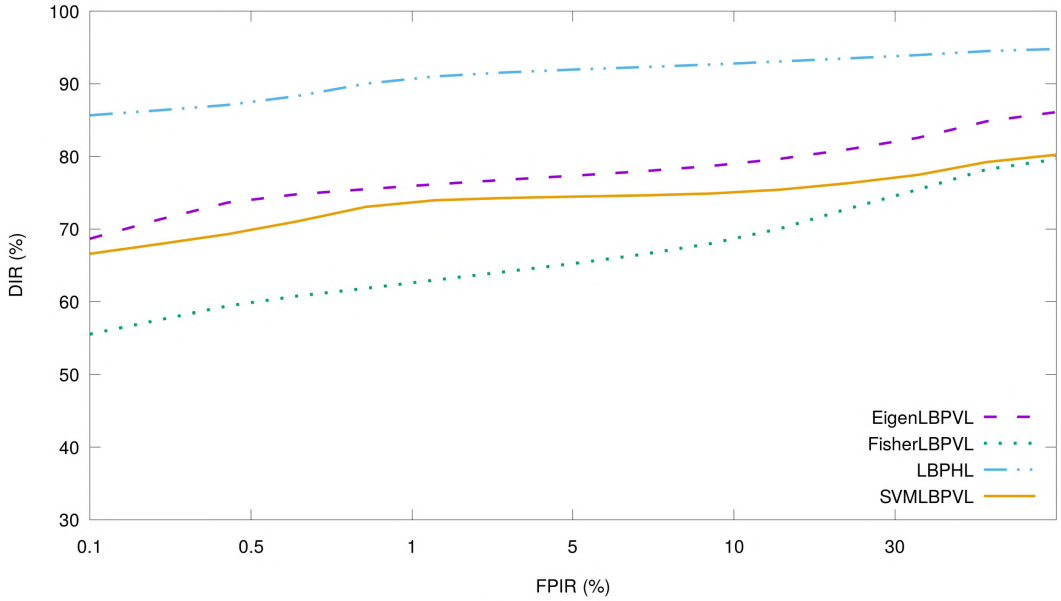
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0.2	0.8	1.8	3	7.8	26.7
1-FisherLBPVL	0	0	0.8	1.8	3.2	8.2	27.3
1-LBPHL	0	0.2	0.2	0.2	1	3.5	15.2
1-SVMLBPVL	0	0.5	2.8	3.2	4.3	8.7	23.5

(b) FNIR values

Figure 7.7: One IITD-Palmprint training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

scale as well as Eigen and LBPH with more training samples, but it is consistent since it does not deviate much from the trends for other modalities. On the other hand, when considering datasets with low intra-class variations, SVM and Fisher scale well with more training data.

The IITD test results show that LBPH performs relatively well with intra-class variations caused by alignment issues. However, another reason may be that the IITD dataset contains the most well-defined sparse texture in terms of principal lines and other texture patterns as it has the highest capture resolution, while PolyU has the lowest. Therefore, the high accuracy of LBPH may be attributed to its ability to aggregate those features more effectively than the other classifiers as was the case with the fingerprint.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	1	2.5	4.5	4.9	13.9
3-FisherLBPVL	0.5	0.5	0.5	1	3.0	6.5	20.4
3-LBPHL	0	0	0	0.5	0.5	0.5	5.3
3-SVMLBPVL	0	0.2	0.7	3.5	4.8	5.2	19.8

(b) FNIR values

Figure 7.8: Three IITD-Palmprint training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

7.4 Summary

Implementation of the palmprint recognition system was described in this chapter. The hand was first presegmented to remove background noise. An approach based on Ding and Ruan (2006)'s work was used to detect an improved MEC, constituting the palmprint. The proposed approach used an extra two finger valleys adjacent to the middle finger, and was validated to be more accurate at aligning the MEC for robustness to hand poses of palmprint images in less constrained conditions. The rest of the process followed the same postprocessing and classifier methods as for the fingerprint and face modalities.

Classifier parameters were tuned on the popular CASIA-Palmprint dataset. LBPH results were optimal when using the same parameters as for the face identification system, but

the addition of a LoG filter only improved the accuracy by 1% on the validation dataset. On the other hand, the same 50×50 used for the fingerprint was found to be optimal for the palmprint when using Eigen, Fisher and SVM. Results reveal that SVM and Fisher scaled well with more training samples on datasets containing low intra-class variations, whereas Eigen and LBPH scaled well regardless.

Chapter 8

Iris Recognition Algorithms and Results

This chapter discusses the algorithms used to detect the pupil and iris boundaries as keypoints. The Hough method is improved for consistent iris segmentation and robustness to occlusions caused by squinting and off-centre eye gazing. The evaluation of the methodology is validated and parameters are tuned. Open-set identification accuracy is tested on three datasets and the results are discussed.

8.1 Aligning Iris Features

The HT method by Wildes (1997), as discussed in Section 3.2.4, is improved to increase the robustness when segmenting the iris prior to classification. The improved process, which is set out in the proposed iris biometric implementation, proceeds as follows.

1. Candidate pupil regions are detected by restricting the HT to approximate circles of a pixel-sized radius of $r[15, 45]$, determined empirically for periocular regions. The parameters used for the HT are a Canny edge threshold, $Th = 100$, and a minimum circle centre inter-distance of 10 pixels.

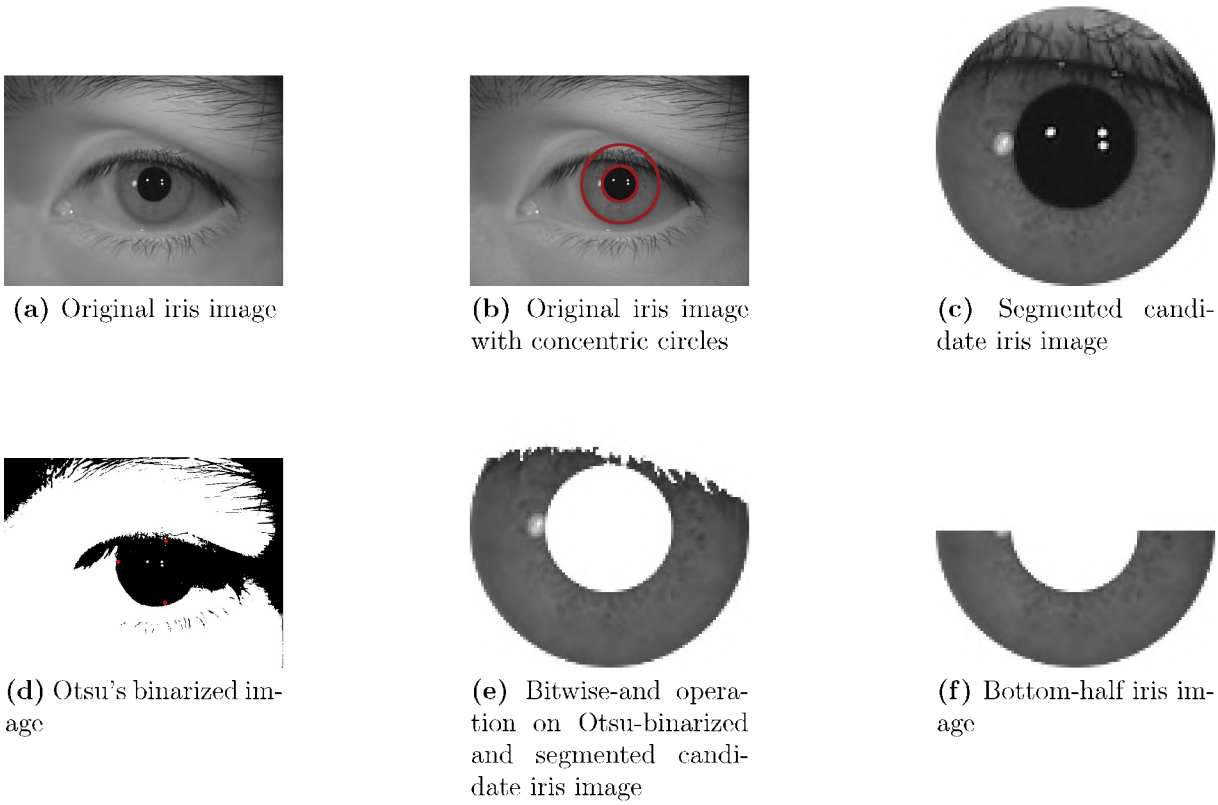
The start of the proposed approach is in contrast with the original HT method, which first detects the iris followed by the pupil. The reverse order is proposed as it is assumed the pupil is easier to detect. This is because when overlapping circles are found, the smaller one with higher contrast is selected as it is assumed the pupil is always smaller and darker than the iris.

2. An approximate concentric circle (candidate iris), with a 3-pixel tolerance for the common centre coordinate, is sought using the same parameters, except with a radius range of $r[30, 80]$, instead of $r[15, 45]$. The candidates that fit within the periocular region are selected. This reduced the number of candidate irises to one in more than 90% of the images when carried out on the right iris for each individual in the CASIA-Iris-Lamp dataset. The candidate iris is highlighted in Figure 8.1b.

3. Segmentation of the outer region without considering eyelid boundaries produces the image in Figure 8.1c. In an effort to counteract occlusions and other inconsistencies, the rest of the process follows a new approach on obtaining the best iris candidate with no known commonalities with the original HT method.

4. The eyelid boundaries are determined by contrasting the sclera and the iris region vertically and horizontally to the first connected pixel found from the centre of the pupil, using Otsu's image binarization discussed in Section 3.1.3 to produce Figure 8.1d. This removes non-iris pixels to better approximate the amount of occlusion and lower intra-class variations when dealing with non-ideal iris image acquisition such as long eyelashes, squinting and off-centre eye gazing. The candidate that falls outside the boundaries by the least number of white pixels is selected as the best candidate, as shown in Figure 8.1e. This new approach continues by improving the consistency of the best iris candidate ROI using the bitwise-and operation on the binarized image and the candidate iris.

5. The resulting image is often found to have occlusions on the top half of the iris due to eyelid positioning. The bottom half of the result is thus used as the final segmented iris. This step further deals with squinting and off-centre eye gazing. The final iris segmentation result is shown in Figure 8.1f.



(a) Original iris image

(b) Original iris image with concentric circles

(c) Segmented candidate iris image

(d) Otsu's binarized image

(e) Bitwise-and operation on Otsu-binarized and segmented candidate iris image

(f) Bottom-half iris image

Figure 8.1: Iris image segmentation procedure.

This modified HT method succeeds when a pair of eyes is horizontal within 15°. The pruned concentric circle is the final segmented iris, which is subsequently used as input for postprocessing and classification.

8.2 Iris Recognition Results

Methodology validation is carried out for the iris using visual inspection and empirical methods, followed by open-set identification and verification on three datasets.

8.2.1 Methodology Validation by Visual Inspection

Figure 8.2d shows the result of removing eyelids present in Figure 8.2b. While more features are visually present before eyelash and eyelid removal, the conflicting pixels in

(a) Original iris image.

(b) Segmented candidate iris image.

(c) Bitwise-and operation on Otsu-binarized and segmented candidate iris image

(d) Bottom-half iris image

Figure 8.2: Iris image segmentation procedure on slightly closed eye.

Figure 8.2b (also Figure 8.1c) can increase intra-class variation, and are not examples of unique features that define a class.

8.2.2 Tuning of Experimental Parameters

The right iris for each individual in the CASIA-Iris-Lamp dataset comprises the validation set. The blank pixels (top half) of the bottom-half iris image are given a zero value so that the resolution remains the same, for compatibility with the fused datasets (introduced in the next chapter). An image resolution of 50×50 for Eigen, Fisher and SVM resulted in a 15% lower accuracy on average, using one training sample. Since all three classifiers were negatively affected to a similar degree, a segmented resolution (100×100 for this dataset) was used in the experiments. The number of principal components remained at 200 for Eigen and Fisher, while the linear SVM results also remained consistently within optimal range with $C = 10^4$. For LBPHL, three neighbours always achieved the best

results regardless of the radius value. Furthermore, only the LoG filter results (LBPHL) are shown because its inclusion improved results by about 10% at the best parameter values. The bottom-half iris results are shown for varying radius values when using three neighbours in Table 8.1. Results are similar for the radius values but a 2-radius value yielded the best average accuracy and is thus used in subsequent experiments.

Table 8.1: Selecting the optimal radius (1–6) for LBPHL using three neighbours.

Classifier	1	2	3	4	5	6
1-LBPHL	71.6	72.8	71.8	72.8	72.9	66.8
3-LBPHL	90.6	90.2	90.6	90.2	89.0	88.8
5-LBPHL	94.8	94.8	93.7	94.8	93.1	92.2

Parameter tuning was also performed on the full irises, but required no changes compared to the bottom-half iris parameters. An accuracy improvement of approximately 10% resulted from the use of bottom-half iris segmentation over the full iris. Although it may not appear to be prudent to discard half the iris, comparing Tables 8.3 and 8.2 shows that using the bottom-half iris images produces superior results and thus this approach is henceforth adopted.

Table 8.2: IR for the full iris on the CASIA-Iris-Lamp dataset.

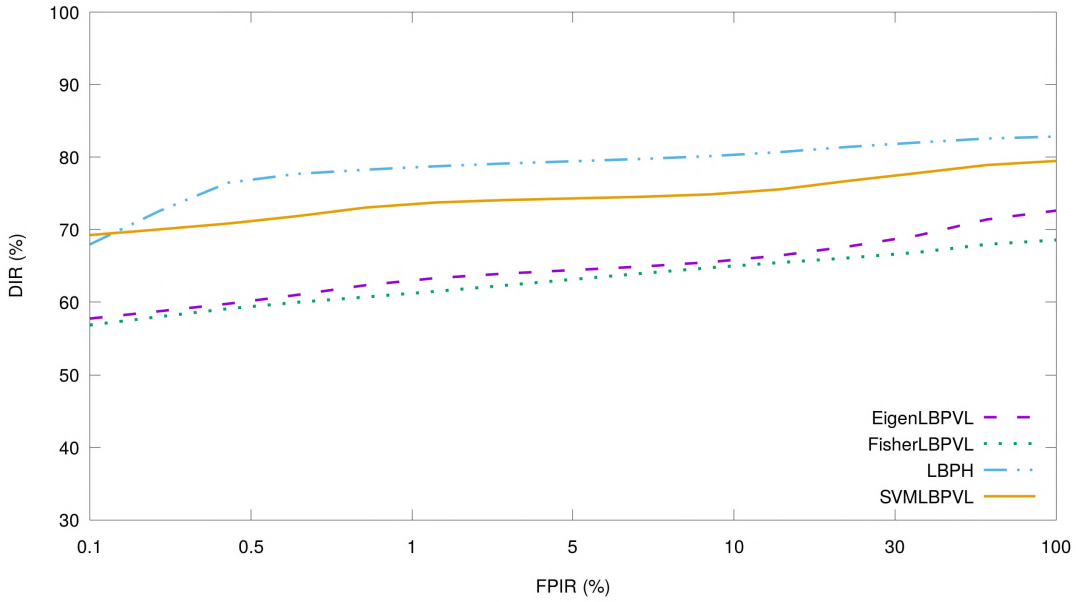
Classifier	1	3	5
Eigen	53.2	85.4	90.7
Fisher	49.3	77.0	70.8
LBPH	68.9	88.2	89.4
SVM	52.2	80.3	84.8

Table 8.3: IR for the bottom-half iris on the CASIA-Iris-Lamp dataset.

Classifier	1	3	5
Eigen	64.4	93.6	96.2
Fisher	59.5	85.0	82.4
LBPH	72.8	90.2	94.8
SVM	65.2	92.0	96.0

8.2.3 CASIA-Iris-Lamp Identification Performance

Referring to Figures 8.3 and 8.4, LBPH does not scale as well beyond one training sample when compared with other classifiers, which was also observed in the results of the previously tested modalities. However, it outperforms Eigen and Fisher significantly, which may be attributed to the intra-class variations caused by the highly disparate pupil sizes in this challenging dataset. Again LBPH shows superior robustness to intra-class variations compared with the other classifiers. Regardless, SVM outperforms LBPH for three training samples in terms of DIR due to the weak scaling of multi-sample training data using LBPH. SVM has up to 3% more misclassifications than LBPH for three training samples.

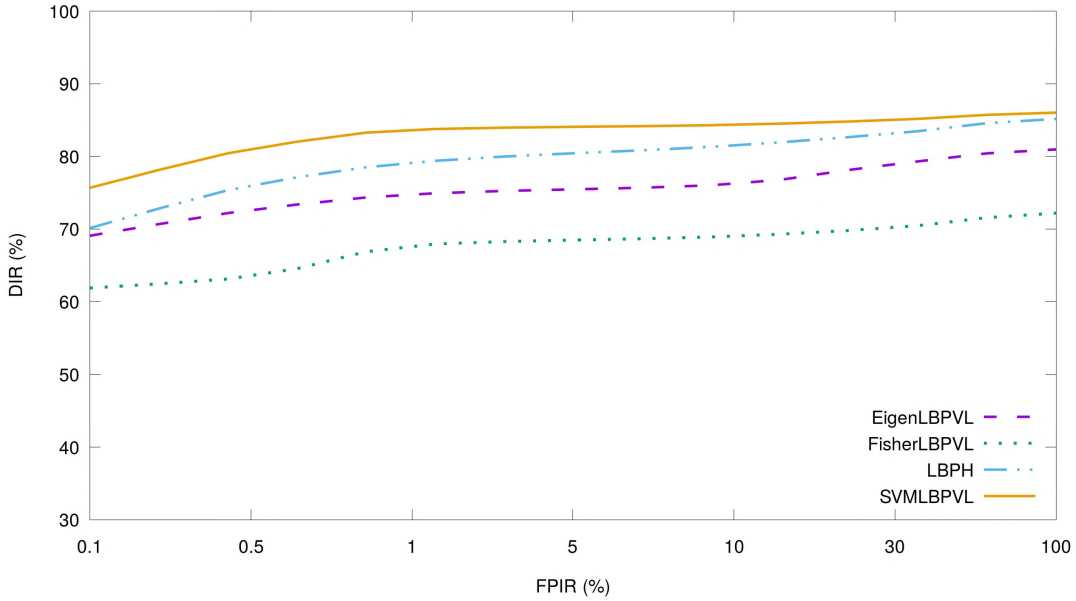


(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0.2	0.4	0.8	2.9	9.9	27.4
1-FisherLBPVL	0	0.3	0.4	0.9	3.2	11.5	31.6
1-LBPHL	0	0	0.9	1.2	2.2	6.1	17.2
1-SVMLBPVL	0	0	0.5	1.1	3.7	7.2	20.6

(b) FNIR values

Figure 8.3: One CASIA-Iris-Lamp training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	1.2	2.1	3.9	19.0
3-FisherLBPVL	0	0	1.8	3.9	4.6	5.3	27.8
3-LBPHL	0	0	0	0	0.8	3.1	14.9
3-SVMLBPVL	0	0	0	0	0.7	3.2	13.9

(b) FNIR values

Figure 8.4: Three CASIA-Iris-Lamp training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

Podder *et al.* (2015) do not provide information on the setup of classes and samples of their iris verification system. For comparative purposes, the proposed system is tested on the left iris of 411 individuals. Results are provided using one or three training samples, with the remaining samples for testing. The proposed system is outperformed by Podder *et al.*'s system by a margin of approximately 0.5% EER, when using a single training sample. However, the proposed system yields better performance, by a slightly larger margin, when using three training samples, as shown in Table 8.4.

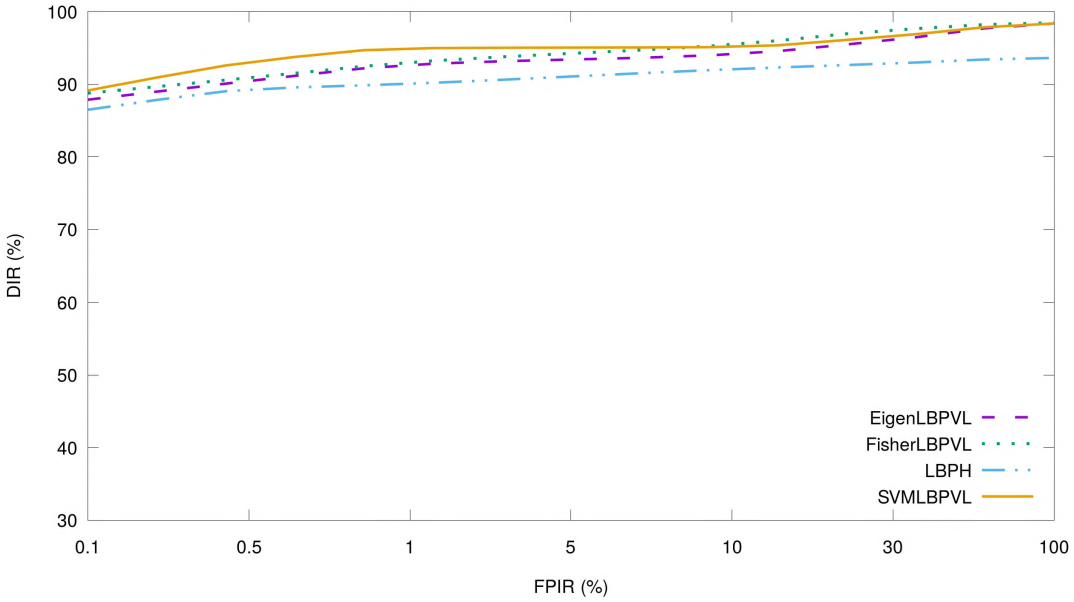
8.2.4 IITD-Iris Identification Performance

The same resolutions used for CASIA-Iris-Lamp, per classifier, remained optimal for the IITD-Iris dataset even though it has double the segmented resolution. This is due to the

Table 8.4: Comparative performance of the proposed approach for iris verification on the CASIA-Iris-Lamp dataset.

Approach	EER (%)
1-LBPHL	1.28
3-LBPHL	0.33
Podder *et al.* (2015)	0.82

use of a sparse (cheap) and relatively old capturing sensor. All classifiers achieve high accuracy as shown in Figure 8.5. Using one training sample, the SVM outperforms all classifiers, except where Eigen and Fisher surpass the SVM for FPIR $\geq$ 20%. LBPH achieves a relatively low DIR above 10% FPIR.



(a) DIR vs. FPIR

Classifier	**0.1**	**0.5**	**1**	**5**	**10**	**30**	**100**
1-EigenLBPVL	0	0	0	0.4	0.4	0.5	2.6
1-FisherLBPVL	0	0	0	0.3	0.3	0.3	2.5
1-LBPHL	0	0	0	0.2	0.8	1.7	4.2
1-SVMLBPVL	0	0	0	0	0.5	1.3	2.6

(b) FNIR values

Figure 8.5: One IITD-Iris training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

When using three training samples, all classifiers achieve a 100% DIR below 30% FPIR, and thus, DIR results at 0% FPIR and 0% FNIR are given in Table 8.5. Fisher scales the

best and achieves the highest DIR. LBPH achieves the lowest accuracy for this dataset regardless of the number of training samples. SVM achieves the best results and it is apparent that LBPH does not scale as well as the other classifiers with more training samples.

Table 8.5: IITD-Iris recognition rate at 0% FPIR and 0% FNIR using three training samples.

Classifier	**DIR**
3-EigenLBPVL	97.8
3-FisherLBPVL	98.9
3-LBPHL	96.2
3-SVMLBPVL	97.1

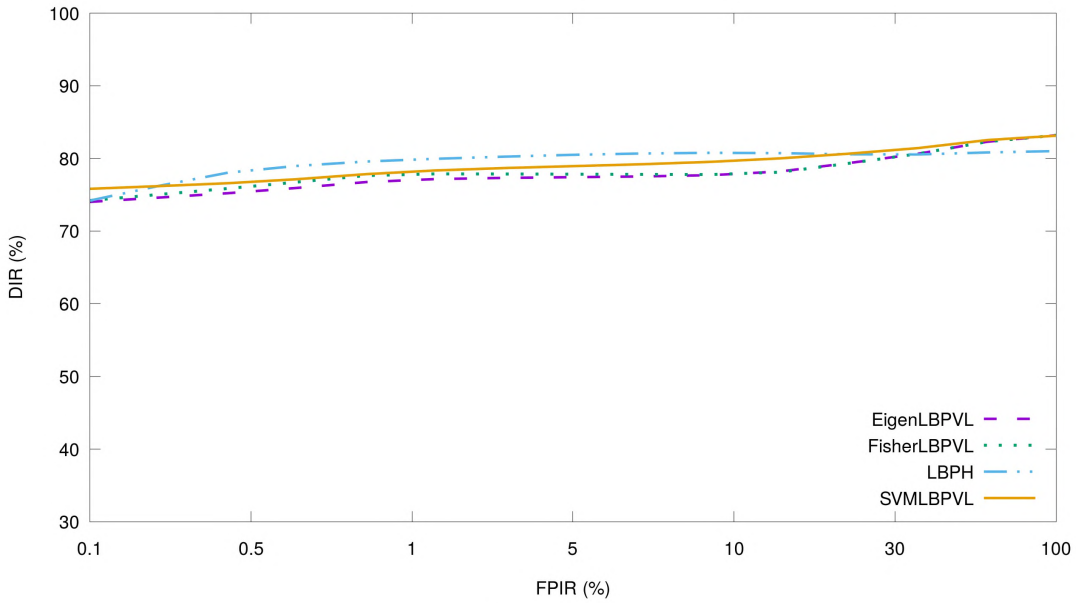
Umer *et al.* (2015) provide test results for their iris verification system using the leave-one-out method (5-fold cross validation). That is, four training samples are used with only a single test sample, per 224 subjects. This system is compared with Zhou and Kumar (2010)'s system, also using the leave-one-out method. Umer *et al.* show that their system achieves a 0.5% improvement in GAR over the original HT method by Wildes (1997). The results for the two systems developed by Umer *et al.* and Zhou and Kumar show EERs of 0.015% and 0.530%, respectively. The proposed SVM approach, however, achieves 0.500% using only a single training sample on four test samples, and achieves perfect accuracy when using more than one training sample or 5-fold cross validation (see Table 8.6).

Table 8.6: Comparative performance of the proposed approach for palmprint verification on the IITD-Iris dataset.

Approach	(EER %)
1-SVMLBPVL	0.5
2-SVMLBPVL	0
Zhou and Kumar (2010)	0.015
Umer *et al.* (2015)	0.530

8.2.5 SDUMLA Iris Identification Performance

For this dataset all classifiers perform similarly, however, LBPH performs better at lower FPIR, while Eigen and Fisher have the edge at higher FPIRs. The rest of the observations correspond to the findings on the IITD-Iris dataset.



(a) DIR vs. FPIR

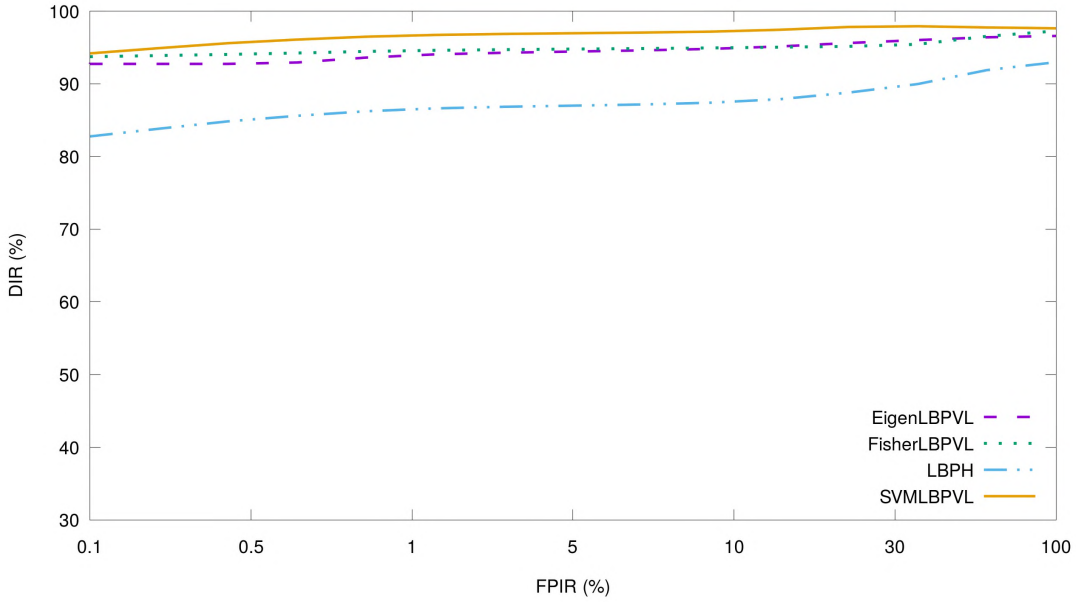
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0.2	0.4	1.2	1.4	4.3	16.9
1-FisherLBPVL	0	0.2	0.4	0.9	1.4	4.6	16.9
1-LBPHL	0	0	0.7	1.8	5.7	8.3	19.0
1-SVMLBPVL	0	0.1	0.4	2.2	1.5	4.9	16.9

(b) FNIR values

Figure 8.6: One SDUMLA iris training sample on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

The previously observed trend where LBPH scales relatively poorly with more training samples continues in Figures 8.6b and 8.7b. However, in this experiment the effect is even more pronounced when this happens. The rest of the classifiers perform similarly with Fisher, scaling slightly better, which concurs with the findings on the IITD-Iris dataset.

The results of Vishi and Yayilgan (2013) are compared with those of the proposed system using the left iris of the first 100 individuals and five samples. Vishi and Yayilgan do not



(a) DIR vs. FPIR

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0.1	0.9	3.5
3-FisherLBPVL	0	0	0	0	0	0.3	2.8
3-LBPHL	0	0	0	0.5	0.4	1.1	7.1
3-SVMLBPVL	0	0	0	0	0	1.2	2.4

(b) FNIR values

Figure 8.7: Three SDUMLA iris training samples on remaining test images: (a) DIR versus erroneously identified impostors (FPIR); (b) FNIR for given FPIR values.

state the training and test sample ratio. The results shown for the proposed SVM system were obtained with one or three training samples, with the rest used as test samples. The proposed system is outperformed by Vishi and Yayilgan's system by a margin of approximately 0.5% EER, when using a single training sample. However, the proposed system yields better performance, by a slightly larger margin, when using three training samples as shown in Table 8.7.

Table 8.7: Comparative performance of the proposed approach for palmprint verification on the SDUMLA-Iris dataset.

Approach	EER (%)
1-SVMLBPVL	3.8
3-SVMLBPVL	1.3
Vishi and Yayilgan (2013)	3.3

8.2.6 Discussion of Iris Results

The previously identified trend of Eigen and SVM performing consistently well continues for iris biometric systems, with the exception of the experimentation on the CASIA-Iris-Lamp dataset using one training sample. Moreover, LBPH appears to perform relatively well on non-ideal data, but does not improve significantly with more training samples. SVM is the best performer on average as it performs consistently well under all conditions. The results on the IITD and SDUMLA iris datasets are promising as they show that identification systems can achieve near-perfect accuracies using any of the four classifiers.

Considering the results of all modalities, SVM appears to yield higher FNIR values than the other classifiers. However, this can be attributed to using the same parameters for all modalities and also due to omitting the SVM kernels such as the radial basis function. The face and iris datasets achieved better accuracies when using the segmented resolution compared with 50×50, while fingerprint and palmprint preferred lower resolution images. Eigen, Fisher and LBPH perform well at higher FPIR values, which is good for closed-set identification systems. On the other hand, SVM performed well on open-set identification, while performing comparably to the other classifiers on closed-set identification systems.

Results from LBPVL are shown in the majority of the experiments as this method produced superior performance compared with the other feature extraction methods. Whether this is also the case for the fused datasets introduced in the next chapter, remains to be seen.

8.3 Summary

This chapter discussed the implementation of an iris recognition system. The iris key-points were determined using a novel HT-based approach, with additions that improve robustness against occlusions caused by eye lashes, squinting and off-centre eye gazing. Classifier parameters were tuned on the CASIA iris dataset. The bottom half of the iris was shown to achieve better results than using the whole iris, which was attributed

to performance degradation due to occlusions such as upper eyelids and eyelashes. The bottom-half iris was thus used in the experiments, which classified the results after applying the same postprocessing techniques as the systems implemented for the other three modalities. Although the 50×50 resolution previously used for fingerprint, face and palmprint, was found to be within optimal range when using Eigen, Fisher and SVM, an approximate 4% accuracy improvement was noted when using the native resolution of iris images. A similar effect was noted for the face results. LBPH results were optimal when using three neighbours, which is significantly fewer than for the other modalities indicating that relevant features are tightly spaced in the iris.

Results revealed that SVM and Fisher scaled well with more training samples on datasets containing low intra-class variations, whereas Eigen scaled well regardless. SVM also outperformed the other classifiers on average and appeared to perform well with the tightly spaced features present in both the face and iris.

Chapter 9

Bimodal Feature Fusion and Recognition Results

This chapter presents the results of multimodal biometric systems created from pairs of modalities combined at the feature level. Modalities are combined in two different ways and applied to the same datasets used in the individual biometric modality experiments. Open-set identification accuracy is evaluated on the fused bimodal combinations. A discussion of the results concludes the chapter.

9.1 Multimodal Biometric Fusion at Feature Level

Two image-based biometric feature-fusion methods, namely feature averaging (FA) and feature concatenation (FC) are used to investigate fusion at the feature level. To implement this fusion, each image of a modality is represented as a 1D feature vector by outputting the pixel values to a text file similar to the method used in the individual modality experiments. The two feature vectors are subsequently combined either by averaging their values, which leads to a vector of the same size, or concatenating the second feature vector to the end of the first feature vector to form an extended vector. In the image space, feature concatenation corresponds to appending the second biometric modality

to the top-right coordinate of the first biometric modality. Feature fusion only takes place after the postprocessing step that outputs feature vectors in Section 4.1.3. However, to emphasize the differences between the two fusion methods, they are applied on images without extracted features to "visualize" how feature vector fusion works specifically on image-based modalities.

9.1.1 Feature Vector Averaging

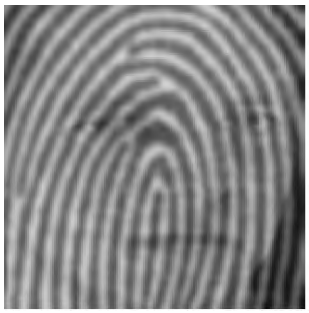
Since the respective pixel values in each of the pair of modalities are averaged, as shown in Figure 9.1a, the dimensionality of the feature vector does not increase, but instead mixing of the texture pattern occurs. This does however, result in some loss of information from each modality. Furthermore, combined texture patterns can reduce the benefits of this type of fusion in some cases. For example, when a group of pixels constituting a unique texture pattern in a single modality is averaged with a texture pattern of another single modality, a new blended texture pattern is formed that may not have any uniqueness aspect. This is also known as parallel fusion (Yang *et al.*, 2003).

9.1.2 Feature Vector Concatenation

This fusion method appends features of the second modality horizontally to the first modality, effectively doubling the dimensionality as shown in Figure 9.1b. No loss of information in either modality occurs as the texture patterns of the individual feature vectors remain unchanged and are complementary. The figure shows the two segmented modalities before feature extraction for visual purposes. This is also known as serial fusion (Yang *et al.*, 2003).

9.2 Preliminary Experiments

Preliminary experiments were first conducted to compare the proposed fusion approach with the related studies discussed in Section 3.5.5. Due to the lack of studies conducted on



(a) Fingerprint and face fusion using feature vector averaging

(b) Fingerprint and face fusion using feature vector concatenation

Figure 9.1: Visual representation of two feature-fusion methods.

disclosed/publicly available datasets, only the experiments by Yao *et al.* (2007), Ahmad *et al.* (2010) and Eskandari and Toygar (2015) could be replicated. However, proposed systems using EigenLBPVL, FisherLBPVL, SVMLBPVL and LBPHL feature-classifier combinations all achieved perfect accuracy in the replicated experiments, using only a single training sample. Comprehensive experiments to further assess the proposed fusion approach are thus conducted on the publicly available datasets discussed in Section 9.3.

9.3 Creation of Multimodal Biometric Datasets

Due to the limited number of publicly available multimodal biometric datasets containing the fingerprint, face, palmprint and iris modalities, pseudo multimodal datasets were created from the individual datasets introduced in the previous four chapters for use in bimodal fusion experimentation.

The FVC2004 fingerprint, Fei face, CASIA-Palmprint and CASIA-Iris-Lamp datasets were fused in all possible six unordered combinations using eight samples per individual/class. Since the FVC2004 dataset contains about 100 unique fingerprints, the datasets of the other three modalities were limited to 100 – comprising 50 individuals and 50 impostors. The order of classes and samples per class were unchanged from the description in Section 4.2 with the exception of the Fei face dataset, in which the number of males

and females was halved to keep the gender ratio 50:50. This pseudo multimodal dataset is referred to as Dataset A.

A pseudo multimodal dataset with a combination of the largest number of classes was created from the MCYT fingerprint, Fei face, CASIA-Palmprint and CASIA-Iris-Lamp datasets, with six bimodal combinations of the fingerprint, face, palmprint and iris for each class. This is the largest combination possible that allows for even distribution and is referred to as Dataset B. The first 200 classes (limited by Fei) are included with eight samples (limited by CASIA-Palmprint) each.

Datasets with more than 1000 classes are often limited to only one or two samples, such as FERET and the "labelled in the wild" type, as these are more specific to verification or detection. Creating larger datasets is possible by combining multiple datasets, but this typically results in an unrealistic class distribution due to differences in the acquisition methods, such as differing face angles and camera sensors. Stacking datasets of individual modalities is thus not performed. The open-set identification accuracy of bimodal combinations of the following pseudo multimodal datasets are similarly evaluated using the largest number of classes and samples common to all modalities:

- Dataset C – FVC2004 fingerprint, Fei face, IITD-Palmprint and CASIA-Iris-Lamp datasets – 100 classes with six samples each.
- Dataset D – FVC2004 fingerprint, Fei face, CASIA-Palmprint and IITD-Iris datasets – 100 classes with five samples each.
- Dataset E – FVC2004 fingerprint, PUT face, CASIA-Palmprint and CASIA-Iris-Lamp datasets – 100 classes with eight samples, with images from PUT face including only left and right angles at 60°, 45°, 30° and 0°.

Any duplicate combinations resulting from the creation of these datasets are only discussed if the number of classes differ. Furthermore, the results of the duplicates are not detailed, but the observed effect of different class sizes is commented on.

Comparison of the open-set identification accuracy on all datasets is shown by ROC curves for FA and FC, in a side-by-side manner, in subsequent sections for each bimodal combination.

9.4 Fingerprint and Face Feature-Fusion Results

In this and subsequent result sections, the optimal resolution for fusion is determined per bimodal combination. However, fusion results only for the optimal resolutions are illustrated as was the case with the four individual modalities discussed in Chapters 5 to 8. Furthermore, parameters or image resolutions that agree with or differ from those recorded in the individual modalities are noted.

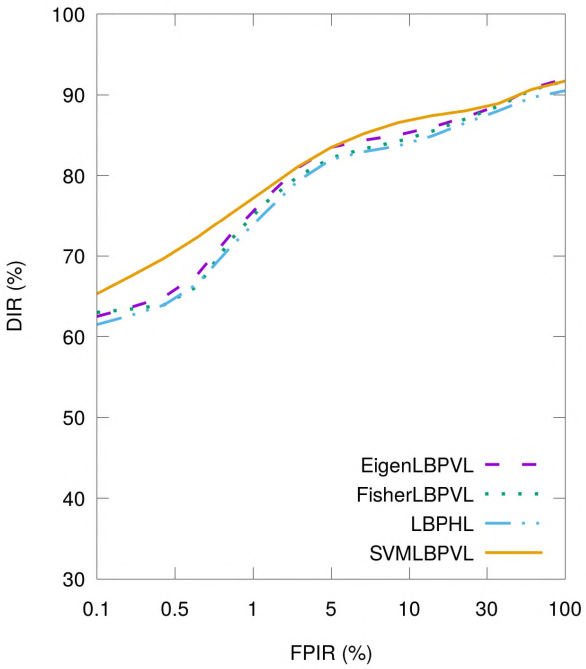
9.4.1 Dataset A: Fusing FVC2004 Fingerprint and Fei Face

For this dataset, Eigen, Fisher and SVM achieved their best results using image resolutions between 50×50 and 100×100. The optimal resolution for both FA and FC was 50×50 – the same resolution used in the single modality experiments.

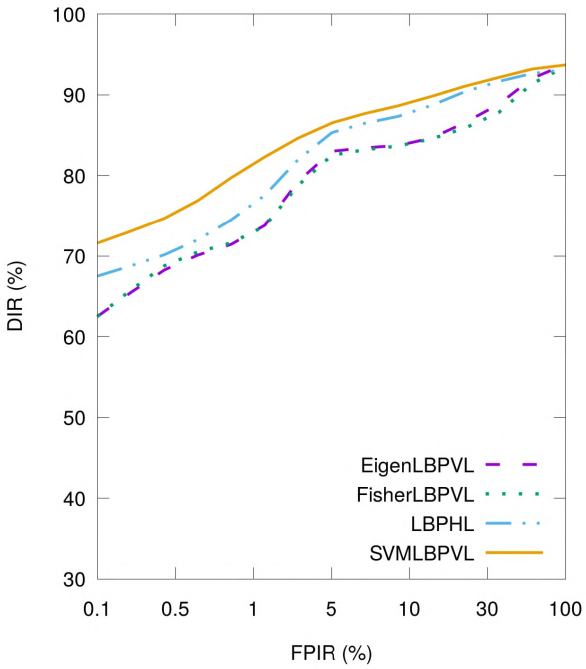
The optimal LBPH parameters when using FC were established as a 12-pixel radius and six neighbours. The corresponding optimal image resolution was 150×150 – often the closest to the segmented resolution of datasets used in the individual experiments. Unexpectedly, FA yielded its best results also with a 12-pixel radius and six neighbours. The results were, on average, markedly better when using the new parameters instead of the established optimal parameters obtained in the single modality fingerprint and face experiments by 16% and 18%, respectively. Of note, the single modalities were retested with a 12-pixel radius and six neighbours but achieved poor performance.

The graphs in Figures 9.2a and 9.2b depict the evaluation of FA and FC fusion methods, respectively when using one training sample on Dataset A. SVM achieves the best average performance for both fusion methods and by a distinct margin when FPIR $< 5\%$.

Eigen achieves the highest DIR when disregarding FPIR – as was the case in the Fei face experiment in Section 6.3.3 – narrowly outperforming SVM in this regard (refer to Figures 9.2c and 9.2d for actual values). On average, Eigen and Fisher achieve the highest FPIR for FC, while LBPH does so for FA. All four classifiers achieve their best accuracies using FC.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0.6	1.5	2.3	8.0
1-FisherLBPVL	0	0	0	0.6	1.5	2.6	8.3
1-LBPHL	0	0	0	1.0	1.0	2.0	9.5
1-SVMLBPVL	0	0	0.1	1.4	2.1	2.9	8.3

(c) FA FNIR values

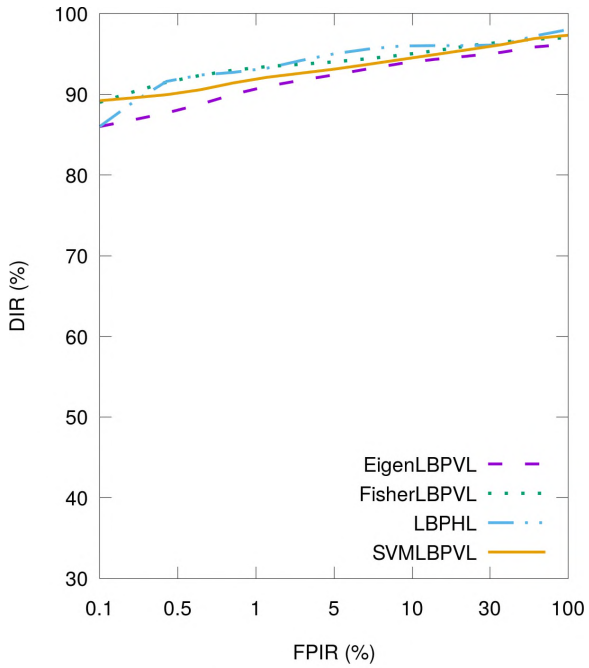
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0	0.4	6.2
1-FisherLBPVL	0	0	1	0.5	1.2	0.4	6.5
1-LBPHL	0.5	0.5	0.5	0.5	0.5	1.5	6.9
1-SVMLBPVL	0	0	0.2	1.3	1.9	2.7	6.3

(d) FC FNIR values

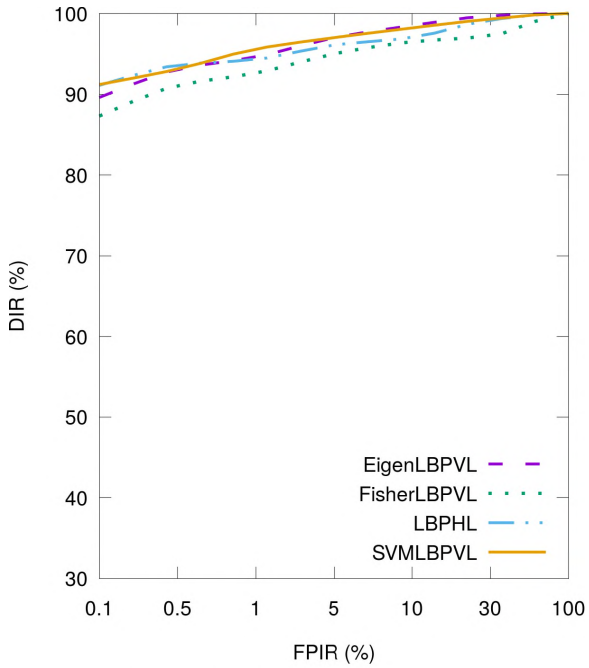
Figure 9.2: One fused FVC2004 fingerprint and Fei face training sample, on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

The results for three training samples illustrated in Figure 9.3a and 9.3b show a similar trend and accuracy of the classifiers. FC achieves perfect DIR values for all classifiers and, on average, approximately 3% better accuracy than FA. Since part of the feature concatenated vector comes from the FVC2004 fingerprint dataset, Fisher achieves the worst scaling with three training samples as expected (see Figure 5.14b). On the other hand, Fisher achieves better DIR values when FPIR $< 5\%$ with the FA method, since the total

variance increases, thereby positively affecting inter-class data separation. Furthermore, when using FA, Fisher outperforms Eigen and SVM in both DIR and FNIR, while LBPH achieves the best accuracy. However, this is reversed in the FC results shown in Figure 9.3b, where Eigen and SVM yield the best overall accuracies. Both fusion methods, as well as all classifier methods yield similar results with no outstanding performer.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.1	3.8
3-FisherLBPVL	0	0	0	0	0	0	3.0
3-LBPHL	0	0	0	0	0	0	2.0
3-SVMLBPVL	0	0	0	0	0	0	2.7

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	0
3-FisherLBPVL	0	0	0	0	0	0	0
3-LBPHL	0	0	0	0	0	0	0
3-SVMLBPVL	0	0	0	0	0	0	0

(d) FC FNIR values

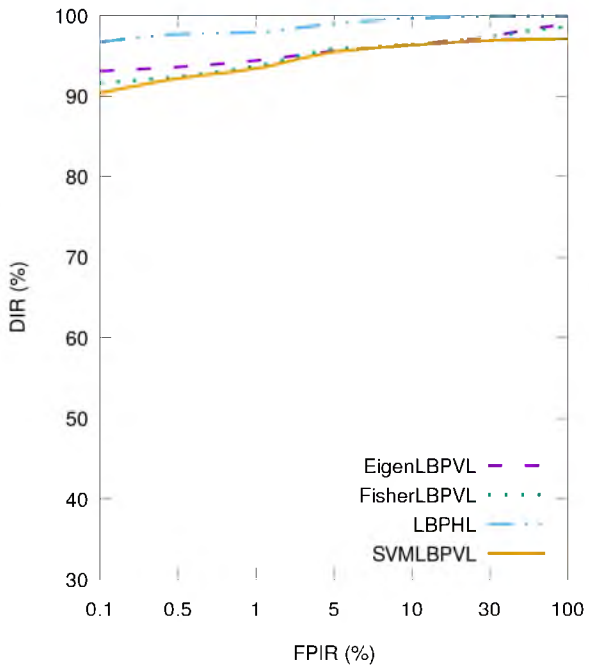
Figure 9.3: Three fused FVC2004 fingerprint and Fei face training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

9.4.2 Dataset B: Fusing MCYT Fingerprint and Fei Face

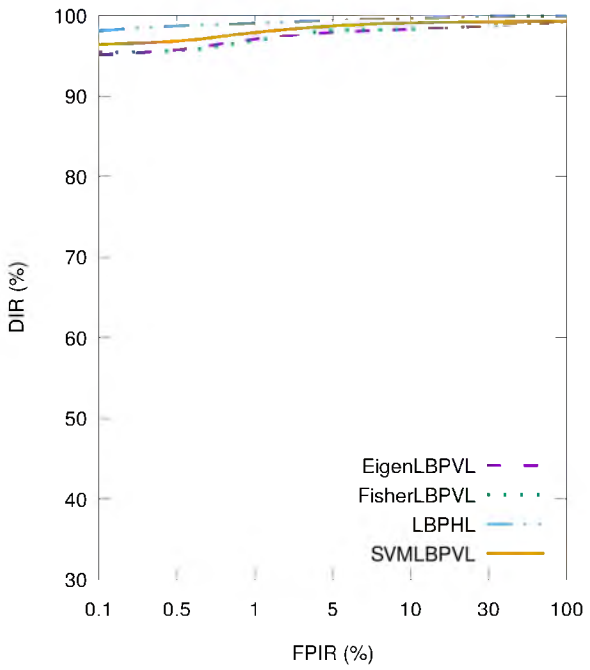
Eigen, Fisher and SVM achieved their best (and similar) results at a 50×50 resolution for both FA and FC. At 150×150, the DIR of FA and FC drops on average by 3% and 8%,

respectively. The optimal LBPH parameters remain a 12-pixel radius with six neighbours for both FA and FC.

LBPH achieves similar accuracies for FA and FC, where FC only slightly outperforms FA when FPIR $< 1\%$. Whereas SVM outperforms LBPH for Dataset A when using one training sample, LBPH is the best performer in all tests on this dataset. Using FC, all four classifiers achieve their best accuracies, which are similar in magnitude and trend pattern for all the DIR and FNIR values shown in the graphs and tables in Figure 9.4.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0.3	0.4	0.7	1.1
1-FisherLBPVL	0	0	0	0.4	0.4	0.7	1.4
1-LBPHL	0	0	0	0	0	0	0.1
1-SVMLBPVL	0	0	0	0.5	1.1	1.4	2.9

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0.1	0.1	0.3	0.4	0.9
1-FisherLBPVL	0	0	0.1	0.3	0.3	0.6	0.7
1-LBPHL	0	0	0	0	0	0	0.1
1-SVMLBPVL	0	0	0	0	0.1	0.5	0.7

(d) FC FNIR values

Figure 9.4: One fused FVC2004 fingerprint and Fei face training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

For FC, Eigen and Fisher achieve a perfect DIR when FPIR $< 5\%$, while SVM and LBPH achieve a perfect DIR when FPIR $= 0\%$. LBPH also achieves a perfect DIR when FPIR $= 0\%$ for FA, while Eigen does so when FPIR $= 0\%$. Fisher and SVM perform poorly

when using FA and both achieve a maximum DIR of 99.8% when FPIR = 100%. Table 9.1 shows the DIR for each classifier at FPIR = 0%, with SVM and LBPH performing the best.

Table 9.1: Results on fingerprint-face fusion at 0% FPIR and 0% FNIR with three training samples on Dataset B.

Classifier	FA	FC
3-EigenLBPVL	97.0	98.4
3-FisherLBPVL	96.0	95.8
3-LBPHL	100	100
3-SVMLBPVL	96.0	100

9.4.3 Dataset E: Fusing FVC2004 Fingerprint and PUT Face

Most results for Dataset E are omitted, as all bimodal combinations involving the PUT dataset, except when combined with FVC2004, achieved a perfect accuracy at FPIR = 0% when trained using the left pose 30°.

It is worth noting that similar to the results on PUT reported in Chapter 6, both FA and FC methods achieve perfect accuracy for all classifiers except LBPH. However, LBPH yields a near-perfect accuracy of 99.7% and 97.1% for FC and FA, respectively. The best results were achieved using the same optimal resolution as reported for PUT in Section 6.3.4, that is, 150×150.

The results when using Dataset E are shown at FPIR = 0% in Table 9.2. LBPH performs relatively poorly compared with all other combinations of methods. For Eigen and Fisher, FC performs poorly when using the LBPVL, while FA performs only slightly better. On the other hand, FC performs similarly or better than FA when using HEL. Finally, SVM clearly favours FC for both LBPVL and HEL. Single modality experiments showed that HEL and LBPVL perform similarly for both the FVC2004 and PUT datasets. However, this was tested for FVC2004 at 50×50. Retesting at 150×150 reveals that HEL outperforms LBPVL on the FVC2004, thereby corroborating the fusion result. Of note, LBPL performed relatively poorly when using either FA or FC. Moreover, this is the first

bimodal combination that results in diminished performance over the better performing component modality, namely, the face, and is thus a special case. The worst performer, LBPH, also had the largest difference in performance between the single modality experiments on PUT and the fused results. All classifiers achieve perfect accuracy when using three training samples for both FA and FC methods including the use of either LBPVL or HEL but not LBPL.

Table 9.2: Comparison of FA and FC, using LBPVL or HEL on Dataset E.

Classifier	**FA LBPVL**	**FC LBPVL**	**FA HEL**	**FC HEL**
1-Eigen	96.6	95.1	99.7	99.7
1-Fisher	97.1	95.4	99.1	99.4
1-LBPHL (not using LBPVL)	91.7	94.8	–	–
1-SVM	85.1	98.0	93.1	99.7

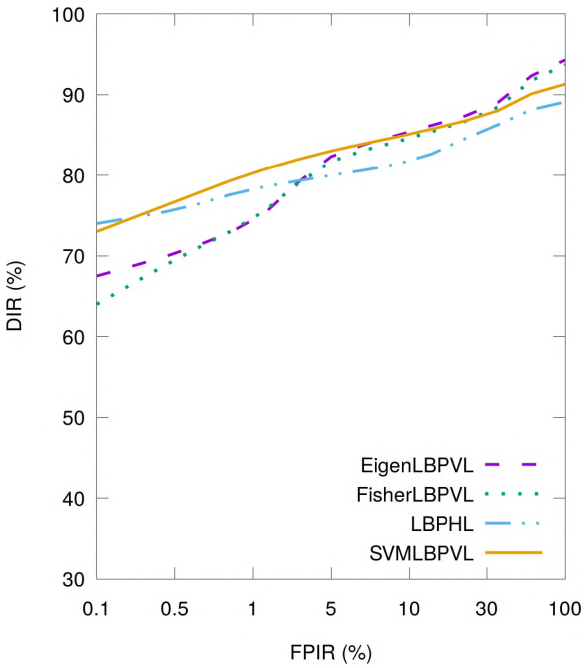
The fused datasets tend to yield better results for the same classifier that yielded the best results in the tests on the individual component modalities. In the case of Dataset A, this is SVM and LBPH on Fei face and FVC2004 fingerprint, for Dataset B, LBPH is optimal on MCYT fingerprint, and for Dataset E, all classifiers except LBPH are optimal on PUT face.

9.5 Fingerprint and Palmprint Feature-Fusion Results

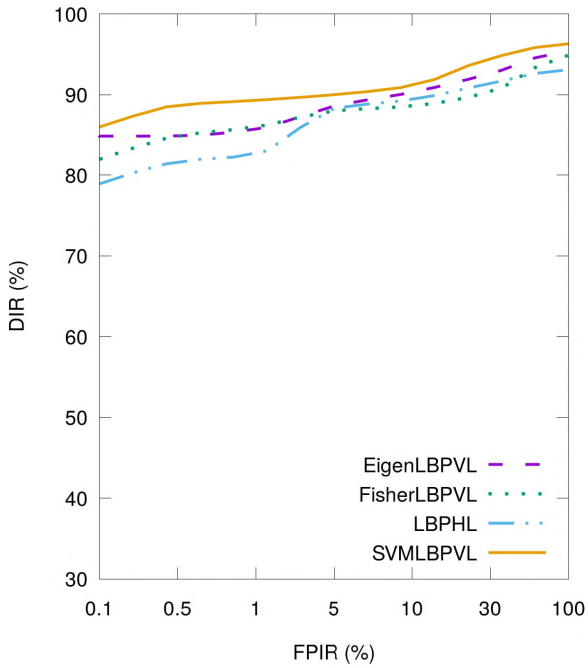
9.5.1 Dataset A: Fusing FVC2004 Fingerprint and CASIA-Palmprint

For this dataset, the optimal image resolution, per single modality, for Eigen, Fisher and SVM was found to be 50×50. This is an expected outcome, as FVC2004 and CASIA-Palmprint performed optimally at that resolution in the single modality experiments. A 65×65 resolution produced similar accuracies to 50×50, although LBPH preferred a 150×150 image resolution per feature vector.

Comparative results for the FA and FC methods on the fused FVC2004 fingerprint and CASIA-Palmprint dataset are shown in Figure 9.5a and 9.5b, respectively. SVM once again achieves the best average (FPIR) performance for both fusion methods when using one training sample. Eigen and SVM achieve the highest DIR values for FA and FC, respectively. On average, Eigen and Fisher achieve the worst DIR when FPIR $< 5\%$ for FC, while LBPH does so for FA. All four classifiers achieve their best accuracies when using FC.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.3	0.9	5.7
1-FisherLBPVL	0	0	0	0	0.3	0.9	6.3
1-LBPHL	0	0	0	0.6	0.6	1.4	10.9
1-SVMLBPVL	0	0	0	0.3	0.3	1.4	8.7

(c) FA FNIR values

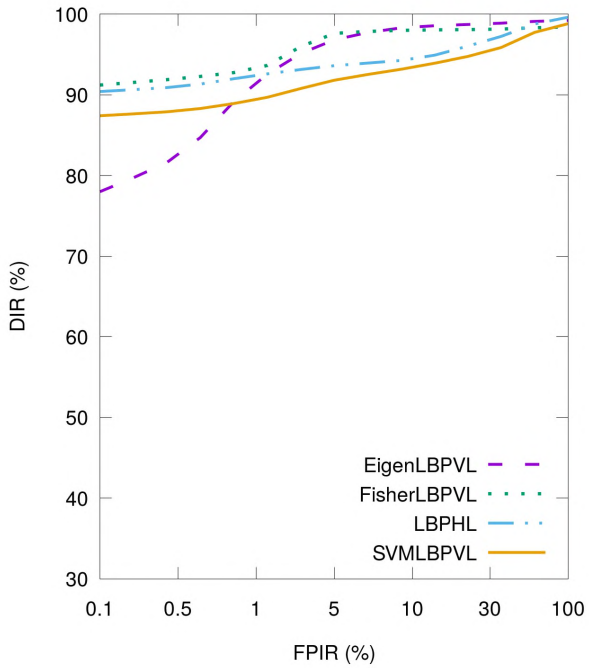
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0.3	0.3	0.6	4.6
1-FisherLBPVL	0	0	0	0	0.6	0.6	5.1
1-LBPHL	0	0	0.5	0	0	1.1	6.9
1-SVMLBPVL	0	0	0	0	0	0.7	3.7

(d) FC FNIR values

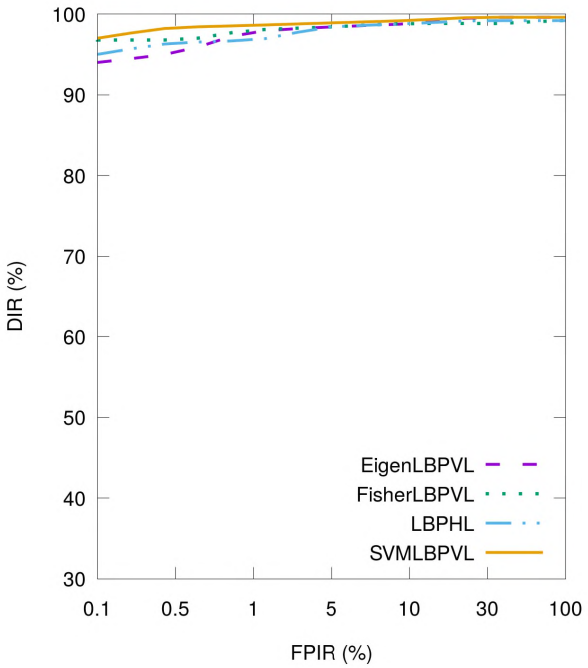
Figure 9.5: One fused FVC2004 fingerprint and CASIA-Palmprint training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

When using three training samples the accuracy advantage of FC is clearly observed for all classifiers, resulting in similar accuracies and trend patterns. Furthermore, SVM is the worst performer when using FA but yields the best average accuracy when using FC, albeit by a narrow margin. Fisher performance drops noticeably when FPIR $< 1\%$. The

large reduction in Eigen's performance when FPIR $< 1\%$ was further investigated and improved to be similar to Fisher by simply reducing the number of principal components to 100. Using 200 principal components determined in Section 5.3.2 proved to be optimal up till this point. The reason for the performance loss is a retained principal component, which Fisher removes by performing extra class-specific dimensionality reduction. Using different resolutions did not improve the result, and hence the optimal number of principal components remains to be verified in future work.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0.4	0.4	0.8
3-FisherLBPVL	0	0	0	0	0	0.4	1.6
3-LBPHL	0	0	0	0	0	0	0.6
3-SVMLBPVL	0	0	0	0	0	0.8	1.2

(c) FA FNIR values

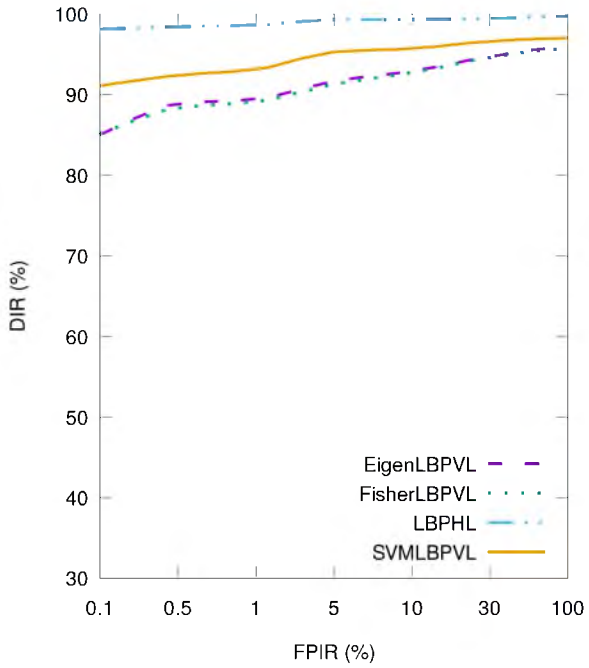
Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	0.4
3-FisherLBPVL	0	0	0	0	0	0	0.6
3-LBPHL	0	0	0	0	0	0	0.6
3-SVMLBPVL	0	0	0	0	0	0	0.4

(d) FC FNIR values

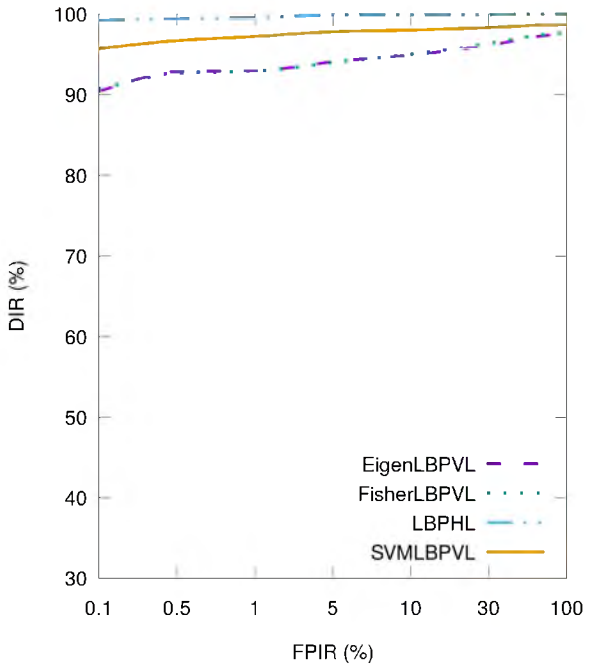
Figure 9.6: Three fused FVC2004 fingerprint and CASIA-Palmprint training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

9.5.2 Dataset B: Fusing MCYT Fingerprint and CASIA-Palmprint

Accuracy results for Dataset B are noticeably lower when using CASIA-Palmprint instead of Fei face as seen in Figure 9.7. LBPH again performs the best on the fingerprint and palmprint combination, with 99.2% DIR at FPIR = 0%. This is followed by SVM, with Eigen and Fisher yielding less desirable results on the fingerprint and palmprint combination similar to the single modality experiments.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0.1	0.1	0.6	0.7	1.9	4.1
1-FisherLBPVL	0.1	0.1	0.3	0.7	0.7	2.1	4.3
1-LBPHL	0	0	0	0	0	0	0.3
1-SVMLBPVL	0	0	0.2	0.5	0.9	1.3	3.0

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.1	0.7	2.4
1-FisherLBPVL	0	0	0	0	0.1	0.6	2.3
1-LBPHL	0	0	0	0	0	0	0.1
1-SVMLBPVL	0	0	0	0	0.1	0.7	1.3

(d) FC FNIR values

Figure 9.7: One fused MCYT fingerprint and CASIA-Palmprint training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Results for three training samples are generally high as shown in Table 9.3. Each classifier performs relatively similarly to the test results in Table 9.1 with LBPH and SVM again achieving perfect accuracy.

Table 9.3: Results on fingerprint-palmprint fusion FPIR and 0% FNIR with three training samples using Dataset B.

Classifier	FA	FC
3-EigenLBPVL	95.4	97.2
3-FisherLBPVL	96.4	98.6
3-LBPHL	100	100
3-SVMLBPVL	98.1	100

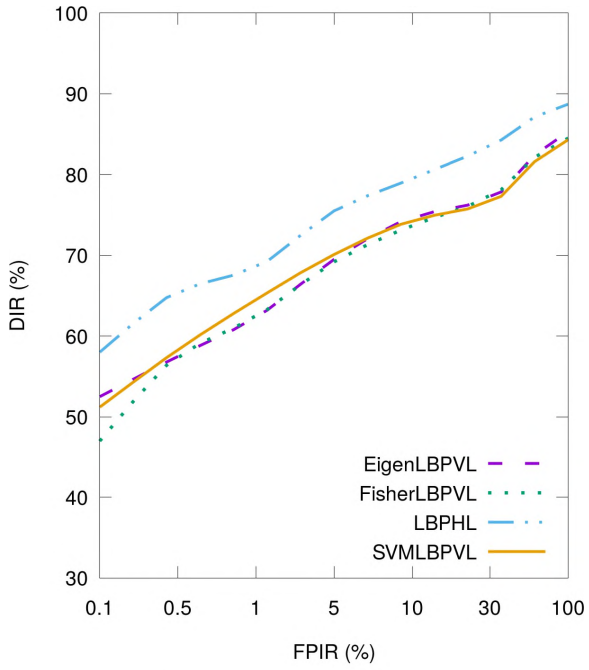
9.5.3 Dataset C: Fusing FVC2004 Fingerprint and IITD-Palmprint

Marked improvements are noted for all classifiers on this fused combination compared with the component modality results. SVM outperforms the rest on average for FC, while LBPH is the best performer by a significant margin for FA, while also achieving the highest DIR at FPIR = 100% as shown in Figure 9.8c.

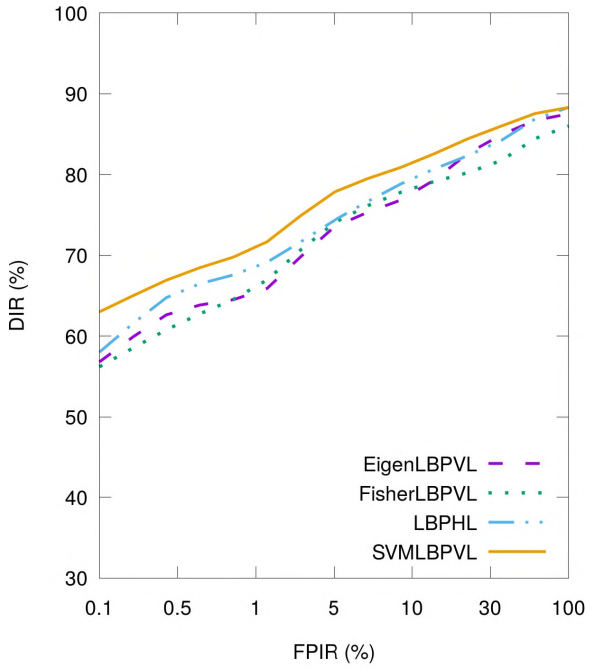
Considerable improvements in accuracy are particularly noted when using three training samples. LBPH and SVM achieve near-perfect accuracies for FC. LBPH on the other hand, again performs poorly when using the FA method as it did when using one training sample; even changing parameters did not improve this result. Eigen achieves the best performance for FA, confirming the consistent trend in results for all datasets thus far. DIR values for FA at FPIR = 100% are 98.5%, 98.0%, 97.7% and 97.9% for Eigen, Fisher, LBPH and SVM, respectively. DIR values for FC at FPIR = 100% are all 100%.

Table 9.4: Results on fingerprint-palmprint fusion at 0% FPIR and 0% FNIR with three training samples using Dataset C.

Classifier	FA	FC
3-EigenLBPVL	90.3	91
3-FisherLBPVL	90.1	90.5
3-LBPHL	85	98.9
3-SVMLBPVL	90.1	99



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.5	5.5	14.7
1-FisherLBPVL	0	0	0	0	0.5	5.5	14.7
1-LBPHL	0	0	0.5	1.0	1.0	4.0	11.3
1-SVMLBPVL	0	0	0.6	0.9	1.1	6.9	15.7

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0.4	3.4	12.5
1-FisherLBPVL	0	0	1	0	0.4	3.4	14
1-LBPHL	0	0	0.5	1.1	1.1	4.2	11.8
1-SVMLBPVL	0	0	0.4	0.7	1.0	4.7	11.8

(d) FC FNIR values

Figure 9.8: One fused FVC2004 fingerprint and IITD-Palmprint training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

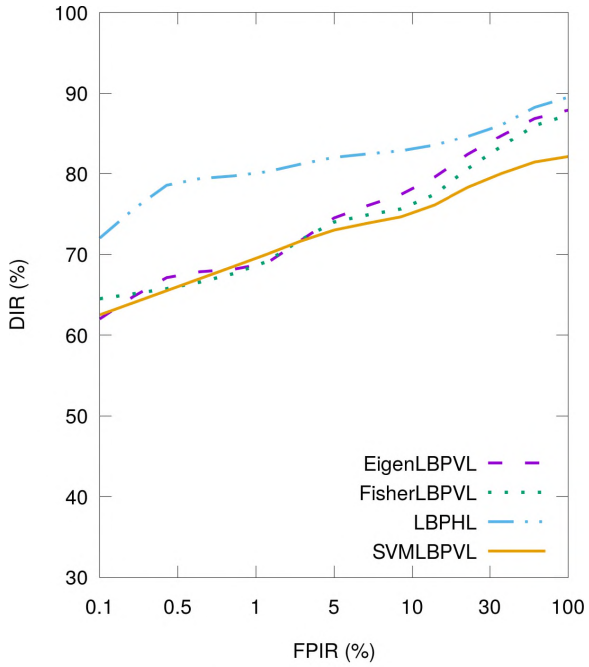
9.6 Fingerprint and Iris Feature-Fusion Results

9.6.1 Dataset A: Fusing FVC2004 Fingerprint and CASIA-Iris-Lamp

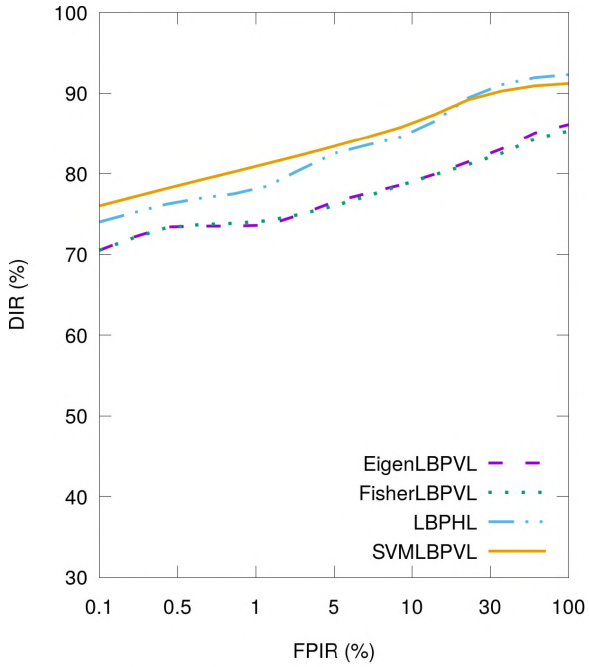
For this dataset, the optimal image resolution of both fusion components (fingerprint and iris) for Eigen, Fisher and SVM was found to be 50×50 to 65×65, as fingerprint strictly preferred low resolutions in the single modality experiments. Using 65×65 instead of 100×100 yielded an average 11% higher DIR for Eigen, Fisher and SVM. It should be noted that using a 65×65 resolution performed 3% better than 50×50 as the reduction in iris resolution affected its finer texture pattern. LBPH preferred a 150×150 image

resolution per feature vector.

Based on the results in Figure 9.9, Eigen and Fisher achieve poor DIR for FPIR $<$ 30%, which corroborates the trend identified in tests on both individual datasets. SVM continues the trend of achieving low accuracy when using FA. SVM and LBPH perform similarly for FC, with SVM achieving the best overall accuracies across all FPIRs, while LBPH surpasses SVM only for FPIR $>$ 30% by a 1% higher DIR. This is particularly interesting as LBPH outperformed SVM on the individual datasets when using a single training sample. A linear SVM deals well with higher dimensionality relative to the other classifiers and hence benefits particularly well from FC on these fused datasets.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0.5	0.5	2.1	5.5	12.1
1-FisherLBPVL	0	0	0	0	1.5	5.0	12.7
1-LBPHL	0	0	0	1.2	1.2	1.2	10.5
1-SVMLBPVL	0	0	0.2	1.5	3.2	7.7	17.9

(c) FA FNIR values

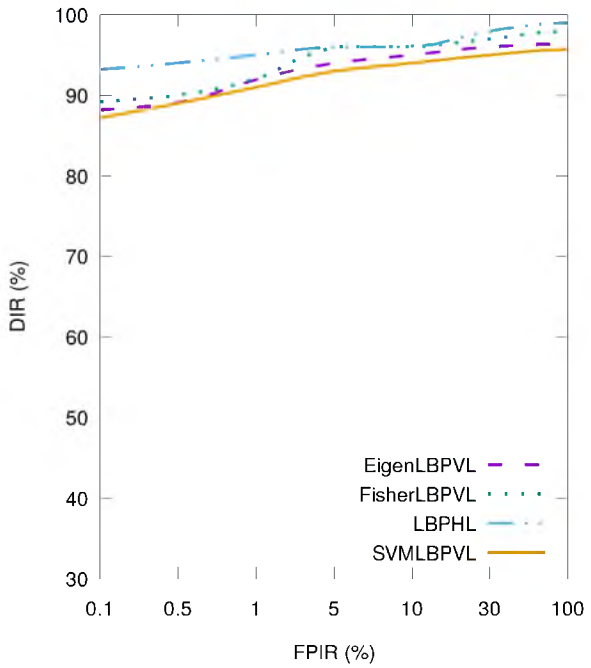
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	1.0	1.4	4.0	13.9
1-FisherLBPVL	0	0	0	1.0	1.0	5.2	14.7
1-LBPHL	0	0	0	0	0	2.5	7.7
1-SVMLBPVL	0	0	0	0.3	1.5	3.3	8.8

(d) FC FNIR values

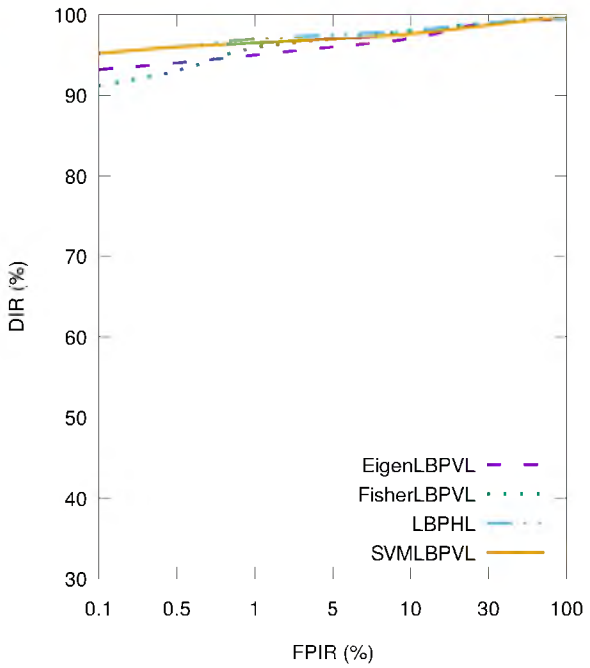
Figure 9.9: One fused FVC2004 fingerprint and CASIA-Iris-Lamp training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Results for three training samples (see Figure 9.10) show that FC again achieves higher

DIR values for all classifiers, but with a smaller margin when FPIR $< 30\%$. SVM again performs relatively poorly when using FA as opposed to FC. It should be noted that all classifiers achieve similar near-perfect DIR values at 100% FPIR (which is essentially closed-set identification). Finally, LBPH achieves the best accuracies for FA by a significant margin – almost on par with its FC error rates when using either one or three training samples.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	2.1	3.6
3-FisherLBPVL	0	0	1.1	1.1	1.4	1.4	2.0
3-LBPHL	0	0	0	0	0	0	1.0
3-SVMLBPVL	0	0	0	0	1.3	2.9	4.3

(c) FA FNIR values

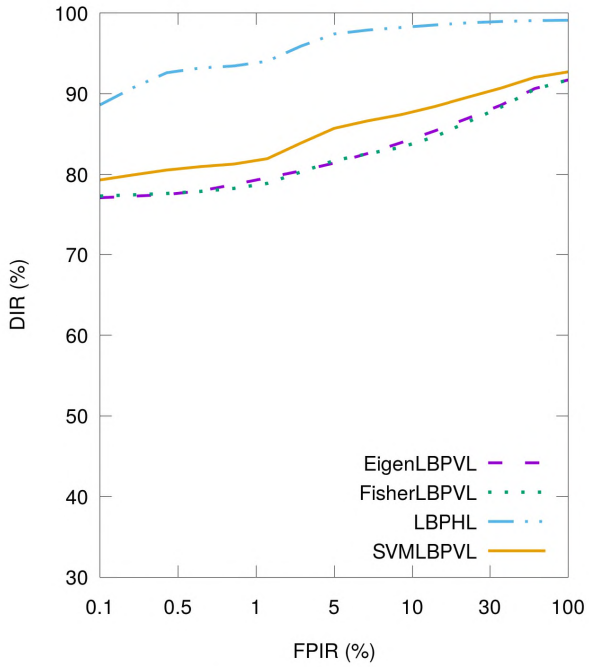
Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0.3	0.6
3-FisherLBPVL	0	0	0	0	0	0.1	0.4
3-LBPHL	0	0	0	0	0	0	0.6
3-SVMLBPVL	0	0	0	0	0	0	0.4

(d) FC FNIR values

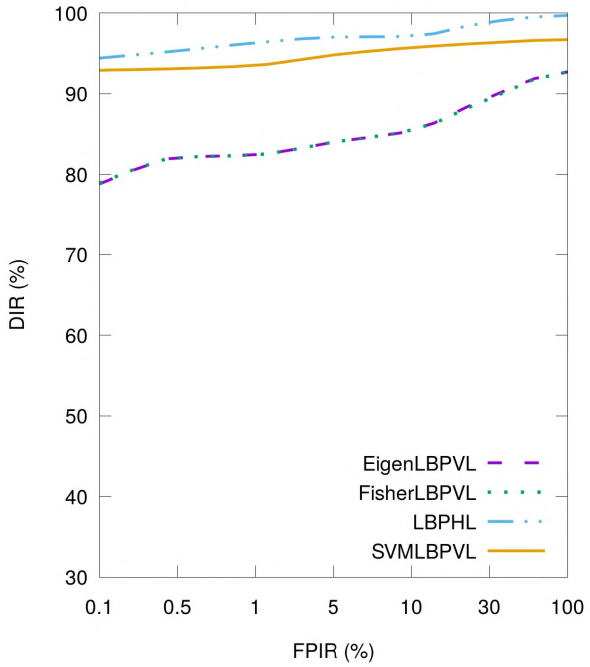
Figure 9.10: Three fused FVC2004 fingerprint and CASIA-Iris-Lamp training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

9.6.2 Dataset B: Fusing MCYT Fingerprint and CASIA-Iris-Lamp

Resolutions remain the same as for Dataset A as the fingerprint preferred low resolutions. Dataset B results are lower when CASIA-Iris-Lamp is the second component instead of Fei face or CASIA-Palmprint as seen in Figure 9.11. LBPH appears to perform the best for all MCYT combinations. SVM results are a narrow second, while Eigen and Fisher yield the weakest results as is the case for all fingerprint combinations. Furthermore, Eigen and Fisher generally achieve near-identical results on this dataset.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0.1	0.1	0.1	0.4	1.1	2.7	8.3
1-FisherLBPVL	0.1	0.1	0.1	0.4	0.7	2.6	8.3
1-LBPHL	0	0	0	0	0	0.1	0.9
1-SVMLBPVL	0	0	0	0.5	1.9	2.5	7.3

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0.1	0.1	0.3	0.3	0.6	1.9	7.3
1-FisherLBPVL	0.1	0.1	0.3	0.3	0.6	1.7	7.3
1-LBPHL	0	0	0	0	0	0	0.3
1-SVMLBPVL	0	0	0	0.4	0.9	1.5	3.3

(d) FC FNIR values

Figure 9.11: One fused MCYT fingerprint and CASIA-Iris-Lamp training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Results for three training samples (see Table 9.5) show that FA and FC perform similarly for Eigen and Fisher, whereas FC outperforms FA for SVM and LBPH. Moreover,

the trend whereby LBPH and SVM achieve perfect accuracy recurs when MCYT is a component in the fusion.

Table 9.5: Results on fingerprint-iris fusion at 0% FPIR and 0% FNIR using three training samples on Dataset B.

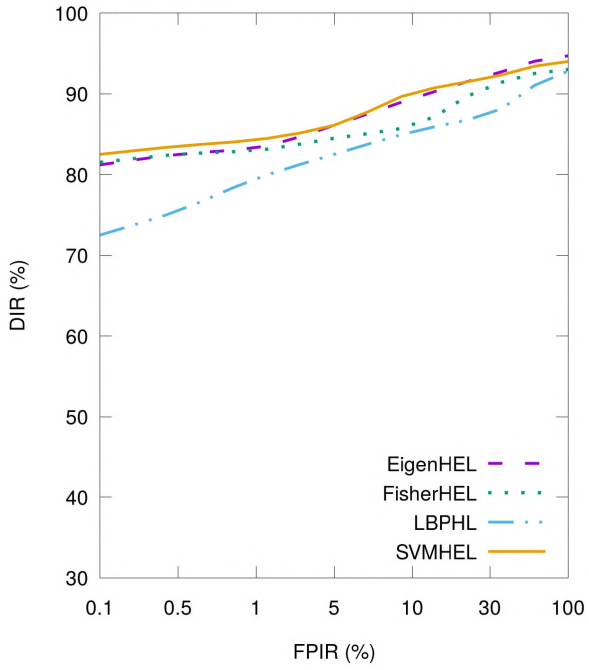
Classifier	FA	FC
3-EigenLBPVL	91.0	91.0
3-FisherLBPVL	93.0	93.2
3-LBPHL	97.8	100
3-SVMLBPVL	97.5	100

9.6.3 Dataset D: Fusing FVC2004 Fingerprint and IITD-Iris

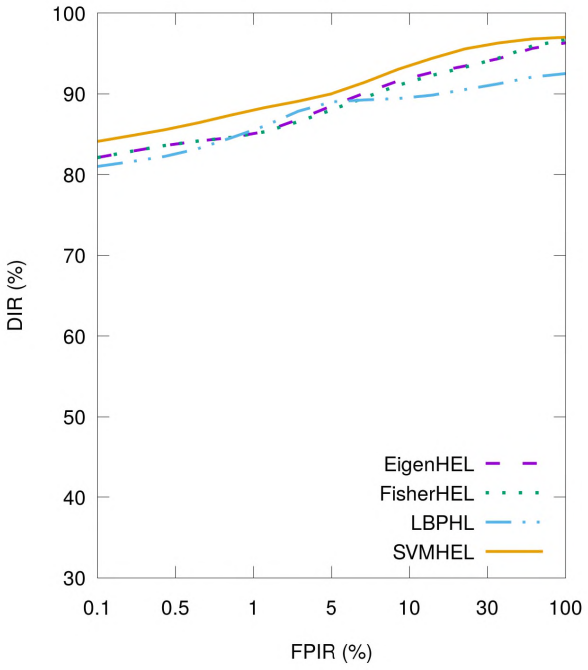
The optimal image resolution of 75 × 75 was found for Eigen and Fisher, while SVM preferred 100×100, which is the same as used in the single modality experiments performed on IITD-Iris. However, results obtained for resolutions between 75 × 75 and 100 × 100 varied by a maximum of only 3%. LBPH once again preferred a 150×150 image resolution per feature vector. Of note, the HEL results are shown as this method results in an 8% better DIR on average than LBPVL (and 10% better than LBPL). In other words, LBPH marginally outperforms the other classifiers when they are combined with LBPVL or LBPL as opposed to the results that follow.

Eigen, Fisher and SVM achieve similar results for FA and FC, but outperform LBPH. Referring to Figures 9.12a and 9.12b, Eigen and SVM achieve the highest DIR for FA and FC, respectively. Although LBPH yields the lowest DIR values for both FA and FC, it has better FNIR values than the other classifiers. All classifiers achieve approximately 3% lower accuracy than the IITD-Iris component. This is due to the FVC2004 and IITD-Iris components having vastly different accuracies (and quality).

Using three training samples (not shown in figures), all classifiers achieve perfect accuracy at FPIR = 0% except LBPH, which achieves 99%. This is an improvement over the accuracy of the dominant IITD-Iris component. This is expected as high accuracy results are achieved using one training sample as was the case with the single modality results on



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenHEL	0	0.5	0.5	1.3	1.5	1.7	5.3
1-FisherHEL	0	1.0	1.0	1.3	2.5	4.3	7.0
1-LBPHL	0	0	0.5	0.3	0.5	1.3	7.2
1-SVMHEL	0	0	0	1.3	1.7	2.7	6.0

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenHEL	0	0	0	0	1.5	2.0	3.7
1-FisherHEL	0	0	0	0	1.5	2.0	3.3
1-LBPHL	0	0	0	0	0	0	7.5
1-SVMHEL	0	0	0	0	0	1.2	3.0

(d) FC FNIR values

Figure 9.12: One fused FVC2004 fingerprint and CASIA-Iris-Lamp training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

the IITD-Iris dataset (although the latter results are expectedly lower than these fused results).

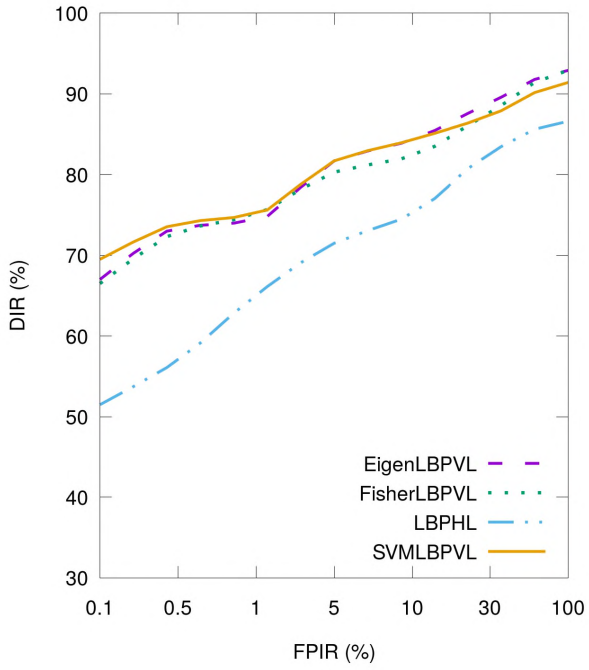
9.7 Face and Palmprint Feature-Fusion Recognition Results

9.7.1 Dataset A: Fusing Fei Face and CASIA-Palmprint

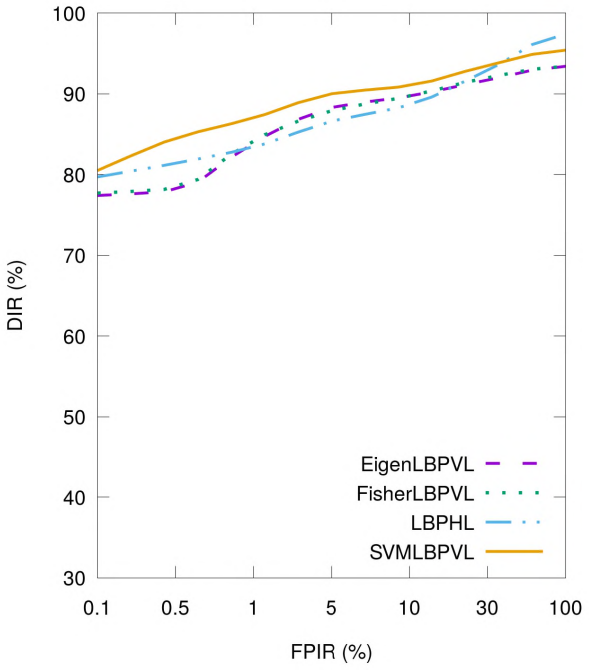
The optimal image resolution, per dataset component, for Eigen, Fisher and SVM was found to be 50×50 for FC and FA. This is attributed to the fact that performance of the experiments on the face modality in Section 6.3.3 did not significantly decrease when using 50×50 instead of the default segmented resolution. These values are identical to those recorded for the fingerprint and face single modality results. LBPH preferred a 150×150 image resolution per feature vector, but FA appears to underperform due to the use of an uncompromising resolution.

On the other hand, the 12-pixel radius with seven neighbours used for LBPH in previous experiments was not optimal for this fused dataset. Therefore, the following new parameters and their improvements in accuracy are shown for one training sample. Large accuracy improvements for FA of 20% and 8% were obtained at FPIR = 0.1% and FPIR = 100%, respectively, when using a 1-pixel radius with seven neighbours. For FC, accuracy improvements of 20% and 5% were obtained at FPIR = 0.1% and FPIR = 100%, respectively, when using a 6-pixel radius with seven neighbours. Using three training samples also showed improved results by 3% and 1% at FPIR = 0.1% and FPIR = 100%, respectively.

Referring to Figures 9.13a and 9.13b, SVM achieves the best average performance for both fusion methods as with the fingerprint and face individual modality experiments. However, LBPH achieves the highest DIR with FPIR > 30% for FC, confirming the trend identified for its palmprint component in Section 7.4. Furthermore, LBPH is outperformed by the other classifiers for FA by the largest margin thus far. This trend has been observed for all face and palmprint combinations. While Eigen and Fisher often show similar performance with one training sample, their performances are near-identical for both FPIR and FNIR on this fused dataset.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0.3	0.6	2.6	7.1
1-FisherLBPVL	0	0	0	0.3	0.6	2.6	7.1
1-LBPHL	0	0	0.6	1.4	1.9	4.5	13.4
1-SVMLBPVL	0	0	0	0.3	1.2	3.3	8.6

(c) FA FNIR values

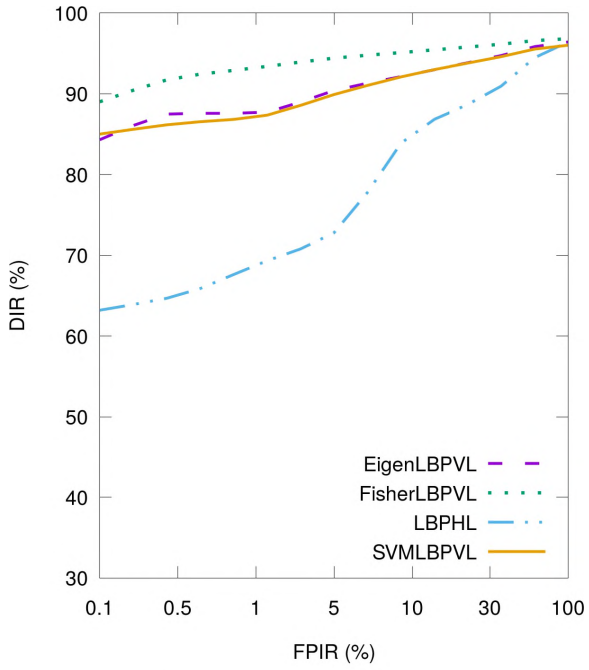
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0	1.1	6.6
1-FisherLBPVL	0	0	0	0	0	0.9	6.6
1-LBPHL	0	0	0	0.6	0.6	1.4	6.6
1-SVMLBPVL	0	0	0	0	0	0.6	4.6

(d) FC FNIR values

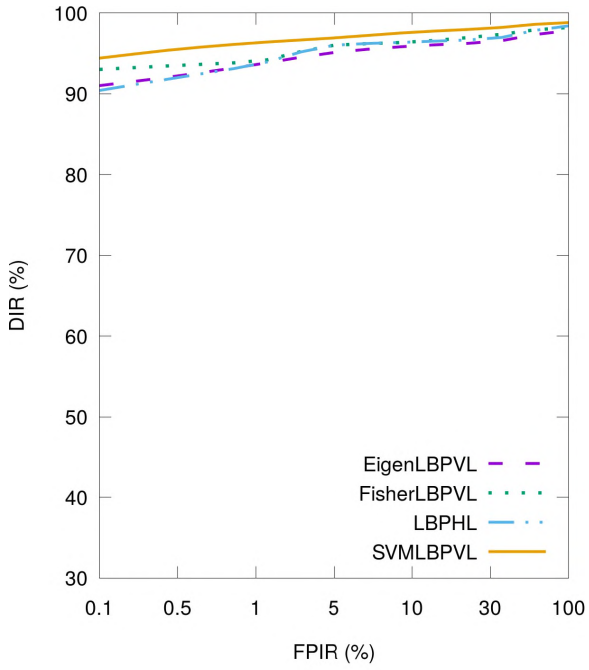
Figure 9.13: One fused Fei face and CASIA-Palmprint training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Referring to Figures 9.14a and 9.14b, Fisher is shown to perform similarly for both FA and FC. When using FA, LBPH achieves significantly different results at low and high FPIR values. SVM achieves the best performance on this dataset, followed by Fisher when considering both FPIR and FNIR values. However, as seen in Figure 9.14a, classifier performances vary by only a tiny margin for FC. This margin is near zero when considering closed-set identification (i.e. with FPIR = 100%).

Since this dataset was tested on 100 (Dataset A) and 200 (Dataset B) classes, it was noted that doubling the number of classes results in fairly similar trends and accuracies – on average, only 1% lower – for all classifiers. This suggests good scaling from the extracted features and classifier parameters. The same approximate changes in accuracies for all



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.2	3.6
3-FisherLBPVL	0	0	0	0	0	0.8	3.2
3-LBPHL	0	0	0	0	1.6	2.4	3.6
3-SVMLBPVL	0	0	0	0	0.4	2.3	4.0

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.2	2.2
3-FisherLBPVL	0	0	0	0	0	0.4	1.8
3-LBPHL	0	0	0	0	0.4	0.8	1.6
3-SVMLBPVL	0	0	0	0	0	0.4	1.2

(d) FC FNIR values

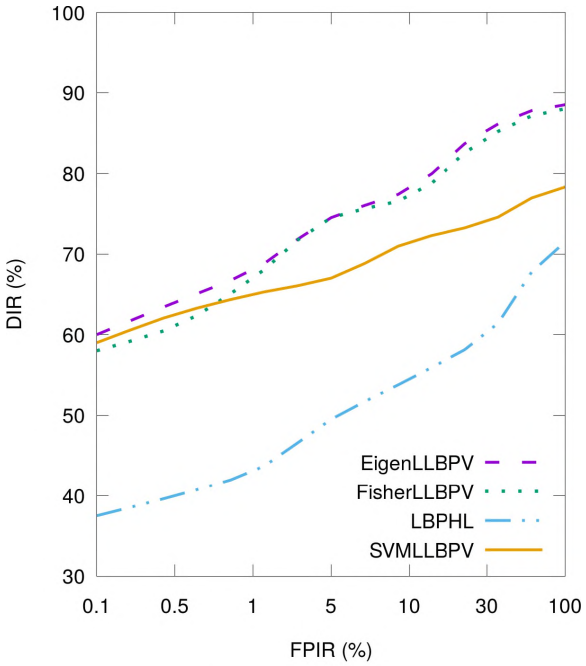
Figure 9.14: Three fused Fei face and CASIA-Palmprint training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

classifiers is an interesting result.

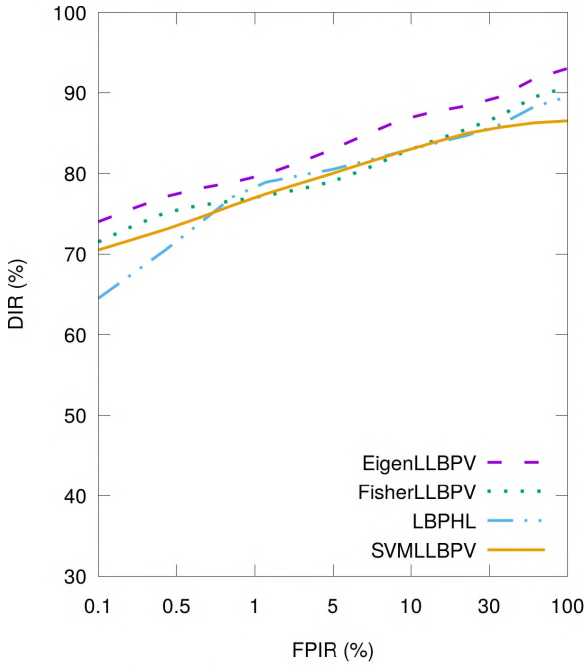
9.7.2 Dataset C: Fusing Fei Face and IITD-Palmprint

Minor improvements are noted for all classifiers on this fused combination compared with the results on the component datasets when using one training sample. Eigen and Fisher greatly outperform the rest for FA. Furthermore, Eigen achieves the best performance for FC, both on average and by the highest DIR at FPIR = 100%. However, these results appear to be a special case as they are achieved by applying the LoG filter before LBPV (LLBPV), resulting in better results than LBPVL by 6% for Eigen, Fisher and SVM.

Furthermore, using a resolution of 100×100 yielded the best results for LLBPV, while 50×50 did the same for LBPVL. The LLBPV results are thus shown in the figures and tables for this dataset (see Figure 9.15). Both the HEL and LBPL methods were also tested but yielded slightly lower accuracies than LBPVL. LBPH performance for FA is poor compared with other results regardless of parameter tuning.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLLBPV	0	0.5	0.5	0.7	1.7	6.0	11.5
1-FisherLLBPV	0.5	0.5	0.7	1.0	2.0	8.5	12.0
1-LBPHL	0	0	0	0	0	7.0	28.0
1-SVMLLBPV	0	0	0.2	1.5	3.2	7.7	21.7

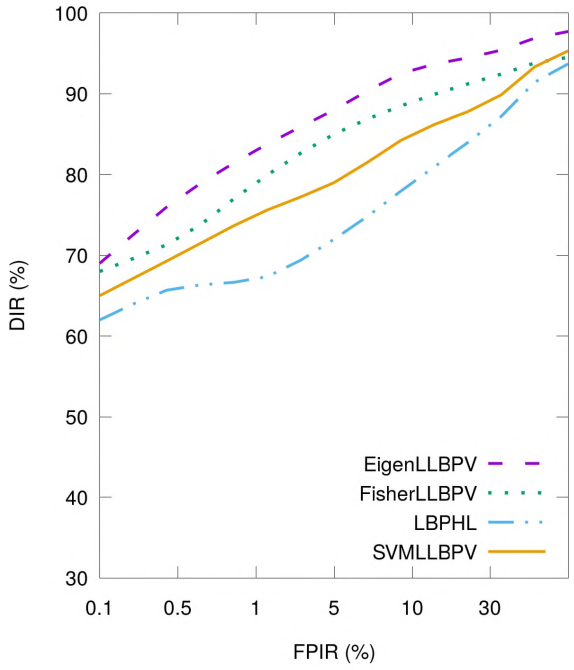
(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLLBPV	0	0	0	1.0	1.0	3.3	9.0
1-FisherLLBPV	0	0	0	0.7	1.7	3.3	10.3
1-LBPHL	0	1.0	1.0	2.0	4.5	6.6	10.5
1-SVMLLBPV	0	0	0	1.0	2.7	4.6	13.5

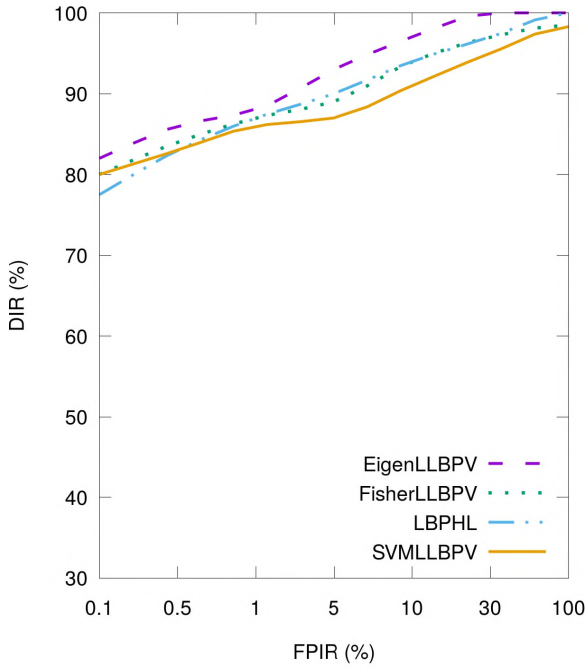
(d) FC FNIR values

Figure 9.15: One fused Fei face and IITD-Palmprint training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

LLBPV improved on the Eigen, Fisher and SVM results by 13% – a highly significant improvement – allowing Eigen to achieve perfect accuracy when FPIR $> 30\%$. LBPH also achieves perfect accuracy but at FPIR $= 100\%$. Figure 9.16 shows Eigen outperforming the other classifiers in general when using three training samples.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLLBPV	0	0	0	0	0	0	2.3
3-FisherLLBPV	0	0	0	0	0	3.1	6.5
3-LBPHL	0	0	0	0	0	0	6.3
3-SVMLLBPV	0	0	0	0	0	2.1	4.7

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLLBPV	0	0	0	0	0	0	0
3-FisherLLBPV	0	0	0	0	0	1.0	1.5
3-LBPHL	0	0	0	0	0	0	0
3-SVMLLBPV	0	0	0	0	0	1.3	2.7

(d) FC FNIR values

Figure 9.16: Three fused Fei face and IITD-Palmprint training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

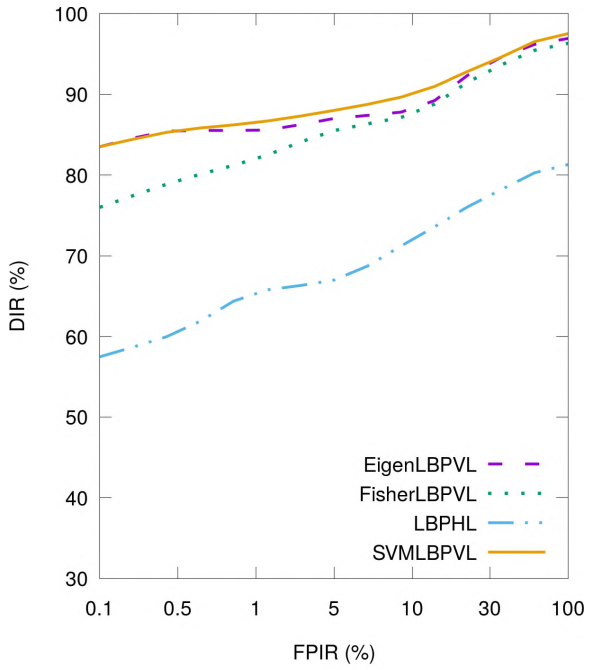
9.8 Face and Iris Feature-Fusion Recognition Results

9.8.1 Dataset A: Fusing Fei and CASIA-Iris-Lamp

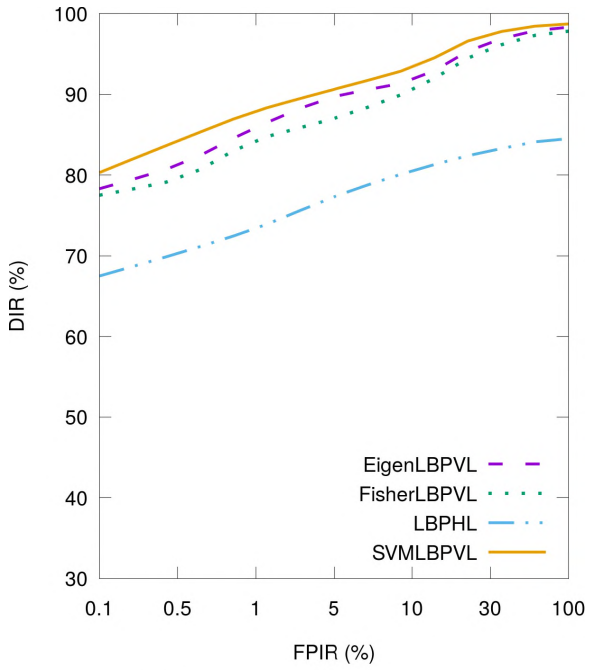
This fused dataset achieved its best results at 100×100 to 150×150 resolution. While the segmented resolution of CASIA-Iris-Lamp is 100×100, accuracy is not negatively affected when increased to 150×150, but yielded an additional 3% DIR (when FPIR = 100%). The non-negative effects of increasing iris images beyond their segmented resolution was also mentioned in Chapter 8 due to the tightly-spaced features of iris images. On the other hand, decreasing the resolution of iris images tends to result in a marked reduction in accuracy. Using a 50×50 resolution, the IR decreased by approximately 11% and 15%

when using FA and FC, respectively for Eigen, Fisher and SVM. The results of a 150×150 resolution are thus shown in the figures in this subsection for all classifiers.

Parameter tuning revealed that LBPH once again achieves its best performance with a 12-pixel radius and six neighbours. Referring to Figures 9.17 and 9.18, however, LBPH performs relatively poorly for both FA and FC when using either one or three training samples. Omitting the LoG filter reduced the accuracy even further and it was thus concluded that the LBPH accuracy cannot be improved for this dataset. SVM achieves the best accuracies in all tests. Furthermore, for the first time, Eigen and Fisher perform better when using FA, although this is not significant.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0	2.0	3.1
1-FisherLBPVL	0	0	0	0	0.5	2.0	3.6
1-LBPHL	0.5	0.9	1.3	1.7	4.7	10.5	18.7
1-SVMLBPVL	0	0	0	0	0	1.3	2.5

(c) FA FNIR values

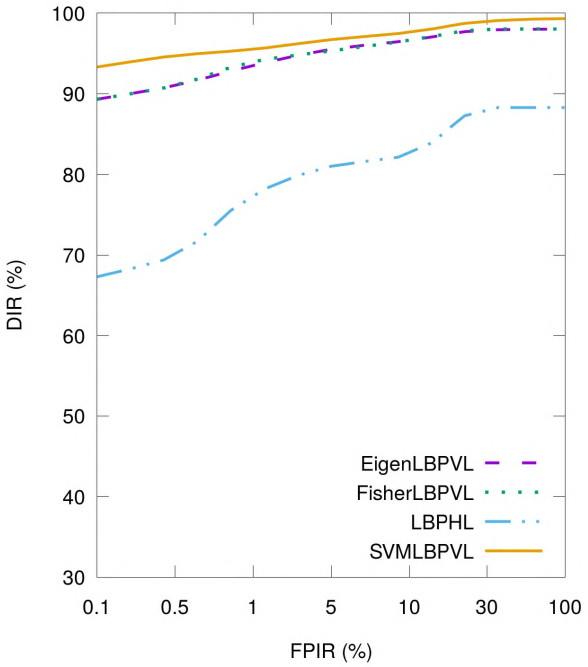
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	0	0	2.7
1-FisherLBPVL	0	0	0	0.5	0.5	1.3	3.2
1-LBPHL	0	0	0	0	0.5	6.3	15.5
1-SVMLBPVL	0	0	0	0	0	0	2.7

(d) FC FNIR values

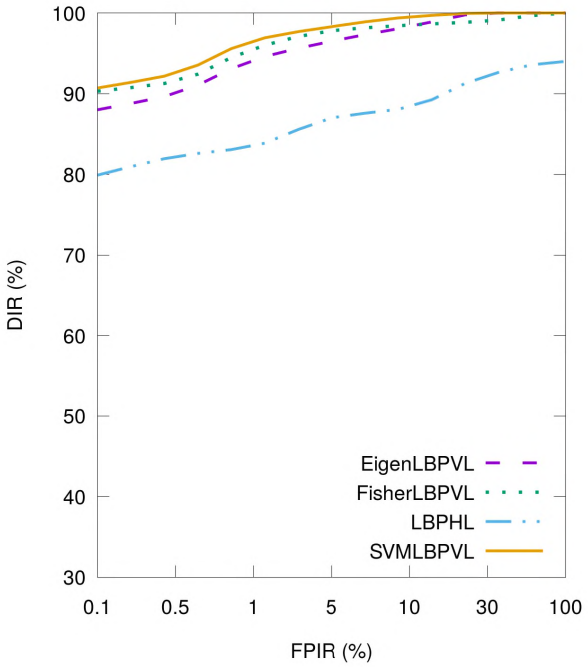
Figure 9.17: One fused Fei face and CASIA-Iris-Lamp training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Figure 9.18a shows that Eigen, Fisher and SVM achieve a higher DIR when FPIR $< 1\%$,

when using FA with three training samples. However, according to Figure 9.18b perfect accuracy is achieved only when using FC for Eigen, Fisher and SVM. LBPH performs poorly showing a similar trend to that identified in the one training sample results for this dataset and for the other bimodal experiments with Fei face as a component. An exception occurred when fusing Fei face with any fingerprint component, which resulted in LBPH being the best performer.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.0	2.0
3-FisherLBPVL	0	0	1.3	1.3	2.0	2.0	2.0
3-LBPHL	0	0	1.0	2.0	3.5	7.7	11.7
3-SVMLBPVL	0	0	0	0	0	0.7	0.7

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	0
3-FisherLBPVL	0	0	0	0	0	0	0
3-LBPHL	0	0	0	0	0	1.0	6.0
3-SVMLBPVL	0	0	0	0	0	0	0

(d) FC FNIR values

Figure 9.18: Three fused Fei face and CASIA-Iris-Lamp training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Doubling the number of classes (duplicate results from Dataset B) once again yields similar accuracies. That is, when using one training sample, a 2% lower average DIR for LBPH was recorded, while for Eigen, Fisher and SVM only a 0.5% lower average DIR was recorded. When using three training samples the negative effect on classifier performance caused by more classes is lowered to 0.5% for LBPH and 0.3% for the other classifiers.

These results, however, assume that the average quality of the additional classes is similar to the first half of the classes.

9.8.2 Dataset D: Fusing Fei Face and IITD-Iris

For this dataset, an optimal image resolution of 150 × 150 was found for all classifiers. However, results obtained using resolutions between 75 × 75 and 150 × 150 varied by a maximum of only 1%. Results in Table 9.6 show that all classifiers achieved high accuracy when using only a single training sample. SVM yields the best accuracies for both FA and FC, followed by Eigen and Fisher. LBPH performs relatively poorly when using FA and once again achieves the lowest accuracy when using FC, albeit by a lower margin.

Table 9.6: Results on face-iris fusion at 0% FPIR and 0% FNIR using three training samples on Dataset D.

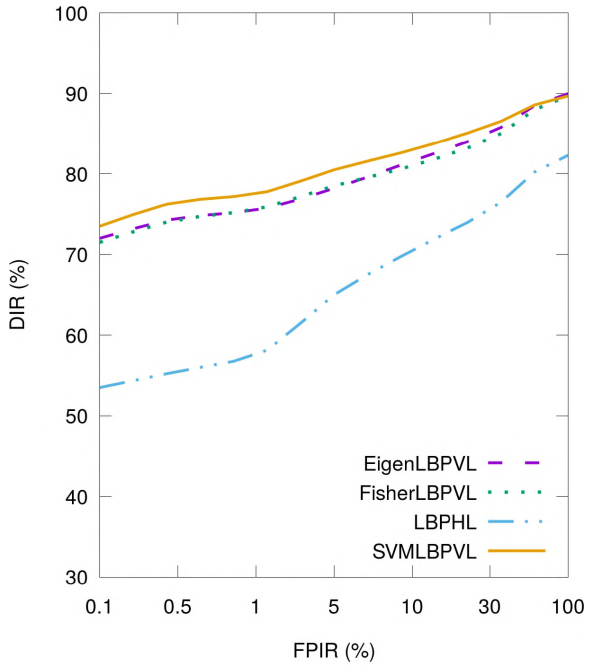
Classifier	FA	FC
1-EigenLBPVL	97.7	98.5
1-FisherLBPVL	96.3	98.5
1-LBPHL	93.5	97.0
1-SVMLBPVL	99.0	99.3

9.9 Palmprint and Iris Feature-Fusion Recognition Results

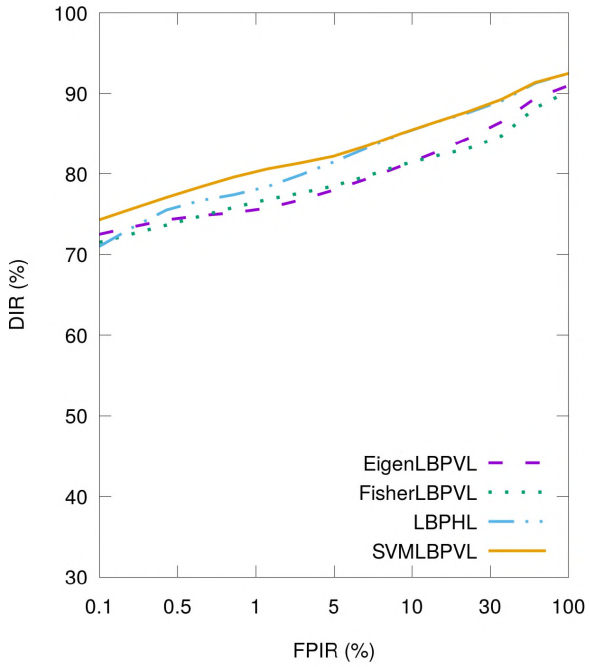
9.9.1 Dataset A: Fusing CASIA-Palmprint and CASIA-Iris-Lamp

For this datased, the optimal image resolution for Eigen, Fisher and SVM was found to be 100 × 100 as this resolution yields accuracies for the palmprint within the optimal range resulting in reasonable compromise for resizing, unlike the fingerprint as discussed in Section 9.6. However, LBPH preferred a 150 × 150 image resolution per feature vector.

Figure 9.19 shows that the SVM performs the best on average for FC, while the other classifiers achieve similar accuracies to SVM. On the other hand, LBPH performs poorly for the FA method (even though Fei face is not a fusion component).



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	1.0	1.0	1.0	2.3	2.3	5.5	10.0
1-FisherLBPVL	1.0	1.0	1.0	2.3	2.3	6.5	10.3
1-LBPHL	0	0	0	1.0	3.3	11.5	21.7
1-SVMLBPVL	0	0.5	0.5	2.3	3.0	5.0	10.3

(c) FA FNIR values

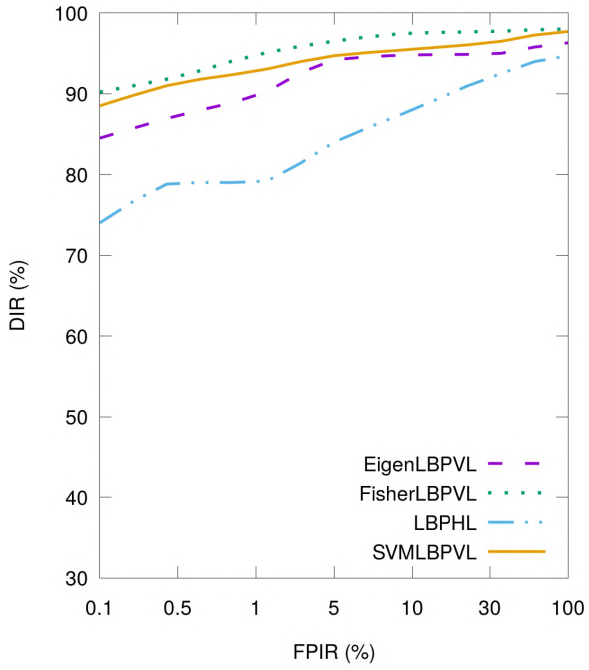
Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	1.0	1.0	1.0	3.3	5.5	9.0
1-FisherLBPVL	0.5	1.0	1.0	2.3	3.3	6.5	9.7
1-LBPHL	0.5	0.5	0.5	0.5	2.3	3.0	7.5
1-SVMLBPVL	0	0	1.3	1.3	2.3	2.7	7.5

(d) FC FNIR values

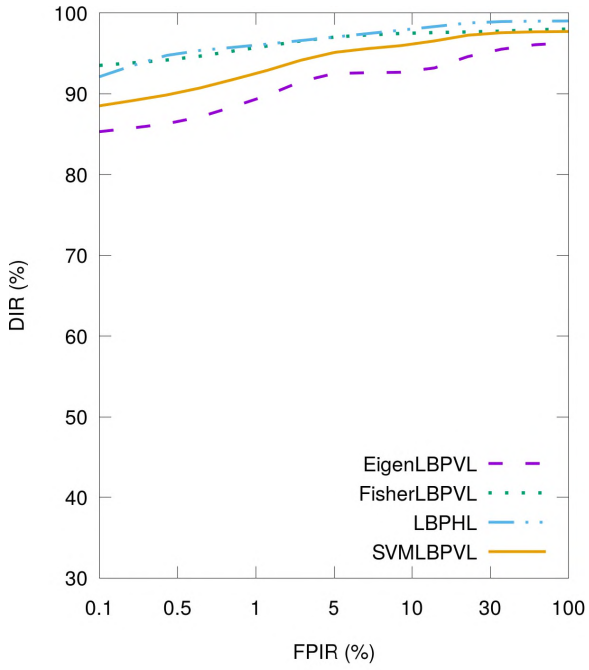
Figure 9.19: One fused CASIA-Iris-Lamp and CASIA-Palmprint training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

When using three training samples (as shown in Figure 9.20), Eigen, Fisher and SVM perform fairly similarly for FA and FC. LBPH results are again significantly better when using FC with three training samples, while Eigen, Fisher and SVM results improve to a lesser extent.

Trends similar to those observed at the end of Section 9.8.1, recurred when testing on this dataset with double the number of classes when comparing results using one and three training samples. However, in this case LBPH's average DIR (decreased by 0.4%)



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.0	3.7
3-FisherLBPVL	0	0	0	0	0	0	2.0
3-LBPHL	0	0	0	2.0	3.0	3.5	5.3
3-SVMLBPVL	0	0	0	0	0	0.5	2.3

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	1.0	3.7
3-FisherLBPVL	0	0	0	0	0	0	2.0
3-LBPHL	0	0	0	0	0	0	1.0
3-SVMLBPVL	0	0	0	0	0	0	2.3

(d) FC FNIR values

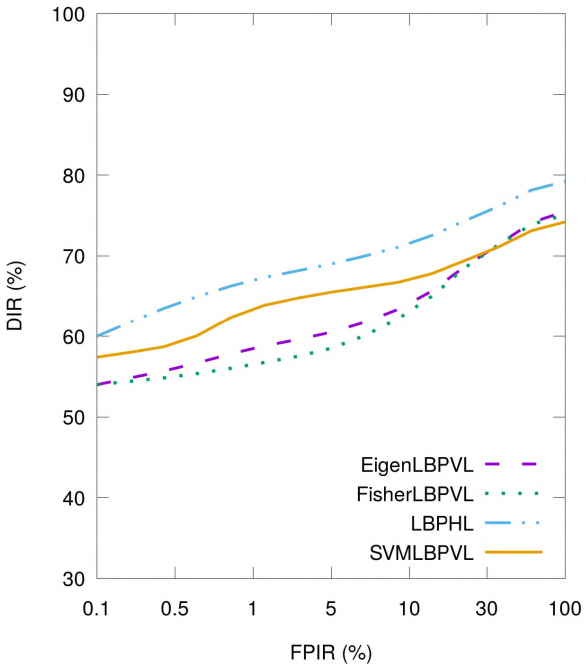
Figure 9.20: Three fused CASIA-Iris-Lamp and CASIA-Palmprint training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

was slightly less affected compared with the other classifiers (decreased by 0.6%) when using three training samples. Using one training sample all classifiers had an average DIR decrease of about 2%.

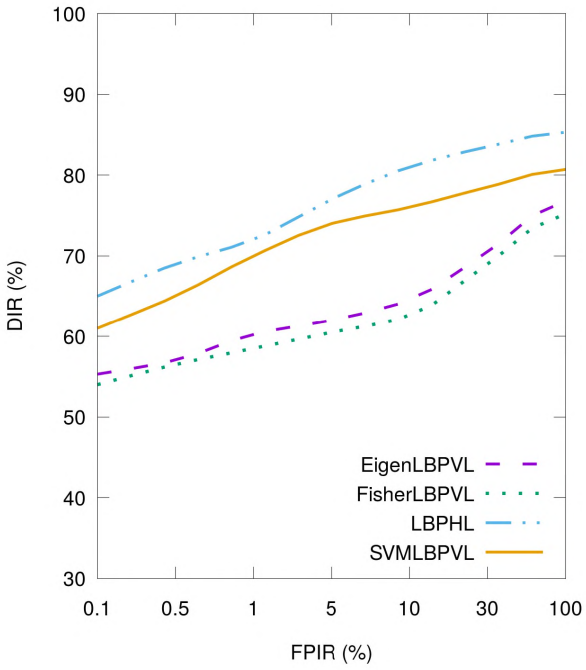
9.9.2 Dataset C: Fusing IITD-Palmprint and CASIA-Iris-Lamp

The optimal image resolution for Eigen, Fisher and SVM was found to be 100×100. However, it should be noted that results using resolutions 50×50 and 150×150 yielded maximum DIR values (FPIR = 100%) that were lower by 5% and 2%, respectively. DIR values at average FPIR, similarly, yielded lower results. LBPH, whose optimal

resolution remained 150 × 150, achieves its best performance using a 12-pixel radius and six neighbours when using FC. However, the FA method preferred a 2-pixel radius, as this improved maximum DIR by 8%. LBPH outperforms all classifiers for both FA and FC and achieves its best results when using the FC method, as shown in Figure 9.21.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0	0	0	0	2.5	9.0	24.5
1-FisherLBPVL	0	0	1.5	1.5	3.0	11.0	24.9
1-LBPHL	0	0.5	1.5	2.0	3.3	11.2	20.8
1-SVMLBPVL	0	0	0.5	2.3	4.1	12.3	25.8

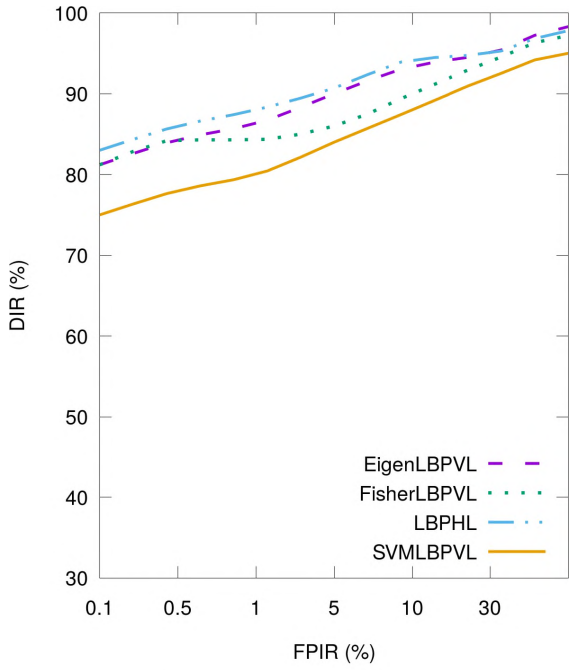
(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
1-EigenLBPVL	0.5	0.5	0.5	2.0	3.5	11.0	23.2
1-FisherLBPVL	0	0.5	0.5	2.0	3.3	10.8	24.8
1-LBPHL	0	0	0	0	0	4.4	14.7
1-SVMLBPVL	0	0	0	0	1.5	5.3	19.3

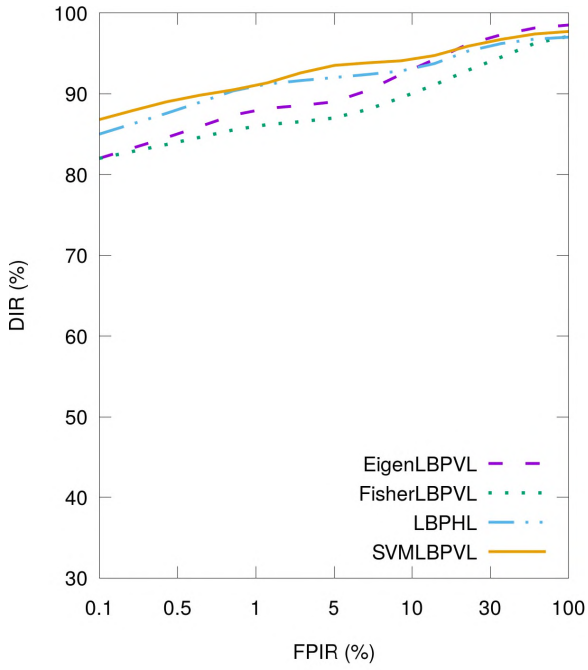
(d) FC FNIR values

Figure 9.21: One fused IITD-Palmprint and CASIA-Iris-Lamp training sample on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

Referring to Figure 9.22, when using three training samples for the FC method, Eigen achieves the highest DIR at FPIR = 100%, while LBPH and SVM achieve the best DIR average values. Overall, individual classifier results are not vastly different when using three training samples, unlike the case when using one training sample.



(a) DIR vs. FPIR using FA



(b) DIR vs. FPIR using FC

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	1.7
3-FisherLBPVL	0	0	0	0	0	1.0	2.8
3-LBPHL	0	0	0	0	0	0.8	2.2
3-SVMLBPVL	0	0	0	0	0.5	2.5	4.9

(c) FA FNIR values

Classifier	0.1	0.5	1	5	10	30	100
3-EigenLBPVL	0	0	0	0	0	0	1.5
3-FisherLBPVL	0	0	0	0	0.5	1.5	2.9
3-LBPHL	0	0	0	0	0	1.0	3.0
3-SVMLBPVL	0	0	0	0	0	0	2.3

(d) FC FNIR values

Figure 9.22: Three fused IITD-Palmprint and CASIA-Iris-Lamp training samples on remaining test images: (a) and (b) DIR versus erroneously identified impostors (FPIR); (c) and (d) FNIR for given FPIR values.

9.9.3 Dataset D: Fusing CASIA-Palmprint and IITD-Iris

The optimal image resolution of 150 × 150 was confirmed for all classifiers. However, resolutions between 75 × 75 and 150 × 150 varied by a maximum of only 2%. LBPH prefers a 2-pixel radius when using FA and a 12-pixel radius when using FC.

Table 9.7: Results on palmprint-iris fusion at 0% FPIR and 0% FNIR using three training samples on Dataset D.

Classifier	FA	FC
1-EigenLBPVL	96.3	97.5
1-FisherLBPVL	96.0	97.0
1-LBPHL	95.5	98.3
1-SVMLBPVL	95.3	99.0

Table 9.7 shows that all classifiers achieve high accuracy when using only a single training sample, similar to the Fei face and IITD-Iris dataset combination. SVM again achieves the best accuracies for both FA and FC. LBPH outperforms Eigen and Fisher marginally when using the FC method.

9.10 Discussion of Bimodal Results

Datasets A and B contained the most challenging bimodal combinations, some of which were duplicated in Dataset B but with double the number of classes. Nevertheless, the feature-classifier combinations showed promise as bimodal fusion definitively improved the accuracy, while also scaling well on these challenging datasets when classifying double the number of classes in Dataset B.

The Fei face and IITD-Palmprint combination in Dataset C was a special case in that Eigen (especially) and Fisher yielded significantly better accuracies than SVM and LBPH. This special case involved the use of the LLBPV feature extractor instead of LBPVL, which was used in the majority of the experiments. LBPH in particular performed poorly on this dataset.

Dataset D only required one training sample to reach near-perfect accuracies, except with FVC2004 as a component. Specific classifier-dataset results are discussed next.

Eigen, Fisher and SVM performed optimally when using lower resolutions on combinations involving the fingerprint or palmprint. A 50×50 to 65×65 resolution generally yielded optimal performance when either the fingerprint or palmprint – with their relatively sparse features of distinct size (ridges or principal lines) were involved in a fusion with one of the other modalities. However, this was not always the case with the palmprint, where specifically CASIA-Palmprint fused effectively at the preferred (and higher) resolution of the face and iris components. This was due to the relatively high segmented resolution of 200×200 for this dataset, compared with the 100×100 segmented resolution for all fingerprint datasets. Intuitively, fusing modalities whose components have substantially

different optimal resolutions should perform best on an equally compromising resolution with respect to individual performance. This is the case with medium resolutions often used in combinations involving the palmprint or the iris, depending on the segmented resolution. High resolutions always yielded the best results for face and iris combinations, while low resolutions were ideal for fingerprint and palmprint combinations. Minor exceptions occurred when two datasets were fused with large differences in DIR ($> 10\%$ on average FPIR value) and when the dominant component has a near perfect accuracy, such as the case with Dataset D, and to a lesser degree Dataset E, due to using FVC2004 as a fusion component in both cases. However, this varies by case as a classifier often favours dominant features, and therefore, for Datasets D and E, the densely-spaced features of PUT face and IITD-Iris were sufficient to not result in a significant drop in the fused accuracy.

Compromising the accuracy of the significantly better performing component is not ideal as was the case with Datasets D and E involving fusion of the FVC2004 Fingerprint and PUT Face or IITD-Iris datasets. This was also the only bimodal combination in Dataset E that did not produce perfect accuracy when using a single training sample. This is due to the FVC2004 fingerprint dataset yielding the worst accuracies of all single modality datasets. Fusing datasets with large differences in quality is thus discouraged. It is, however, noted that the HEL extractor performs better in such cases.

Since the CASIA-Palmprint dataset had a relatively high segmented resolution of 200×200, it was less affected in terms of performance than other fingerprint and palmprint datasets. While the face was markedly less sensitive to resolution reduction, it only slightly preferred the default segmented resolutions, typically 150×150. This is due to texture patterns of relatively variable size (contours and pores compared to eyes and nose). The iris was similarly adaptable to resizing, but only when enlarging the images, as reducing the already fine texture pattern resulted in loss of discriminatory information. An exception was with the IITD-Iris dataset, which was captured using a low DPI sensor resulting in less fine textures. Face and iris combinations thus perform optimally at segmented resolutions. Eigen, Fisher and SVM net higher improvements from fusion than LBPH suggesting that they aggregate features more effectively. However, they are

not as favoured when optimal resolutions are vastly different.

An advantage of LBPH is that it generally prefers the default segmented resolution of 150× 150 and is less affected by differences in resolution of various bimodal components than other classifiers. The consistency of resolution size is a promising trait of LBPH as it allows for greater compatibility among image-based modalities. However, this is associated with frequent additional parameter tuning for a particular modality and resolution. This means that generally more preparation is required to achieve optimal performance in real-world applications. Furthermore, LBPH performs poorly when using FA on face, palmprint and iris bimodal combinations. The particular combination of the face and iris results in poor performance under all tested conditions (using LBPH). An exception is the palmprint and iris, as LBPH is favoured when the fingerprint or palmprint is a component of the fusion.

Some classifier trends change when comparing single modality experiments with their fusion counterparts, such as LBPH achieving poor DIR for FPIR $< 1\%$ in single modality experiments, but this no longer being the case when using fused modalities. Eigen and Fisher, however, performed poorly on the fingerprint and palmprint combinations as well as the single modality experiments. Fusing by FC outperforms FA in all but face and iris fusion. Furthermore, SVM achieves similar performance when using either FA or FC for the palmprint and iris. SVM performs consistently for both individual and fused datasets and achieves the best performance for fused datasets when using the FC method. This is especially the case when the fingerprint is one of the fused components. Eigen and Fisher achieve similar accuracies, although Fisher often performs better when using three training samples of good quality.

9.11 Summary

The individual datasets were fused in bimodal combinations, with their open-set identification accuracy tested using the same method as the experiments on the single modality datasets. The interactions of bimodal combinations were noted. Fusing the datasets using the FC method improved the accuracy over the best performing component even when

the individual performances of modalities were vastly different, except for two combinations. The first is the FVC2004 and IITD-Iris combination in Dataset D, which achieved approximately a 3% lower DIR on average than the dominant IITD-Iris component for all classifiers. For the FVC2004 and PUT combination in Dataset E, the 100% accuracy at FPIR = 0% using the PUT dataset dropped by 5% when fusing for LBPH and at most by 0.4% for the other classifiers. Excluding these two exceptions, the feature-classifier combinations implemented are promising since the improvements resulting from especially the LBPVL feature extractor on Eigen, Fisher and SVM is substantial (as shown in Section 5.3.2) and proved to be effective on all four biometric modalities. The results when using LBPVL and SVM were particularly consistent and the best performing on average FPIR, in the majority of experiments. The next chapter draws conclusions from the feature-classifier interactions and shows how the classes (individuals) affect performance of biometric systems in an effort to avoid loss of accuracy as is the case with the two combinations within Datasets D and E.

Chapter 10

Improving Feature-Fusion Accuracy through Weighted Modalities and Specific Guidelines

This chapter discusses the various conditions that affect image-based biometric recognition based on the extensive experimentation carried out in Chapters 4 to 9. To achieve optimal feature-fusion results under various conditions, weighted fusion is considered and results from final experimentation are reported on a per class basis. A number of feature-fusion guidelines are documented in an attempt to improve biometric fusion accuracy and applicability under a variety of conditions for future systems.

10.1 Trends and Commonalities Identified in Features and Classifiers

The analyses of the comprehensive set of experiments covered in previous chapters, including parameter tuning, are aggregated such that the respective favoured conditions of features and classifiers for both unimodal and bimodal biometric systems can serve as guidelines for future biometric systems. Some of these trends are documented below.

10.1.1 General Unimodal Observations

The following observations were noted throughout unimodal experimentation. The new LBPVL feature extraction approach allows Eigen, Fisher and SVM to clearly outperform most related systems. Open-set identification systems seem to benefit particularly from the use of SVM for effective discrimination against impostors. The following properties were noted that encourage high open-set identification accuracy:

- The image segmentation process should aim to capture all unique features for lower FNIR.
- Appropriate feature extraction after the segmentation process is imperative for Eigen, Fisher and SVM.
- High inter-class separation is dependent on both the classifier and segmented region.
- Low intra-class separation helps prevent overfitting of all classifiers and reduces impostor acceptance. This is especially beneficial to Fisher recognition performance.
- If any one of the above is poorly implemented, a low DIR is likely.

10.1.2 General Bimodal Observations

- Fusion lowers FNIR more than FPIR due to the increase in unique features and results in better inter-class separation.
- The FA and FC fusion methods have similar optimal resolutions per modality.
- FC outperforms FA in the majority of the bimodal experiments.
- SVM performs the best on average and is noticeably better than the other classifiers when using FC and when impostor rejection is imperative.
- Both FA and (to a lesser extent) FC can achieve lower accuracy than the dominant component when the quality of the two fusion components is unbalanced.

10.1.3 Classifier Preference in Image-based Biometric Systems

Experiments revealed that the Eigen, Fisher and SVM classifiers operate effectively at similar resolutions for specific modalities. The fingerprint and palmprint generally prefer low resolutions in the vicinity of 50×50, while the face and iris recognition systems prefer images closer to their segmented resolutions. LBPH also prefers the segmented resolutions and has relatively good performance (compared with the other classifiers) without feature extraction techniques such as LoG, and the proposed LBPVL and LBPL. However, LBPH is consistently a poor performer on face data relative to the other classifiers, regardless of the feature extraction algorithm used. It is noteworthy that many of the feature-level fusion related studies (Yao *et al.*, 2007, Ahmad *et al.*, 2010, Rattani *et al.*, 2011) do not use additional feature extraction techniques and rely solely on the classifier for feature reduction. While training and test speeds were measured for all classifiers, discussing of timings are beyond the scope of this research.

The following classifier-specific traits were also identified.

- Eigen: Poor results are achieved if features of the segmented image are not significantly reduced with feature extraction methods other than those inherent to Eigen (i.e., PCA). This classifier performs consistently when used after feature extraction methods such as LoG filters and other variants proposed in this research. However, Eigen has the slowest training time and largest space requirements of the four classifiers and is greatly affected by resolution variances.

- Fisher: Performance trends are similar to Eigen except for the following differences. Fisher often surpasses Eigen with good training data and is geared toward biometric systems that require low dimensionality data, thereby producing smaller training models, lower training times, and the best test time of the four classifiers. However, bad training data or high intra-class variance often leads to insignificant accuracy improvements or even worse accuracy when using more training samples (bad model scaling causes overfitting). This negative effect is lessened when inter-class variance is high. This classifier is also greatly affected by resolution variances.

- LBPH: This classifier achieves the best accuracy for all modalities when additional feature extraction is not performed before classification. It also consistently produces the best results for the fingerprint and palmprint. LBPH has the fastest training time with the smallest model size (fraction of the rest), but the slowest test time, which becomes much slower when increasing the neighbourhood parameter beyond six. LBPH does not scale as well as the other classifiers when given more training samples. It prefers the default segmented resolution in all cases. For both types of fusion, a large radius of 12 and six neighbours yields the best performance for all datasets. Similar parameters (when neighbours > 6) yield significantly lower recognition (and processing) performance for single modalities. LBPH produces relatively poor results for the face, but often classifies the best at lower FPIR values for FA. LBPH is preferred in identification systems that require regular class updating, as it does not require retraining data already present in the training model.

- SVM: The linear SVM that has been used in this research is known to handle high dimensionality data relatively well compared to other popular classifiers. Experiments revealed that it achieves marginally better performance over Eigen and Fisher when not using prior additional feature extraction, and yet its recognition of image-based biometric patterns improves significantly with the use of LBPVL. Additionally, it shows consistent performance across all single modalities and bimodal combinations. Since the multi-threaded linear version is used, training time is lower than for Eigen and Fisher, while the test time is second to Fisher. SVM performs well with only a single sample of training data, but scales similar to Eigen with more data. Moreover, SVM strongly prefers FC fusion in the majority of cases. Finally, it is less affected by resolution than the other classifiers.

Table 10.1: Summary of feature-fusion guidelines.

Stage	Name	Advantage	Disadvantage	Suggested Use
Feature Normalization and Extraction	Pixel Normalization	Reduces inconsistent lighting.	Not very effective for big changes in lighting.	All biometrics affected by lighting. The first step of quality enhancement.
	Histogram Equalization	Reduces inconsistent lighting.	Minor histogram distortion. Introduces some noise.	Biometrics affected by lighting. Histogram-shape invariant classifiers such as Eigen and Fisher.
	NL-means Filter	Denoises and preserves edges.	Can remove fine texture.	Use before Poincaré index and other algorithms that do not rely on fine texture but on a distinct collection of patterns.
	LoG Filter	Improves feature discrimination, before transforming features for classification.	Requires consistent lighting.	Remove noise in the upper and lower frequencies before fusion.
	Modified ELBP Operator	Minimises inconsistent lighting.	Introduces noise and thus does not benefit images with consistent lighting.	Biometrics affected by lighting. Combine with feature extractor such as LoG (LBPL) for classifiers that require low dimensionality for optimal fitting – particularly important when fusing features and often outperforms HEL.
	Modified ELBPV Operator	Same benefits as ELBP but without introducing noise.	Used as a feature extractor, LBPVL is outperformed by HEL in only one case, but gives improved results over LBPL.	Biometrics affected by lighting. Same as ELBP.
Feature Transformation and Classification	Eigen Classifier	LoG and modified ELBPV supplement this classifier.	Slow training on large datasets. Relatively high dimensionality.	Prefers small discriminative image regions for low dimensionality. Useful for both single and fused image-based biometrics.
	Fisher Classifier	Similar to Eigen, but with a higher potential accuracy with good data and lowers dimensionality.	Requires more consistent intra-class data than Eigen.	When dimensionality is too high for effective Eigen modelling and data quality is above average.
	LBPH Classifier	Robust to misalignment in general.	Benefits the least from feature extraction methods and performs relatively poorly on the face modality. Often requires parameter tuning per application.	General purpose classifier. Distinct texture patterns – fingerprint and palmprint. Applications with low storage requirements.
	SVM Linear Classifier	Similar to Eigen performance and often better single training sample performance. Faster training and testing than Eigen.	Effectively aggregating training samples is less trivial than the other classifiers.	Single training sample applications. When using FC fusion method.
Fusion Method	FA	Lower dimensionality – same vector size as single modality.	Lower accuracy and consistency.	Useful for larger dataset identification – high population performing millions of matches. Useful to reduce the dimensionality for fusing modalities with large optimal resolutions like face and iris.
	FC	Improves accuracy with consistency.	Higher dimensionality – increases training and test times.	Useful for high security and smaller population size.

10.2 DIR by Class

This section provides the SVM closed-set identification results per individual (class) as the number of correct matches and average IR per modality and fusion method. Moreover, these experimental results are correlated with visual sample quality indicators, such that weighted modality fusion can be effectively implemented.

The SDUMLA is a (true) multimodal biometric dataset consisting of the fingerprint, face, iris and various other biometrics of the same 106 subjects but lacking the hand/palmprint. The first 53 subjects and five samples of the SDUMLA fingerprint, face and iris and CASIA-Palmprint dataset are hence ranked individually and with fused bimodal combinations. Of these five samples, only a single training sample is used to predict the class of the remaining four samples, for all experiments that follow.

In Chapters 5 to 9, the effect of image quality on recognition accuracy was evaluated using datasets that include viewing angle, distance and/or lighting etc. This was based on their reported acquisition procedure or metadata. In this chapter, image quality observations are reported by using visual inspection because of the differing quality standards among various modalities. However, optimal resolution values per modality, that were systematically acquired during experimentation, are also provided in Table 10.2. The table shows that the fingerprint and palmprint generally prefer low resolutions in the vicinity of 50×50, while the face and iris recognition systems prefer images closer to their segmented resolutions. This is apparent in terms of individual and fused resolutions. Furthermore, the values in the table correspond with the observations recorded in Section 9.10.

The SVM's classification performance of individual samples per class is used in this section to determine the worst ranked classes per FA and FC. Each table per modality combination shows the number of correct matches per class for the FC method, in order of ascending rank, with a corresponding FA result (not necessarily in ascending order). Separate ranks per class are shown to the right with respect to FC and FA methods. Only the 10 worst results are provided per method. However, if FC achieves its best result (four correct) before reaching rank 10, the remaining results are displayed in order

Table 10.2: Optimal resolutions used for modalities and fused combinations.

Modality	Resolution
Fingerprint	50×50
Face	200×200
Palmprint	50×50
Iris	100×100
Fingerprint + Face	50×50
Fingerprint + Palmprint	50×50
Fingerprint + Iris	65×65
Face + Palmprint	50×50
Face + Iris	150×150
Palmprint + Iris	65×65

of ascending rank by FA. Therefore, classes that rank greater than 10 are indicated by "–" as an omitted result. The analysis per table of results is not repeated for duplicate classes of the same modality as there is no change. Where visual comparisons are made, the corresponding segmented images with original lighting are provided for the reader in Appendix A. In this chapter, the better performing bimodal component with an average IR that is 10% greater than the other modality is referred to as the dominant modality, while the latter is referred to as the weaker modality.

10.2.1 Fingerprint and Face

Table 10.3 shows the results for the fingerprint and face, where the FC and FA methods are compared. FC performs significantly better than FA for Class 47 as it achieves three correct matches compared to zero. The corresponding single modality face and fingerprint matches are four and zero, respectively. Visual inspection of the fingerprint image samples showed good quality test images, but the corresponding training image suffered from poor lighting, low clarity and misalignment. This trend was also recorded for the MCYT fingerprint tests, which are grouped according to fingerprint quality using hard constraints during acquisition instead of visually. The fingerprint samples were also inspected for Classes 23, 31 and 49, revealing that the training samples had good lighting consistency but they were all partials. Both FC and FA achieved poor accuracies for those classes. FC outperforms FA by 4%, on average for all classes. However, the high accuracy of the face

is unmatched by both fusion methods and is further investigated in later combinations involving the face. Only three face test images are incorrectly matched and this is due to the noticeable facial expression differences as shown in Appendix A (Figure A.4).

Table 10.3: IR per class for fingerprint-face fusion using FA and FC in ascending order of rank for FC.

Class	**Fingerprint** (84.4%)	**Face** (98.6%)	**FC** (91.9%)	**Rank**	**FA** (87.7%)	**Rank**
23	0	4	0	1	0	1
31	0	4	0	2	0	2
49	0	4	1	3	0	3
4	1	4	2	4	1	5
26	2	4	3	5	3	7
34	3	4	3	6	3	8
40	3	4	3	7	1	6
47	0	4	3	8	0	4
44	3	4	4	–	3	9
53	2	4	4	–	3	10

10.2.2 Fingerprint and Palmprint

The result with the greatest difference in Table 10.4 is for Class 31, as FC achieves four correct matches compared to only one for FA, indicating the palmprint classes are well fitted within the model. This is contrary to FC achieving zero correct in the previous test when fusing the same fingerprint class with the face instead of the palmprint. While both the face and palmprint have low intra-class variations in terms of alignment and lighting, all five face samples have varying facial expressions causing insufficient inter-class separation after fusion. In other words, introducing these inadequately fitted fingerprint samples – that share features with samples of other classes – detriments the classifier model. This result is of concern as it shows that fusing two modalities of conflicting quality can produce a lesser accuracy than the modality with the better quality. Considering Classes 4, 21 and 47, FC achieves two more correct than FA and performs better than FA for classes that are not in the top ten, to produce an overall 94.8% IR. Both fusion methods are outperformed by the single palmprint modality for Class 23. However, this is to a lesser extent than the test (Class 23) repeated in Table 10.3, but is still attributed

to the same reason, that is, the fingerprint samples are partials. The classes of interest are provided in Appendix A.

Table 10.4: IR per class for fingerprint-palmprint fusion using FA and FC in ascending order of rank for FC.

Class	Fingerprint (84.4%)	Palmprint (76.9%)	FC (94.8%)	Rank	FA (86.3%)	Rank
8	4	2	2	1	2	5
23	0	3	2	2	1	1
2	3	3	3	3	3	8
7	4	3	3	4	3	9
26	2	3	3	5	3	10
35	1	2	3	6	2	7
40	3	3	3	7	3	–
42	3	1	3	8	3	–
47	0	2	3	9	1	3
31	0	4	4	–	1	2
4	1	4	4	–	2	4
21	4	1	4	–	2	6

10.2.3 Fingerprint and Iris

Table 10.5 shows that the fingerprint and iris achieve similar average accuracies. FC achieves a perfect score when using a high quality iris image with a low quality fingerprint image. The result of a perfect score or improved accuracy over the dominant modality is a recurring trend and is thus referred to as A-rank fusion. When the fusion result is equal to the dominant modality, it is referred to as B-rank fusion. Visual inspection of the iris training image and the test sample shows high similarity. This corroborates the argument that FC can perform on par with the better performing individual modality for a particular class. On the other hand, FA often underperforms under these conditions to the point of recurring reduction in accuracy. Achieving fewer matches than the dominant modality is referred to as F-rank fusion. A clear exception is observed for Class 23, where FA obtains a perfect score under the same apparent conditions. This is significant because Class 23 has consistently obtained F-rank fusion results until this observation. Visual inspection reveals that the Class 23 iris images are in fact of high quality, thus

confirming the unexpected good performance by FA. Another possibility is that besides the subjectiveness of visual inspection, perceiving data conditions such as fine textures and modality differences can be physically limited by the human senses and image space. FC significantly outperforms FA for the remaining results, and is on average 8.5% better. It is worth noting that both fusion methods generally improve the average accuracy and the tables shown in this chapter are worst case results aimed at identifying suboptimal conditions of both unfused and fused data for classification. This is contrary to fusing dominant and weaker modalities, where the average accuracy of the fused result is lower than the dominant modality.

Table 10.5: IR per class for fingerprint-iris fusion using FA and FC in ascending order of rank for FC.

Class	**Fingerprint** (84.4%)	**Iris** (84.9%)	**FC** (94.8%)	**Rank**	**FA** (86.3%)	**Rank**
47	0	4	1	1	0	1
4	1	2	2	2	1	2
31	0	2	2	3	2	4
23	0	4	3	4	4	–
26	2	4	3	5	1	3
42	3	4	3	6	4	–
44	3	2	3	7	3	9
35	1	4	4	–	2	5
27	4	1	4	–	3	6
38	4	4	4	–	3	7
40	3	4	4	–	3	8
49	0	4	4	–	3	10

10.2.4 Face and Palmprint

The disadvantage of fusing a dominant modality with a weaker modality is clearly shown in Table 10.6 as F-rank fusion results are again prevalent as was the case when fusing the fingerprint and face.

Table 10.6: IR per class for face-palmprint fusion using FA and FC in ascending order of rank for FC.

Class	Face (98.6%)	Palmprint (76.9%)	FC (92.5%)	Rank	FA (84.4%)	Rank
20	4	1	1	1	1	1
43	4	1	1	2	1	2
1	4	3	3	3	3	9
18	4	3	3	4	3	–
21	4	1	3	5	2	5
26	4	3	3	6	3	–
36	4	3	3	7	3	–
38	2	2	3	8	3	7
39	3	3	3	9	3	–
40	4	3	3	10	3	–
2	4	3	4	–	2	3
8	4	2	4	–	2	4
34	4	3	4	–	2	6
41	4	2	3	–	2	8
12	4	3	4	–	3	10

10.2.5 Face and Iris

Table 10.7 shows that FC achieves a significantly higher accuracy than FA, but is still outperformed by the face with a single match less for Classes 4 and 27.

Table 10.7: IR per class for face-iris fusion using FA and FC in ascending order of rank for FC.

Class	Face (98.6%)	Iris (84.9%)	FC (94.8%)	Rank	FA (86.3%)	Rank
4	4	2	3	1	2	1
27	4	1	3	2	3	2
16	4	0	4	–	4	–
30	4	0	4	–	4	–
22	4	1	4	–	4	–
41	4	2	4	–	4	–
44	4	2	4	–	4	–
38	2	4	4	–	4	–

10.2.6 Palmprint and Iris

Table 10.8 shows that FC achieves A-rank fusion results for the palmprint for Class 16, while FA continues to show poor performance. On the other hand, FA achieves A-rank fusion results for Class 12. Both methods perform poorly for Classes 20 and 21. FC generally performs better than the individual modalities for the rest and on average by 4.7% showing some potential for modalities with unbalanced quality. However, this increase is less significant than the large improvement introduced by fingerprint and palmprint fusion, which has similarly unbalanced component quality. This indicates that the factor of feature synergy noted in Chapter 9 also plays a role. Furthermore, FA achieving a lower accuracy than the iris on average should be well noted as a further indication of lack of synergy.

Table 10.8: IR per class for palmprint-iris fusion using FA and FC in ascending order of rank for FC.

Class	Palmprint (76.9%)	Iris (84.9%)	FC (89.6%)	Rank	FA (82.5%)	Rank
20	1	3	1	1	1	3
42	1	4	1	2	2	10
10	1	2	2	3	1	2
21	1	4	2	4	1	4
1	3	4	3	5	3	–
2	3	2	3	6	2	6
4	4	2	3	7	3	–
12	3	2	3	8	4	–
13	2	3	3	9	2	7
16	2	0	3	10	0	1
43	1	4	3	–	1	5
32	3	4	4	–	2	8
41	2	2	3	–	2	9

10.3 Weighted Feature Fusion

Selectively fusing biometric modalities of some classes may be an effective means for solving scenarios such as universality or prevalent weaker modalities, but simply discarding

the weaker modality may not yield the optimal result. This section investigates weighting bias that can provide a result equal or better than that of the dominant modality.

The weighting method proposed in this research adjusts the contrast gain of both images in proportion to their original contribution for FA and FC. The default contributions of the fusion components used in the bimodal experiments (Chapter 9) for FA is 50% each since they are averaged. In the case of FC, each modality contributes 100% and for the sake of simplicity, the contrast values given in this section for FC are normalized to that of FA by dividing them by two.

Table 10.9 results show that weighting the fingerprint and face in the ratio 30:70 can increase the fused IR above the dominant modality IR. This was further investigated and found to be true for ratios up to 10:90. This is attributed to the extra match in Class 38 due to the positive contribution of the fingerprint, as the face performed poorly for this class. These results were achieved by using the preferred face resolution of 200×200, as the preferred fingerprint resolution of the original experiment 50×50 was no longer favoured.This was to be expected due to the face features having a greater effect on classification. Retesting at a face image contrast of 40% or lower enabled Classes 38 and 39 to achieve perfect scores. Furthermore, a 100% weighting for the face produced the same results for both FA and FC as the individual face modality. The fact that optimal weighting values for both FA and FC were the same should be noted.

Table 10.9: IR per class for fingerprint and face fusion weighted at 30:70, using FA and FC, for the top three worst ranked FC results followed by the worst performing face classes.

Class	**Fingerprint** (84.4%)	**Face** (98.6%)	**FC** (91.9%)	**FA** (87.7%)	**Weighted FC** (99.06%)	**Weighted FA** (99.06%)
23	0	4	0	0	4	4
31	0	4	0	0	4	4
49	0	4	1	0	4	4
38	4	2	3	3	3	3
39	4	3	3	3	3	3

The weighted fusion results in Tables 10.10 and 10.11 show improvements for Classes 38 and 39 identical to Table 10.9.

Table 10.10: IR per class for face and palmprint fusion weighted at 30:70, using FA and FC, for the top three worst ranked FC results followed by the worst performing face classes.

Class	Face (98.6%)	Palmprint (76.9%)	FC (92.5%)	FA (84.4%)	Weighted FC (99.06%)	Weighted FA (99.06%)
20	4	1	1	1	4	4
43	4	1	1	1	4	4
1	0	4	1	0	4	4
38	4	2	3	3	3	3
39	4	3	3	3	3	3

Table 10.11: IR per class for face and iris fusion weighted at 30:70, using FA and FC, for the top three worst ranked FC results followed by the worst performing face classes.

Class	Face (98.6%)	Iris (84.9%)	FC (94.8%)	FA (86.3%)	Weighted FC (99.06%)	Weighted FA (99.06%)
4	4	2	3	2	4	4
27	4	1	3	3	4	4
16	4	0	4	4	4	4
38	4	2	3	3	3	3
39	4	3	3	3	3	3

Since Classes 38 and 39 performed poorly due to bad face test samples, optimal weights were manually assigned to the remaining individuals to simulate a dynamic weighting process. The results (not shown) revealed that using a dynamic weighting strategy could result in perfect accuracy for Classes 38 and 39. The manual weighting ratio assigned for all three tests was 70:30 (dominant modality:weaker modality), simply a reverse bias, together with adoption of the preferred resolution of the respective weaker modality rather than 200×200.

Employing a dynamic weighting strategy, also known as user-specific fusion schemes (Ross *et al.*, 2006), can be effective with accumulated training data and results over time per user. This strategy is implemented for its simplicity. Genetic algorithms could have been used to fine tune individual weights, but that level of fine tuning is deemed beyond the scope of this research. Furthermore, before constructing multimodal biometric systems that use more advanced weighting strategies, security issues need to be considered. For example, an impostor could attempt to circumvent a biometric system by targeting specific users with known problems within their biometric data. An example of such an individual could be

one with hands burnt beyond recognition, prompting the dynamic multimodal biometric system to use a less secure biometric, alternative means of authentication or parameter adjustment that can be planned for by the impostor. Accessing the dynamically weighted features and new parameters could provide further insight into ascertaining optimal fusion schemes, but as stated previously, is beyond the scope of this research.

With the positive results achieved on weighting the datasets in this chapter, the two combinations identified in the bimodal experiments were also weighted in an attempt to achieve equal or better accuracy than the respective dominant modalities. Various ratios were experimented with and 30:70 was again found to be ideal for the FVC2004 and IITD-Iris combination as it achieved perfect accuracy. Increasing the ratio above 80:20 resulted in a lower improvement for the FVC2004 and IITD-Iris combination, confirming that the weaker modality still has discriminative properties. Since the PUT face dataset achieved a perfect accuracy, simply using it at 100% weight gave a perfect accuracy. However, the ratio could be lowered to 60:40 before a reduction in accuracy occurred.

Bimodal fusion resulted in perfect accuracies after weighting, exceeding the accuracies of even the dominant individual modalities. Combinations consisting of more than two modalities were therefore not investigated, as further accuracy gains could not be achieved.

10.4 Discussion of Class Results

This section showed that large variations in image segmentation leads to poor results. This was determined with visual comparisons where a bad training sample often caused no correct identifications – as opposed to a bad test sample only affecting itself – for a particular class. Experiments were run using the novel LBPVL feature extractor followed by SVM classification as this produced the best results. Inspection of class images revealed that LBPVL is not negatively affected by dynamic lighting nor severely underlit conditions. Weighted versions of FA and FC were also tested, where reducing the contrast of the weaker modality led to a positive contribution of the remaining features to the

fused result, enabling feature fusion of non-ideal modalities without compromising recognition performance (A-rank fusion). The ratio applied was 30:70, where 30% applies to the weaker modality to lessen its contribution during classification. Weighted fusion was also applied to the FVC2004 and IITD-Iris combination in Dataset D and the FVC2004 and PUT combination in Dataset E using the same ratio as the SDUMLA experiments. This resulted in both combinations achieving perfect accuracy. In fact, both FA and FC achieved (the same) improved results over the respective dominant modalities. This indicates that weighted FA can lead to optimal fusion result as it has the potential to achieve similar results using half the dimensionality of weighted FC.

10.5 Summary

This chapter discussed the various conditions that affect image-based biometric recognition. Class-specific results were given for a multimodal dataset consisting of four biometric modalities, one of which was the face that contained ideal frontal data resulting in it being the dominant modality. Weighted fusion further improved the results of FA and FC such that they surpassed the performance of the dominant modality, caused by fusing non-ideal modalities. Fusion guidelines at the feature level were given with the aim of providing future biometric systems with a methodical approach to solving the dynamic scenarios often posed by multimodal biometric applications.

Chapter 11

Conclusion

11.1 Summary of this Research

This thesis focussed on exploring the wide range of approaches necessary for developing effective multimodal biometric fusion systems at the feature level. The research statement in Chapter 1 sets this in motion by considering a detailed study of feature-classifier combinations on four image-based modalities individually, followed by their bimodal combinations. The various biometric concepts pertaining to the biometric modalities were highlighted and related studies were reviewed. The gaps found within the algorithms used in related systems were considered before establishing four proposed biometric systems. The proposed systems thus contained modified or novel algorithms as appropriate. Although several feature extraction algorithms were considered, only results from the best performers were provided in the experiments. The same feature extraction method was applied to four different modalities for compatibility purposes of feature fusion, which was subsequently followed by classification using different methods.

The experimentation sections of the thesis were conducted as follows. The four proposed biometric systems were first visually validated as appropriate to image-based modalities. The noted observations were on the correct segmentation per biometric modality, thus ensuring a consistent image region was selected, before performing feature extraction and

classification. Parameter tuning was performed during initial experimentation where it was determined that datasets of the same modality generally have similar or even identical optimal parameter values. Specific sets of parameters were determined as being optimal for individual modalities (unimodal), while bimodal combinations often required a larger radius than the neighbourhood size for LBPH, resulting in sparser texture descriptors to deal with the increased texture information. This was always required when using the higher dimensional FC method and occasionally required when using the FA method on bimodal combinations containing at least one fused component with finer texture. On the other hand, Eigen, Fisher and SVM classifiers used the same parameters in unimodal and bimodal experiments, and instead, required specific image resolutions depending on the modality.

One of the fingerprint and face fused combinations resulted in a reduced DIR (99.7%) compared to the DIR (100%) of the face component due to fusing two highly unbalanced datasets in terms of quality. This was due to the PUT face dataset (dominant modality) achieving a perfect accuracy when using left pose 30° as the only training sample, while the FVC2004 fingerprint dataset achieved a best DIR of 38.25% at the corresponding FPIR of 0%. Another unbalanced bimodal dataset consisting of the FVC2004 and IITD-Iris combination, resulted in a 3% decreased DIR compared with the dominant IITD-Iris component. Both of these unbalanced combinations achieved these results using FC, while FA resulted much greater accuracy reduction.

The fingerprint and palmprint preferred smaller image resolutions – typically between 50×50 and 65×65 – while the face and iris preferred the default segmented resolutions – typically between 100×100 and 200×200. This was due to the size of the pattern of fingerprints (ridges) and palmprints (principal lines) being sparse compared with the finer features of the face and iris which included speckles.

Bimodal fusion was also evaluated based on the performance per individual (rank), where it was shown that weighted fusion solves the problem of fusing unbalanced data by achieving greater or equal accuracy to the dominant modality when using either FA or FC. As a concept, it was also demonstrated that the accuracy can be further improved by em-

ploying dynamic weighting, which automatically considers the weight to be assigned to a modality on a per class basis, allowing modalities to complement each other effectively.

11.2 Contributions of this Research

This research has made several contributions to the field of multimodal biometrics. The most significant contribution relates directly to the research statement by providing insight on the interactions of a variety of feature-classifier combinations on four different image-based modalities. This was illustrated through the commonalities and differences of approaches on four modalities, and results were extended to bimodal fusion combinations of the four modalities.

A further contribution was made to the face biometrics field with the introduction of a novel face segmentation algorithm called VRA, enabling larger pose angles to be recognized with limited training data. This could be particularly useful in surveillance security where pose angles are vastly different due to the uncontrolled nature of this application.

Another important contribution made was a methodology that can be used to optimize, or in some cases modify, computer vision techniques in biometric applications. Two new feature extractors, LBPVL and LBPL, were particularly useful in combating bad lighting conditions that are especially prevalent in image-based biometrics, while also effectively reducing feature dimensionality. While both improved classification results compared with HE, HEL and the other methods mentioned in the literature (Chapter 3), LBPVL proved to be most effective overall as it was efficient under various lighting conditions without contributing noise as in the case of LBPL.

All contributions and general observations that resulted from addressing the research statement were collated and presented as foundational guidelines for biometric fusion at the feature level. These guidelines are intended as a starting point for future studies to construct efficient biometric systems and upon implementation extend the guidelines based on additional scenarios.

With the conclusion of this comprehensive study on multimodal biometric fusion at the feature level, all research objectives have been met.

In addition, this research highlights the relevance of computer vision in solving gaps within the biometrics field of study. The knowledge passed on in this research serves as a foundation and aid to further advancements in multimodal biometrics at the feature level and related fields.

11.3 Future Work

Three directions for future work are provided below.

Dynamic weighting approaches can be automated with the use of a quality metric algorithm based on structural similarity index. This is suggested as one of the steps for making biometric applications environmentally aware, which is especially helpful when user/admin intervention is discouraged and towards testing self sufficient biometric systems.

Additional classifiers and feature extraction algorithms can be evaluated by following a similar methodology and set of research objectives. These can ultimately be compared with the feature-classifier combinations used in this thesis.

The guidelines can be structured in a practical way that allows for automatic application (adaptation) to biometric systems in the field. Additional guidelines can also be recorded by following the proposed methodology in future related systems.

References

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

Ahmad, M. I., Woo, W. L., and Dlay, S. S. Multimodal biometric fusion at feature level: Face and palmprint. In *Proceedings of the 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, pages 801–805. IEEE, 2010.

Ahonen, T., Hadid, A., and Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

Anitha, M. and Rao, K. R. Extraction of region of interest (ROI) for palm print and inner knuckle print. *International Journal of Computer Applications*, 124(14), 2015.

Badrinath, G. and Gupta, P. Stockwell transform based palm-print recognition. *Applied Soft Computing*, 11(7):4267–4281, 2011.

Bausinger, O. and Tabassi, E. Fingerprint sample quality metric NFIQ 2.0. In *Proceedings of the International Conference on Biometrics Special Interest Group (BIOSIG)*, pages 167–171. 2011.

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

Bharadwaj, S., Vatsa, M., and Singh, R. Biometric quality: from assessment to multibiometrics. Technical report, Department of Computer Science and Engineering, IIIT-Delhi, 2015.

Boudaoud, L. B., Solaiman, B., and Tari, A. A modified ZS thinning algorithm by a hybrid approach. *The Visual Computer*, pages 1–18, 2017.

Bovik, A. C. Handbook of Image and Video Processing. Academic Press, 2010.

Brown, D. and Bradshaw, K. An investigation of face and fingerprint feature-fusion guidelines. In **Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., and Kostrzewa, D.**, editors, *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery: 12th International Conference, Ustroń, Poland, May 31 - June 3, 2016, Proceedings*, pages 585–599. Springer International Publishing, 2016a. doi:10.1007/978-3-319-34099-9_45.

Brown, D. and Bradshaw, K. A multi-biometric feature-fusion framework for improved uni-modal and multi-modal human identification. In *Technologies for Homeland Security (HST)*. IEEE, 2016b.

Brown, D. and Bradshaw, K. Feature-fusion guidelines for image-based multi-modal biometric fusion. *South African Computer Journal*, 29(1):92–121, 2017a.

Brown, D. and Bradshaw, K. Improved automatic face segmentation and recognition for applications with limited training data. In **Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., and Kostrzewa, D.**, editors, *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation: 13th International Conference, Ustroń, Poland, May 30 - June 2, 2017, Proceedings*, pages 415–426. Springer International Publishing, 2017b. doi:10.1007/978-3-319-58274-0_33.

Buades, A., Coll, B., and Morel, J.-M. A non-local algorithm for image denoising. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65. IEEE, 2005.

Budhi, G. S., Adipranata, R., and Hartono, F. J. The use of Gabor filter and back-propagation neural network for the automobile types recognition. In *Proceedings of the 2nd International Conference on Standardization and Innovation in Information Technology (SIIT)*. 2010.

Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

Chapelle, O., Haffner, P., and Vapnik, V. N. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, 1999.

Chen, G., Bui, T. D., and Krzyzak, A. Palmprint classification using dual-tree complex wavelets. In *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP)*, pages 2645–2648. IEEE, 2006a.

Chen, X., Tian, J., and Yang, X. A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure. *IEEE Transactions on Image Processing*, 15(3):767–776, 2006b.

Chikkerur, S., Cartwright, A. N., and Govindaraju, V. Fingerprint enhancement using STFT analysis. *Pattern Recognition*, 40(1):198–211, 2007.

Choge, H. K., Oyama, T., Karungaru, S., Tsuge, S., and Fukumi, M. A circle-based region-of-interest segmentation method for palmprint recognition. In *Proceedings of the International Conference on Control, Automation, and Systems and Society of Instrument and Control Engineers Joint Conference (ICCAS-SICE)*, pages 4993–4997. IEEE, 2009.

Crammer, K. and Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(Dec):265–292, 2001.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

Daugman, J. Iris recognition border-crossing system in the UAE. *International Airport Review*, 8(2), 2004.

Ding, B. and Ruan, Q. The localization of the palmprint images based on the maximal effective circle. In *Proceedings of the 8th International Conference on Signal Processing*, volume 4. IEEE, 2006.

Eskandari, M. and Toygar, Ö. Selection of optimized features and weights on face-iris fusion using distance images. *Computer Vision and Image Understanding*, 137:63–75, 2015.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

Feng, J. and Jain, A. Fingerprint reconstruction: From minutiae to phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):209–223, 2011. doi: 10.1109/TPAMI.2010.77.

Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

Guo, Z. and Hall, R. W. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989.

Guo, Z., Zhang, L., and Zhang, D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition*, 43(3):706–719, 2010.

Haghighat, M., Abdel-Mottaleb, M., and Alhalabi, W. Fully automatic face normalization and single sample face recognition in unconstrained environments. *Expert Systems with Applications*, 47:23–34, 2016.

Hsu, C.-W. and Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

Iloanusi, O. N. Fusion of finger types for fingerprint indexing using minutiae quadruplets. *Pattern Recognition Letters*, 38(0):8 – 14, 2014. doi:http://dx.doi.org/10.1016/j.patrec.2013.10.019.

Jain, A., Hong, L., and Pankanti, S. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000a.

Jain, A., Ross, A., and Prabhakar, S. Fingerprint matching using minutiae and texture features. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 282–285. IEEE, 2001.

Jain, A., Ross, A. A., and Nandakumar, K. Introduction to Biometrics. Springer Science & Business Media, 2011.

Jain, A. K. and Li, S. Z. Handbook of Face Recognition. Springer, 2011.

Jain, A. K., Prabhakar, S., Hong, L., and Pankanti, S. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000b.

Jillela, R. and Ross, A. A. Methods for iris segmentation. In **Bowyer, K. W. and Burge, M. J.**, editors, *Handbook of Iris Recognition*, pages 137–184. Springer London, 2016. doi:10.1007/978-1-4471-6784-6_7.

Kafai, M., Eshghi, K., An, L., and Bhanu, B. Reference-based pose-robust face recognition. In *Advances in Face Detection and Facial Image Analysis*, pages 249–278. Springer, 2016.

Karki, M. V. and Selvi, S. S. Multimodal biometrics at feature level fusion using texture features. *International Journal of Biometrics and Bioinformatics*, 7(1):58–73, 2013.

Kasinski, A., Florek, A., and Schmidt, A. The PUT face database. *Image Processing and Communications*, 13(3-4):59–64, 2008.

Kazemi, V. and Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874. 2014.

Khan, Z., Shafait, F., Hu, Y., and Mian, A. Multispectral palmprint encoding and recognition. *Clinical Orthopaedics and Related Research (CoRR)*, abs/1402.2941, 2014.

Khanyile, N. P., Tapamo, J.-R., and Dube, E. A comparative study of fingerprint thinning algorithms. In *Proceedings of the 11th Annual Information Security for South Africa Conference*, pages 15–17. IEEE, 2011.

Khellat-Kihel, S., Abrishambaf, R., Monteiro, J. L., and Benyettou, M. Multimodal fusion of the finger vein, fingerprint and the finger-knuckle-print using kernel fisher analysis. *Applied Soft Computing*, 42:439–447, 2016.

Kindt, E. J. An introduction into the use of biometric technology. In *Privacy and Data Protection Issues of Biometric Applications: A Comparative Legal Analysis*, pages 15–85. Springer Netherlands, 2013. doi:10.1007/978-94-007-7522-0_2.

King, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

King, D. E. Max-margin object detection. *Clinical Orthopaedics and Related Research (CoRR)*, abs/1502.00046, 2015.

Kocharyan, D. A modified fingerprint image thinning algorithm. *American Journal of Software Engineering and Applications*, 2(1):1–6, 2013.

Kouzani, A., Nahavandi, S., Kouzani, N., Kong, L., and She, F. A morphing technique for facial image representation. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1378–1383. IEEE, 2000.

Kumar, A. Incorporating cohort information for reliable palmprint authentication. In *Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, pages 583–590. IEEE, 2008.

Kumar, A. and Passi, A. Comparison and combination of iris matchers for reliable personal authentication. *Pattern Recognition*, 43(3):1016–1026, 2010.

Li, P., Zhang, T. *et al.* Proposal of novel histogram features for face detection. In *Proceedings of the International Conference on Pattern Recognition and Image Analysis*, pages 334–343. Springer, 2005.

Li, W., Zhang, B., Zhang, L., and Yan, J. Principal line-based alignment refinement for palmprint recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1491–1499, 2012.

Luo, X., Tian, J., and Wu, Y. A minutiae matching algorithm in fingerprint verification. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 833–836. IEEE, 2000.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., and Jain, A. K. Fvc2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

Malek, O., Venetsanopoulos, A., Androutsos, D., and Zhao, L. Subspace state estimator for facial biometric verification. In *Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI)*, volume 1, pages 137–143. IEEE, 2014.

Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. Handbook of Fingerprint Recognition. Springer Science & Business Media, 2009.

Mansfield, T., Kelly, G., Chandler, D., and Kane, J. Biometric product testing final report. Technical report, Technical Report X92A/4009309, Computing, National Physical Laboratory, 2001.

Methani, C. and Namboodiri, A. M. Pose invariant palmprint recognition. In **Tistarelli, M. and Nixon, M. S.**, editors, *Advances in Biometrics: Third International Conference, Alghero, Italy, June 2-5, 2009. Proceedings*, pages 577–586. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-01793-3_59.

Morales, A., Ferrer, M. A., and Kumar, A. Towards contactless palmprint authentication. *IET Computer Vision*, 5(6):407–416, 2011.

Ojala, T., Pietikäinen, M., and Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

Ojala, T., Pietikainen, M., and Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.-J., Vivaracho, C. *et al.* MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings - Vision, Image and Signal Processing*, 150(6):395–401, 2003.

Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

Paul, P. P. and Gavrilova, M. Multimodal biometrics using cancelable feature fusion. In *Proceedings of the International Conference on Cyberworlds (CW)*, pages 279–284. IEEE, 2014.

Paul, P. P., Gavrilova, M. L., and Alhajj, R. Decision fusion for multimodal biometrics using social network analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11):1522–1533, 2014. doi:10.1109/TSMC.2014.2331920.

Peralta, D., Triguero, I., Sanchez-Reillo, R., Herrera, F., and Benitez, J. Fast fingerprint identification for large databases. *Pattern Recognition*, 47(2):588–602, 2014. doi:10.1016/j.patcog.2013.08.002.

Phillips, P. J., Martin, A., Wilson, C. L., and Przybocki, M. An introduction evaluating biometric systems. *Computer*, 33(2):56–63, 2000.

Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. J. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

Podder, P., Khan, T. Z., Khan, M. H., Rahman, M. M., Ahmed, R., and Rahman, M. S. An efficient iris segmentation model based on eyelids and eyelashes detection in iris recognition system. In *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–7. IEEE, 2015.

Podio, F. JTC 1/SC 37. https://www.iso.org/isoiec_jtc1sc37.html, 2017. Accessed: 2017-01-30.

Porwik, P. and Wrobel, K. The new algorithm of fingerprint reference point location based on identification masks. In **Kurzyński, M., Puchała, E., Woźniak, M., and żołnierek, A.**, editors, *Computer Recognition Systems: 4th International Conference on Computer Recognition Systems*, pages 807–814. Springer Berlin Heidelberg, 2005. doi:10.1007/3-540-32390-2_95.

Quinn, G. W., Grother, P., and Matey, J. IREX IX. Technical report, NIST Interagency/Internal Report (NISTIR)-7836, Image Group, Information Access Division, Information Technology Laboratory, 2016.

Raghavendra, R., Dorizzi, B., Rao, A., and Kumar, G. H. Designing efficient fusion schemes for multimodal biometric systems using face and palmprint. *Pattern Recognition*, 44(5):1076–1088, 2011.

Rattani, A., Kisku, D. R., Bicego, M., and Tistarelli, M. Feature level fusion of face and fingerprint biometrics. In *Proceedings of the Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–5. 2011.

Ribaric, S. and Fratric, I. Experimental evaluation of matching-score normalization

techniques on different multimodal biometric systems. In *Proceedings of the IEEE Mediterranean Electrotechnical Conference (MELECON)*, pages 498–501. IEEE, 2006.

Ross, A. A., Nandakumar, K., and Jain, A. Handbook of Multibiometrics, volume 6. Springer Science & Business Media, 2006.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18, 2016.

Sagonas, C., Panagakis, Y., Zafeiriou, S., and Pantic, M. Face frontalization for alignment and recognition. *Clinical Orthopaedics and Related Research (CoRR)*, abs/1502.00852, 2015.

Samaria, F. S. and Harter, A. C. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994.

Samet, R., Shokouh, G. S., and Baskurt, K. B. An efficient pose tolerant face recognition approach. In **Gavrilova, M. L., Tan, C. K., Iglesias, A., Shinya, M., Galvez, A., and Sourin, A.**, editors, *Transactions on Computational Science XXVI: Special Issue on Cyberworlds and Cybersecurity*, pages 161–172. Springer Berlin Heidelberg, 2016. doi:10.1007/978-3-662-49247-5_10.

Sepasian, M., Balachandran, W., and Mares, C. Image enhancement for fingerprint minutiae-based algorithms using CLAHE, standard deviation analysis and sliding neighborhood. In *Proceedings of the World Congress on Engineering and Computer Science*, pages 22–24. 2008.

Shams, M. Y., Tolba, A. S., and Sarhan, S. H. Face, iris, and fingerprint multimodal identification system based on local binary pattern with variance histogram and combined learning vector quantization. *Journal of Theoretical and Applied Information Technology*, 89(1):53, 2016.

Sharma, P. and Kaur, M. Multimodal classification using feature level fusion and SVM. *International Journal of Computer Applications*, 76(4):26–32, 2013.

Simon-Zorita, D., Ortega-Garcia, J., Sanchez-Asenjo, M., and Gonzalez-Rodriguez, J. Facing position variability in minutiae-based fingerprint verification through multiple references and score normalization techniques. In **Josef Kittler, M. N.**, editor, *Proceedings of the International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 1060–1060. 2003.

Subha, V. and Mariammal, G. Comparative analysis of palmprint matching techniques for person identification. *International Journal of Advanced Research in Computer Science*, 8(3), 2017.

Sun, Z., Tan, T., Wang, Y., and Li, S. Z. Ordinal palmprint represention for personal identification [represention read representation]. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 279–284. IEEE, 2005.

Suzuki, S. and Keiichi, A. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

Tabassi, E., Watson, C., Fiumara, G., Salamon, W., Flanagan, P., and Cheng, S. L. Performance evaluation of fingerprint open-set identification algorithms. In *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2014.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. DeepFace: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708. 2014.

Tan, T. CASIA Iris Image Database. http://biometrics.idealtest.org/, 2016. [Online; accessed 14-March-2016].

Tang, Y. Deep learning using support vector machines. *Clinical Orthopaedics and Related Research (CoRR)*, 2, 2013.

Thepade, S. D., Bhondave, R. K., and Mishra, A. Comparing score level and feature level fusion in multimodal biometric identification using iris and palmprint traits with fractional transformed energy content. In *Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 306–311. IEEE, 2015.

Thomaz, C. E. and Giraldi, G. A. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010.

Tran, L. B. and Le, T. H. Personal authentication using relevance vector machine (rvm) for biometric match score fusion. In *In Proceedings of the Seventh International Conference on Knowledge and Systems Engineering (KSE)*, pages 7–12. IEEE, 2015.

Umer, S., Dhara, B. C., and Chanda, B. A fast and robust method for iris localization. In *Proceedings of the Fourth International Conference on Emerging Applications of Information Technology (EAIT)*, pages 262–267. IEEE, 2014.

Umer, S., Dhara, B. C., and Chanda, B. Iris recognition using multiscale morphologic features. *Pattern Recognition Letters*, 65:67–74, 2015.

Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518. IEEE, 2001.

Vishi, K. and Yayilgan, S. Y. Multimodal biometric authentication using fingerprint and iris recognition in identity management. In *Proceedings of the Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 334–341. IEEE, 2013.

Wang, Y., Ruan, Q., and Pan, X. An improved square-based palmprint segmentation

method. In *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 316–319. IEEE, 2007.

Wang, Z., Liu, C., Shi, T., and Ding, Q. Face-palm identification system on feature level fusion based on CCA. *Journal of Information Hiding and Multimedia Signal Processing*, 4(4):272–279, 2013.

Wayman, J. Biometric verification/identification/authentication/recognition: The terminology. In **Li, S. and Jain, A.**, editors, *Encyclopedia of Biometrics*, pages 263–268. Springer US, 2015.

Wayman, J., Jain, A., Maltoni, D., and Maio, D. An introduction to biometric authentication systems. In *Biometric Systems*, pages 1–20. Springer, 2005.

Wildes, R. P. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

Wu, X., Wang, K., and Zhang, D. Line feature extraction and matching in palmprint. In *Proceedings of the Second International Conference on Image and Graphics*, volume 4875, pages 583–591. International Society for Optics and Photonics, 2002.

Xu, H., Veldhuis, R. N., Kevenaar, T. A., and Akkermans, T. A. A fast minutiae-based fingerprint recognition system. *IEEE Systems journal*, 3(4):418–427, 2009.

Yang, J., yu Yang, J., Zhang, D., and feng Lu, J. Feature fusion: parallel strategy vs. serial strategy. *Pattern Recognition*, 36(6):1369 – 1381, 2003.

Yao, Y.-F., Jing, X.-Y., and Wong, H.-S. Face and palmprint feature level fusion for single sample biometrics recognition. *Neurocomputing*, 70(7):1582–1586, 2007.

Yi, D., Lei, Z., and Li, S. Z. Towards pose robust face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3545. 2013.

Yin, Y., Liu, L., and Sun, X. SDUMLA-HMT: A multimodal biometric database. In **Sun, Z., Lai, J., Chen, X., and Tan, T.**, editors, *Biometric Recognition: 6th Chinese Conference, Beijing, China, December 3-4, 2011. Proceedings*, pages 260–268. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-25449-9_33.

Zhang, D. Palmprint segmentation by key point features. In *Palmprint Authentication*, pages 73–83. Springer US, 2004. doi:10.1007/1-4020-8097-2_6.

Zhang, D., Guo, Z., and Gong, Y. An online system of multispectral palmprint verification. In *Multispectral Biometrics*, pages 117–137. Springer, 2016.

Zhang, D., Guo, Z., Lu, G., Zhang, L., and Zuo, W. An online system of multispectral palmprint verification. *IEEE transactions on instrumentation and measurement*, 59(2):480–490, 2010.

Zhang, D., Kong, W.-K., You, J., and Wong, M. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1041–1050, 2003.

Zhang, T. and Suen, C. Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.

Zhenan, S., Tieniu, T., Wang, Y., and Stan, Z. L. CASIA-Palmprint image database. http://biometrics.idealtest.org, 2016. Accessed: 2016-01-30.

Zhou, Y. and Kumar, A. Personal identification from iris images using localized Radon transform. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR),*, pages 2840–2843. IEEE, 2010.

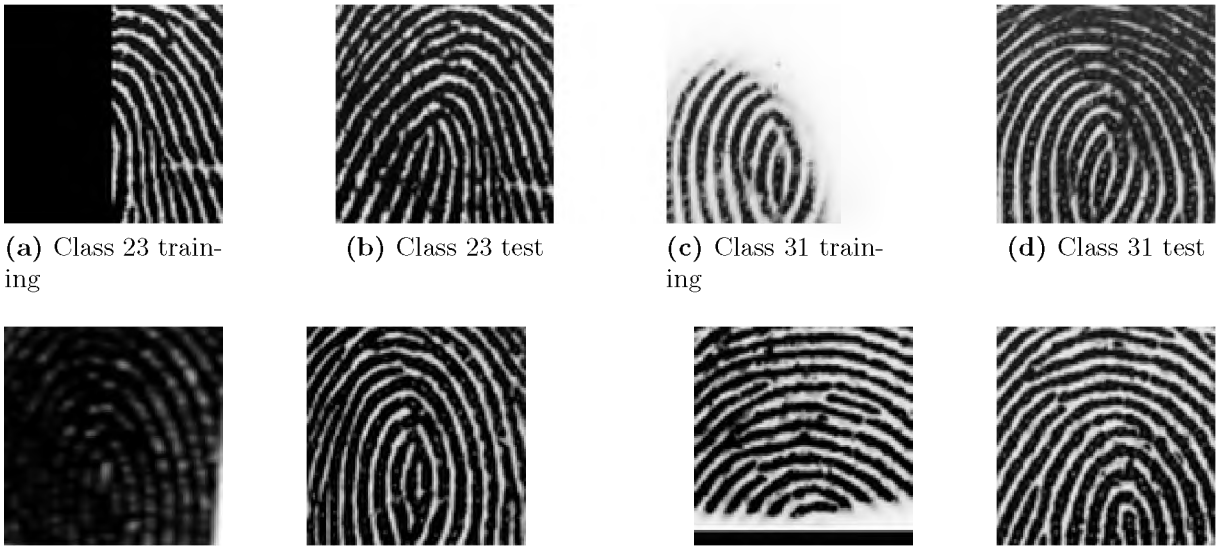
Zou, J., Feng, J., Zhang, X., and Ding, M. Local orientation field based nonlocal means method for fingerprint image de-noising. *Journal of Signal and Information Processing*, 4:150–153, 2013.

Zuobin, W., Kezhi, M., and Ng, G.-W. Effective feature fusion for pattern classification based on intra-class and extra-class discriminative correlation analysis. In

Proceedings of the 20th International Conference on Information Fusion, pages 1–8. IEEE, 2017.

Appendix A

Ranked Samples



(a) Class 23 training **(b)** Class 23 test **(c)** Class 31 training **(d)** Class 31 test

(e) Class 47 training **(f)** Class 47 test **(g)** Class 49 training **(h)** Class 49 test

Figure A.1: Ranked fingerprint samples with bad training images.

(a) Class 21 training **(b)** Class 21 test **(c)** Class 44 training **(d)** Class 44 test

Figure A.2: Ranked iris samples.

(a) Class 50 training

(b) Class 50 test

(c) Class 50 test

Figure A.3: Correctly identified face samples.

(a) Class 38 training

(b) Class 38 test

(c) Class 38 test

(d) Class 39 training

(e) Class 39 test

Figure A.4: Misidentified face samples.

(a) Class 31 good training

(b) Class 31 good test

(c) Class 20 bad training

(d) Class 20 bad test

Figure A.5: Ranked palmprint samples

Appendix B

Code Implementation

The code provided in this appendix uses OpenCV and Dlib libraries which are under the terms of the GNU General Public License. Sections B.1, B.2, B.3 and B.4, provide code specific to respective biometric modalities. Section B.5 contains the main code that carries out preprocessing, modality specific processing, postprocessing and classification, as described in the overview of Chapter 4.

Eigen, Fisher and LBPH models are built using OpenCV's Facerecognizer class, while SVM models are built by piping pixel data to an external file, which is used by Liblinear. Code largely similar to respective library examples is omitted (space limitations) but mentioned. Furthermore, bash and Python scripts used to organize datasets for training and testing of all combinations are also omitted from this appendix. The full version of this code is available at: https://github.com/browndane/fuseFF.

B.1 Fingerprint Specific Code

The function determineCore() is based on code snippets in Biometric Methods course at University of Wrocaw and is available at: https://github.com/rtshadow/biometrics.

Listing B.1: Fingerprint Algorithm (fingerprint.cpp).

```
1
2  void cropFinger( Mat& img)
3  {
4      normalize(img, img, 0.0, 255.0, NORM_MINMAX, CV_8U);
5
6      float roiSize = preseg(img);
7
8      determineCore(img);
9
10     Mat shift = shiftImg (img, img.cols/2 - core.x, img.rows/2 - core.y);
11
12     Mat temp1, temp2, temp3, temp4, temp5, temp6;
13
14     Mat rotat = rotImg (shift , core, 180/CV_PI * angleBetween (core, minPoint), 1);
15
16     //crop according to core position
17
18     Rect roi = Rect(shift.cols/2 - roiSize*0.75, shift.rows/2 - roiSize*0.75, roiSize,
        roiSize);
19     Rect roi2 = Rect(shift.cols/2 - roiSize*0.5, shift.rows/2 - roiSize*0.5, roiSize/1.5,
        roiSize/1.5);
20     Rect roi3 = Rect(shift.cols/2 - roiSize*0.25, shift.rows/2 - roiSize*0.25, roiSize
        *0.75, roiSize*0.75);
21
22     //rotate by core and minutiae using affine transformation
23     //roi = 75%
24     //roi2 = 50%
25     //roi3 = 25%
26     rotat (roi);
27     rotat (roi2);
28     rotat (roi3);
29
30     //resize to 50th pixel
31     resize(rotat (roi),temp4,Size(roiSize,roiSize),0,0,INTER_LANCZOS4);
32     resize(rotat (roi2),temp5,Size(roiSize/1.5,roiSize/1.5),0,0,INTER_LANCZOS4);
33     resize(rotat (roi3),temp6,Size(roiSize*0.75,roiSize*0.75),0,0,INTER_LANCZOS4);
34 }
```

B.2 Face Specific Code

The frontalization code for face segmentation was adopted from the examples by author Satya Mallick[1]. VRA uses the frontalization code as a base.

Listing B.2: Face landmarks (Dlib).

```
1
2 Two functions detectL and applyTrainL(), based on facelandmark.cpp in dlib examples,
    calculate and generate the face mesh, respectively.
```

Listing B.3: Calculate Delaunay triangles.

```
1  static void calculateDelaunayTriangles(
2      Rect rect, vector<Point2f>& points, vector<vector<int> >& delaunayTri)
3  {
4
5      // Create Subdiv2D to store all triangles as a set of pixel blocks
6      Subdiv2D subdiv(rect);
7
8      // Insert points into subdiv
9      for (vector<Point2f>::iterator it = points.begin(); it != points.end(); it++)
10         subdiv.insert(*it);
11
12     vector<Vec6f> triangleList;
13     subdiv.getTriangleList(triangleList);
14     vector<Point2f> pt(3);
15     vector<int> ind(3);
16
17     for (size_t i = 0; i < triangleList.size(); i++)
18     {
19         Vec6f t = triangleList[i];
20         pt[0] = Point2f(t[0], t[1]);
21         pt[1] = Point2f(t[2], t[3]);
22         pt[2] = Point2f(t[4], t[5]);
23
24         if (rect.contains(pt[0]) && rect.contains(pt[1]) && rect.contains(pt[2]))
25         {
26             for (int j = 0; j < 3; j++)
27                 for (size_t k = 0; k < points.size(); k++)
28                     if (abs(pt[j].x - points[k].x) < 1.0 && abs(pt[j].y - points[k].y) <
    1)
29                         ind[j] = k;
30
```

[1] https://www.learnopencv.com

```
            delaunayTri.push_back(ind);
        }
    }
}
```

Listing B.4: Calculate pose angle.

```

float calcPose(Mat& img)
{
    // 6 landmarks on ideal frontal face
    std::vector<cv::Point2d> image_points;
    image_points.push_back(trainLandmarks[currentIdx][37]); // Left eye left corner
    image_points.push_back(trainLandmarks[currentIdx][46]); // Right eye right corner
    image_points.push_back(trainLandmarks[currentIdx][36]); // Nose tip
    image_points.push_back(trainLandmarks[currentIdx][36]); // Chin
    image_points.push_back(trainLandmarks[currentIdx][49]); // Left Mouth corner
    image_points.push_back(trainLandmarks[currentIdx][55]); // Right mouth corner

    // Camera internals
    double focal_length = im.cols; // Approximate focal length.
    Point2d center = cv::Point2d(im.cols / 2, im.rows / 2);
    cv::Mat camera_matrix
        = (cv::Mat_<double>(3, 3) << focal_length, 0, center.x, 0, focal_length, center.y
    , 0, 0, 1);
    cv::Mat dist_coeffs
        = cv::Mat::zeros(4, 1, cv::DataType<double>::type); // Assuming no lens
    distortion in dataset images

    cout << "Camera Matrix " << endl << camera_matrix << endl;
    // Output rotation and translation
    cv::Mat rotation_vector; // Rotation in axis-angle form
    cv::Mat translation_vector;

    // Solve for pose using perspective-n-point
    cv::solvePnP(model_points, image_points, camera_matrix, dist_coeffs, rotation_vector,
        translation_vector);

    // Project a 3D point (0, 0, 1000.0) onto the image plane.
    // We use this to draw a line sticking out of the nose

    vector<Point3d> nose_end_point3D;
    vector<Point2d> nose_end_point2D;
    nose_end_point3D.push_back(Point3d(0, 0, 1000.0));

    projectPoints(nose_end_point3D, rotation_vector, translation_vector, camera_matrix,
```

```
    dist_coeffs,
38          nose_end_point2D);
39
40      for (int i = 0; i < image_points.size(); i++)
41      {
42          circle(im, image_points[i], 3, Scalar(0, 0, 255), -1);
43      }
44
45      //cv::line(im, image_points[0], nose_end_point2D[0], cv::Scalar(255, 0, 0), 2);
46
47     //return the estimated pose angle
48
49      return angleBetween(image_points[0], nose_end_point2D[0]);
50  }
```

The following VRA implementation uses frontalization code as a base.

Listing B.5: Face segmentation (VRA) (face.cpp).

```
1   void faceSeg(float modelPoseAngle)
2   {
3       // get pose
4       float poseAngle = calcPose(faceImages);
5
6       detectL(faceImages, 1); // dlib face landmark detection
7
8       applyTrainL(); // generate Delaunay mesh
9
10      int numImages = 2; //trained model image against all test images
11
12      // Ideal eye positions as reference for initial frontalization
13      vector<Point2f> eyecornerDst;
14      eyecornerDst.push_back(Point2f(0.35 * w, h / 3));
15      eyecornerDst.push_back(Point2f(0.65 * w, h / 3));
16
17      vector<Mat> imagesNorm, betweenFaces;
18      vector<vector<Point2f> > pointsNorm;
19
20      // 8 Boundary points for Delaunay Triangulation and ideal landmarks for
        frontalization
21      vector<Point2f> boundaryPts;
22      boundaryPts.push_back(Point2f(0, 0));
23      boundaryPts.push_back(Point2f(w / 2, 0));
24      boundaryPts.push_back(Point2f(w - 1, 0));
25      boundaryPts.push_back(Point2f(w - 1, h / 2));
26      boundaryPts.push_back(Point2f(w - 1, h - 1));
```

```
27      boundaryPts.push_back(Point2f(w / 2, h - 1));
28      boundaryPts.push_back(Point2f(0, h - 1));
29      boundaryPts.push_back(Point2f(0, h / 2));
30
31      // find average of transformed landmarks for VRA and
32      // warp images and transformed landmarks to output coordinate system
33
34      int cntEven = 0;
35      int cntEven2 = 0;
36
37      for (size_t idx = 0; idx < faceImages.size() - 1;
38           idx++) // train(first img) & iterated test image with same angle
39      {
40          // Space for average landmark points
41          vector<Point2f> pointsAvg(trainLandmarks[0].size());
42          int globalIdx = 0;
43
44          for (size_t i1 = idx; i1 < idx + 2; i1++)
45          {
46              int currentIdx = i1;
47
48              if (cntEven % 2 == 0)
49                  currentIdx = 0;
50              else
51                  currentIdx = i1;
52
53              vector<Point2f> eyecornerSrc, eyecornerRef;
54
55              vector<Point2f> points = trainLandmarks[currentIdx];
56
57              eyecornerSrc.push_back(trainLandmarks[currentIdx][37]);
58              eyecornerSrc.push_back(trainLandmarks[currentIdx][46]);
59
60              //          Calculate similarity transform
61              Mat tform;
62              similarityTransform(eyecornerSrc, eyecornerDst, tform);
63
64              //          Apply similarity transform to input image and landmarks
65              Mat img = Mat::zeros(h, w, CV_32FC1);
66              warpAffine(trainCropFaceImages[currentIdx], img, tform, img.size());
67
68              transform(points, points, tform); // update points pos based on tran mat
69
70              //          Calculate average landmark locations
71
72              for (size_t j = 0; j < points.size(); j++) // ref sample problem here
```

```
73                pointsAvg[j] += points[j] * (1.0 / numImages);//warping between angle
74
75            // Append boundary points. Will be used in Delaunay Triangulation
76            for (size_t j = 0; j < boundaryPts.size(); j++)
77                points.push_back(boundaryPts[j]);
78
79            pointsNorm.push_back(points);
80            imagesNorm.push_back(img); // straightened face according to
    similarityTransform (eyes)
81
82            cntEven++;
83        }
84
85        // Append boundary points to average points.
86        for (size_t j = 0; j < boundaryPts.size(); j++)
87        {
88            pointsAvg.push_back(boundaryPts[j]);
89        }
90
91        // get Delaunay triangles
92        Rect rect(0, 0, w, h);
93        vector<vector<int> > dt;
94        calculateDelaunayTriangles(rect, pointsAvg, dt);
95
96        // Space for output image
97        Mat output = Mat::zeros(h, w, CV_32FC1);
98        Size size(w, h);
99
100       // check for VRA method
101       angleDifference = poseAngle - modelPoseAngle;
102       if (angleDifference < 0)
103       {
104          mirror(img);
105          transform(poseAngle, modelPoseAngle); //update angles
106       }
107
108
109       if (poseAngle - modelPoseAngle > 30)
110       {
111          if(identification = 1)
112          {
113             getFixedModelPose(); //30,45,60,75 degrees
114             goto finishIden;
115          }
116          else
117             getModelPose(); // all available (verification)
```

```
118             angleDifference(poseAngle, modelPoseAngle);
119
120             // Warp input images to average image landmarks
121         if(poseAngle<60 || modelPoseAngle<60) //warp both input and database to
    inbetween angle
122             for (size_t i2 = idx; i2 < idx + 2; i2++)
123             {
124
125                 int currentIdx = i2;
126
127                 if (cntEven2 % 2 == 0)
128                     currentIdx = 0;
129                 else
130                     currentIdx = i2;
131
132                 Mat img = Mat::zeros(h, w, CV_32FC1);
133                 // Transform triangles one by one
134                 for (size_t j = 0; j < dt.size(); j++)
135                 {
136                     // Input and output points corresponding to jth triangle
137                     vector<Point2f> tin, tout;
138                     for (int k = 0; k < 3; k++)
139                     {
140                         Point2f pIn
141                             = pointsNorm[cntEven2][dt[j][k]]; // cntEven2 instead of
    currentIdx
142                         constrainPoint(pIn, size); // caters for overlapping pixels of
    triangles
143
144                         Point2f pOut = pointsAvg[dt[j][k]];
145                         constrainPoint(pOut, size); // caters for overlapping pixels of
    triangles
146
147                         tin.push_back(pIn);
148                         tout.push_back(pOut);
149                     }
150
151                     // Warp both input and database images and transform landmarks to
    output coordinate system,
152
153                     warpTriangle(
154                         imagesNorm[cntEven2], img, tin, tout); // input and database
    images
155                    //warp only input for iden
156                      finishIden:
157                       warpTriangle(
```

```
158                     imagesNorm[cntEven2], iImg, tin, tout); // input images only
159                     if(identification = 1)
160                     {
161                        img.clear();    //remove database images
162                       img (iImg);
163                     }
164                 }
165             }
166
167             // Add pixel block intensities for blending warped triangle
168             output = output + img;
169
170             img.convertTo(img, CV_8UC1, 255);
171
172             betweenFaces.push_back(img);
173             betweenPaths.push_back(paths[currentIdx]); // new paths
174
175             cntEven2++;
176         }
177
178         else
179         {
180             transform(points, boundaryPts,
181                 tform); // warp to reference angle for < 30 degrees (or frontalization)
182             warpAffine(trainCropFaceImages[currentIdx], img, tform, img.size());
183         }
184         // Divide by numImages to get average
185         output = output / (double)numImages;
186     }
187
188     // store VRA face
189
190
191     vector<Mat> betweenFacesFinal;
192     for (size_t i = 0; i < betweenFaces.size(); i++)
193     {
194         Mat finalImg = Mat::zeros(betweenFaces[i].rows, betweenFaces[i].cols, CV_8UC1);
195
196         if (i % 2 == 0)
197         {
198             finalMask = Mat::zeros(betweenFaces[i + 1].rows, betweenFaces[i + 1].cols,
    CV_8UC1);
199             for (int y = 0; y < betweenFaces[i + 1].cols; y++)
200             {
201                 for (int x = 0; x < betweenFaces[i + 1].rows; x++)
202                 {
```

```
203                     if (betweenFaces[i + 1].at<uchar>(x, y) != 0)
204                     {
205                         finalMask.at<uchar>(x, y) = 255;
206                     }
207                 }
208             }
209             betweenFaces[i].copyTo(finalImg, finalMask); // fix overlap
210         }
211         else
212         {
213             betweenFaces[i].copyTo(finalImg);
214         }
215         betweenFacesFinal.push_back(finalImg);
216     }
```

B.3 Palmprint Specific Code

Listing B.6: palmprint.cpp

```
1
2  void findMIC()
3  {
4     //bounded by palm
5      findContours(thr, contours, hierarchy, CV_RETR_CCOMP,
6          CV_CHAIN_APPROX_SIMPLE); // Find the contours in the image
7      for (int i = 0; i < contours.size(); i++) // iterate through each contour.
8      {
9          double a = contourArea(contours[i], false); //  Find the area of contour
10         if (a > largest_area)
11         {
12             largest_area = a;
13             largest_contour_index = i; // Store the index of largest contour
14             bounding_rect = boundingRect(contours[i]); // Find the bounding rectangle for
       biggest contour
15         }
16     }
17 }
18 void palmSegmentation()
19 {
20     Mat skin;
21     if (isGrayImage(image) == 0) // use skin if color image
22         skin = GetSkin(eImg);
23     else
```

```
24      {
25          cvtColor(eImg, eImg, CV_BGR2GRAY);
26          threshold(eImg, skin, 60, 255, THRESH_BINARY | CV_THRESH_OTSU);
27          //          cvtColor(skin, skin, CV_GRAY2BGR);
28      }
29      // contours
30      if (isGrayImage(image) == 0)
31          cvtColor(skin, skin, CV_BGR2GRAY);
32
33      int erosion_size = 4;
34      Mat element = getStructuringElement(MORPH_CROSS,
35          Size(2 * erosion_size + 1, 2 * erosion_size + 1), Point(erosion_size,
        erosion_size));
36
37      findContours(skin, contours, hierarchy, CV_RETR_TREE, CV_CHAIN_APPROX_SIMPLE, Point
        (0, 0));
38      int s = findBiggestContour(contours); // remove inner contours
39
40      Mat drawing = Mat::zeros(eImg.size(), CV_8UC1);
41      drawContours(drawing, contours, s, Scalar(255, 0, 0), -1, 8, hierarchy, 0, Point());
        // get hand
42
43      drawing.copyTo(cleanHand);
44
45      contours.clear();
46      hierarchy.clear();
47
48      Mat cleanHand2;
49      cleanHand.copyTo(cleanHand2);
50
51      findContours(cleanHand2, contours, hierarchy, CV_RETR_TREE, CV_CHAIN_APPROX_NONE,
        Point(0, 0));
52
53      for (int i = 0; i < contours.size(); i++)
54      {
55          for (int j = 0; j < contours[i].size(); j++)
56          {
57              if (contours[i][j].y < contours[i][j + 1].y)
58                  cntUp++;
59
60              if (contours[i][j].y > contours[i][j + 1].y && foundFing != 1)
61                  cntDown++;
62
63              if (cntUp > 3)
64              {
65
```

```
66                  cntUp = 0;
67                  foundFing = 0;
68              }
69              // initial check for fingers
70              if (cntDown > thresDist && contours[i][j - thresDist].x > xBound
71                  && contours[i][j - thresDist].x < image.cols - xBound
72                  && contours[i][j - thresDist].y > yBound
73                  && contours[i][j - thresDist].y < image.rows / 1.5)
74
75                  cntDown = 0;
76              foundFing = 1;
77
78              sortContoursY.push_back(contours[i][j - thresDist]);
79          }
80      }
81      // sort by y then x of contour array
82      sort(sortContoursY.begin(), sortContoursY.end(), sortY);
83      sort(sortContoursY.begin(), sortContoursY.end(), sortX);
84
85      for (int j = 0; j < sortContoursY.size(); j++)
86      {
87
88          if ((abs(sortContoursY[j].y - sortContoursY[j + 1].y) < yThresh * 3
89                  && abs(sortContoursY[j].x - sortContoursY[j + 1].x) < xThresh / 2 && cntY
     == 0)
90              || (abs(sortContoursY[j].y - sortContoursY[j + 1].y) < yThresh * 1.5
91                      && abs(sortContoursY[j].x - sortContoursY[j + 1].x) < xThresh && cntY
    == 0))
92          {
93              cout << "1st point" << j << endl;
94
95              newRef = sortContoursY[j]; // update ref
96              pois.push_back(sortContoursY[j]);
97              cntY++;
98          }
99          else if ((abs(newRef.y - sortContoursY[j].y) < yThresh * 3
100                     && abs(newRef.x - sortContoursY[j].x) < xThresh / 2 && cntY == 1)
101             || (abs(newRef.y - sortContoursY[j].y) < yThresh * 1.5
102                     && abs(newRef.x - sortContoursY[j].x) < xThresh && cntY == 1))
103         {
104
105             cout << "2nd point" << j << endl;
106             if (sortContoursY[j].x > image.cols / 4
107                 && sortContoursY[j].x < image.cols - image.cols / 4)
108             {
109                 pois.push_back(sortContoursY[j]);
```

```
110                 newRef = sortContoursY[j];
111             }
112             else if (norm(newRef - sortContoursY[j]) < distThresh && cntY == 2)
113             {
114                 cout << "3rd point" << j << endl;
115 
116                 pois.push_back(sortContoursY[j]);
117                 cntY++;
118                 break;
119             }
120 
121             oldPois.clear();
122 
123             for (int j = 0; j < pois.size(); j++)
124                 oldPois.push_back(pois[j]);
125             rotImg(image, angleBetween(pois[0], pois[2]) - 90);
126 
127             transform(pois, pois, rot_mat); // update points pos based on tran mat
128 
129             if (pois[0].x < pois[2].x)
130             {
131                 cout << "pois[0] " << pois[0] << endl;
132                 cout << "pois[2] " << pois[2] << endl;
133 
134                 findMIC();
135                 // improved MEC, shift to 3 valley points
136 
137                 shiftImg(img, img.cols / 2 - pois[0].x, img.rows / 2 - pois[2].x);
138 
139                 Rect roi1 = Rect(
140                     pois[0].x, pois[0].y, norm(pois[0] - pois[2]), 1.5 * norm(pois[0] -
    pois[2]));
141             }
142             return image(roi1);
143         }
```

B.4 Iris Specific Code

Listing B.7: iris.cpp

```
1 
2 void detectPupil(Mat& img)
3 {
```

```
    normalize(img, img, 0.0, 255.0, NORM_MINMAX, CV_8U);
    equalizeHist(img, img);

    // LoG filter
    Mat LOGImg = LoG(img);

    int eleSize = 1;
    Mat element = getStructuringElement(
        MORPH_CROSS, Size(2 * eleSize + 1, 2 * eleSize + 1), Point(eleSize, eleSize));

    img.copyTo(LOGImgOrig);
    // do hough

    //   vector<Vec3f> circlesI;
    vector<Vec3f> circlesL;
    tuneHoughParameters(pupilSize);
    HoughCircles(
        LOGImg, circlesL, CV_HOUGH_GRADIENT, 1, 100, param1, param2, minRad, maxRad); //
    LOGImg

    // how close to centre of image (assumption)
    int tolerance = LOGImg.rows / 2;
    Point centerPupil = Point(0, 0);
    // Loop over all detected circles and outline them on the original image
    for (int i = 0; i < circlesL.size(); i++)
    {
        radiusPupil = round(circlesL[i][2]);

        // assume pupil is close to centre of image as inital search point
        if ((norm(Point(round(circlesL[i][0]), round(circlesL[i][1]))
                - Point(LOGImg.cols / 2, LOGImg.rows / 2))
               < tolerance)
            && radiusPupil > 15)
        {
            centerPupil = Point(round(circlesL[i][0]), round(circlesL[i][1]));

            circle(LOGImgOrig, centerPupil, radiusPupil, Scalar(255, 255, 255), 7);

            if (biggestRadius < radiusPupil)
                biggestRadius = radiusPupil;
            if (smallestRadius > radiusPupil)
                smallestRadius = radiusPupil;

            i = circlesL.size(); // done
        }
    }
```

```
49
50      // find start of sclera now that we have pupil centre (searching directions)
51      Point scleraStart;
52      Point scleraStartBak;
53
54      cleanSclera.copyTo(sclera);
55      // otsu method
56      threshold(img, sclera, scleraThresh, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
57
58      int erosion_size = 1;
59      Mat element2 = getStructuringElement(MORPH_CROSS,
60          Size(2 * erosion_size + 1, 2 * erosion_size + 1), Point(erosion_size,
    erosion_size));
61
62      for (int y = pixelJump; y < imageSizeY; y++) // usually not bigger than image size
63      {
64
65          int pixVal = sclera.at<uchar>(Point(centerPupil.x, centerPupil.y + y +
    radiusPupil));
66
67          if (pixVal != 0) // if white
68          {
69              scleraStart = Point(centerPupil.x, centerPupil.y + y + radiusPupil);
70
71              for (int i = 0; i < pixelJump; i++) // min size of iris is 20
72              {
73                  int pixVal2
74                      = sclera.at<uchar>(Point(centerPupil.x, centerPupil.y + y +
    radiusPupil + i));
75                  if (pixVal2 != 0) // if white
76                  {
77                      scleraStartBak = Point(centerPupil.x, centerPupil.y + y + radiusPupil
    );
78                      cnt++;
79                      circle(LOGImgOrig, scleraStart, 7, Scalar(255, 255, 255), CV_FILLED,
    8, 0);
80                  }
81              }
82              cout << "cnt " << cnt << endl;
83              cout << "y " << y << endl;
84
85              if (cnt > 5)
86              {
87                  break; // done
88              }
89              cnt = 0;
```

```
90          }
91
92          if (y > imageSizeY) // sclera not found
93          {
94              scleraStart = scleraStartBak;
95              break; // done
96          }
97      }
98
99      // find iris and check boundaries
100     if (irisDist + centerPupil.x > border) // out of bounds check
101         findBetterIris();
102     else
103         irisDist = border; // bound it at border as fallback option
104
105     circle(LOGImgOrig, centerPupil, irisDist, Scalar(255, 255, 255), 5);
106
107     // mask ROI for eyelash/occulsion removal
108
109     Mat maskEye = Mat::zeros(img.rows, img.cols, CV_8UC1);
110     circle(maskEye, centerPupil, irisDist, Scalar(255), CV_FILLED, 8, 0);
111
112     Mat maskPupil = Mat::zeros(img.rows, img.cols, CV_8UC1);
113     circle(maskPupil, centerPupil, oldRadiusPupil, Scalar(255), CV_FILLED, 8,
114         0); //+3 for imperfect pupil shape
115
116     Mat maskIris = Mat::zeros(img.rows, img.cols, CV_8UC1);
117     maskEye.copyTo(maskIris, 255 - maskPupil);
118
119     Mat irisROI = Mat::zeros(img.rows, img.cols, CV_8UC1);
120     img.copyTo(irisROI, maskIris);
121
122     // bottom half iris
123
124     Rect radiusRoi
125         = Rect(centerPupil.x - irisDist, centerPupil.y - irisDist, irisDist * 2, irisDist
     * 2);
126
127     Mat tempRoi;
128     (irisROI(radiusRoi)).copyTo(tempRoi);
129     Rect myROI = Rect(0, tempRoi.rows / 2, tempRoi.cols, tempRoi.rows / 2);
130 }
```

B.5 Unimodal and Multimodal Main Code

Listing B.8: Alignment operations (masterFusion.cpp)

```

Mat shiftImg(Mat &img, int offsetx, int offsety)
{
    Mat trans_mat = (Mat_<double>(2,3) << 1, 0, offsetx, 0, 1, offsety);
    warpAffine(img,img,trans_mat,img.size());
    return trans_mat;
}

Mat rotImg(Mat img, double angle)
{

    Point center = Point( img.cols/2, img.rows/2 );
    double scale = 1.0;

    // Get the rotation matrix with the specifications above
    rot_mat = getRotationMatrix2D( center, angle, scale );

    // Rotate the warped image
    warpAffine( img, img, rot_mat, img.size() );
    return img;
}

float angleBetween(const Point &v1, const Point &v2)
{

    float angle = (atan2(v1.y - v2.y, v1.x - v2.x) * 180 / CV_PI) - 90;
    return angle;
}
```

Listing B.9: Feature-classifer combinations (masterFusion.cpp)

```

//include necessary and declared variables here

static void optimalResize(Mat& reImage, Size finalSize)
{

    if (origSize.width >= finalSize.width && (matcher != "L"))
    {
        resizeType = 3;
        resize(reImage, reImage, finalSize, resizeType);
    }
```

```
12      if (origSize.width < finalSize.width && (matcher != "L"))
13      {
14          //      cout<< "increasing Size" << origSize.width<<endl;
15          resizeType = 4;
16          resize(reImage, reImage, finalSize, 1.5, 1.5, resizeType);
17      }
18      if (matcher == "L")
19      {
20          resizeType = 4;
21          resize(reImage, reImage, finalSize, 1.5, 1.5, resizeType);
22      }
23  }
24  // LoG filter
25
26  void LoG(temp)
27  {
28      if (matcher == "L")
29      {
30          gKernel = 7;
31          lKernel = 5;
32          sharpHigh = 0.5;
33      }
34      else
35      {
36          gKernel = 17;
37          lKernel = 7;
38      }
39
40      GaussianBlur(temp, temp, Size(gKernel, gKernel), 0, 0);
41      Laplacian(temp, input, CV_8U, lKernel, 1, 0, BORDER_DEFAULT);
42  }
43
44  static void read_csv(const string& filename, vector<Mat>& images, vector<int>& labels,
45      vector<string>& paths, char separator = ';')
46  {
47
48      ifstream file(filename.c_str(), ifstream::in);
49      if (!file)
50      {
51          string error_message = "No valid input file was given, please check the given
        filename.";
52          CV_Error(CV_StsBadArg, error_message);
53      }
54      string line, path, classlabel;
55      while (getline(file, line))
56      {
```

```
57          stringstream liness(line);
58          getline(liness, path, separator);
59          getline(liness, classlabel);
60          if (!path.empty() && !classlabel.empty())
61          {
62              Mat input = imread(path, 0);
63
64              Mat temp;
65
66              origSize = Size(input.cols, input.rows);
67
68              optimalResize(input, Size(finalSize, finalSize));
69
70              if (enhanceType == 1) // pixel normalization
71              {
72                  normalize(input, input, 0.0, 255.0, NORM_MINMAX, CV_8U);
73
74                  images.push_back(input);
75              }
76              else if (enhanceType == 2) // HE
77              {
78                  equalizeHist(input, input);
79
80                  images.push_back(input);
81              }
82
83              else if (enhanceType == 3 || enhanceType == 4)
84              {
85                  equalizeHist(input, input);
86
87                  LoG(input);
88
89                  images.push_back(temp);
90              }
91              // CLAHE
92              else if (enhanceType == 4)
93              {
94
95                  Ptr<CLAHE> clahe = createCLAHE();
96                  clahe->setClipLimit(3);
97                  clahe->apply(input, input);
98              }
99
100             else if (enhanceType == 5 || enhanceType == 6)
101                 ELBP and LLBP(V)
102                 {
```

```
103                     if (enhanceType == 6)
104                     {
105                         LoG(input);
106                     }
107                     else
108                         input.copyTo(temp);
109 
110                     Mat temp2;
111 
112                     VARLBP(temp, temp2, lbpRadiusSize,
113                         lbpNeighbourSize);
114 
115                     normalize(temp2, temp2, 0, 255, NORM_MINMAX, CV_8U);
116 
117                     optimalResize(temp2, Size(finalSize, finalSize));
118                     temp2 = (input - temp2) * 1.1; // orig - bad features
119 
120                     images.push_back(temp2);
121                 }
122             else if (enhanceType == 7) // LBPL(V)
123             {
124                 Mat temp2;
125 
126                 VARLBP(input, temp, lbpRadiusSize,
127 
128                 normalize(temp, temp, 0, 255, NORM_MINMAX, CV_8U);
129 
130                 optimalResize(input, Size(finalSize, finalSize));
131                 optimalResize(temp, Size(finalSize, finalSize));
132 
133                 // varlbp only
134 
135                 temp = (input - temp) * 1.1; // enhanced LBPV
136 
137             LoG(input);
138 
139                 images.push_back(input);
140             }
141             labels.push_back(atoi(classlabel.c_str()));
142             paths.push_back(path);
143         }
144     }
145 }
146 // assign train and test samples from total images
147 
148 Mat enhance(Mat image)
```

```
149 {
150     // nlmeans and ridge enhancement for fingerprint only
151 
152     int denoiseStr = 7;
153     int templateSize = 7;
154     int searchSize = 21;
155 
156     int sharpHigh = 3;
157     int sharpLow = 2;
158 
159     int filterX = 17;
160     int filterY = 11;
161 
162     Mat denoise;
163     image.copyTo(denoise);
164 
165     // denoise
166     fastNlMeansDenoising(
167         image, denoise, denoiseStr, templateSize, searchSize); // remove gaussian white
    noise
168 
169     // sharpen
170 
171     GaussianBlur(denoise, image, Size(filterX, filterY), 0, 0);
172     addWeighted(denoise, sharpHigh, image, sharpLow * -1, 0, image);
173 
174     return image;
175 }
176 
177 Mat gaborFilter(Mat& image, int varGabor, int lam)
178 {
179     //tune parameters
180 
181     Mat gaborKernel = getGaborKernel(KernalSize, pos_sigma, Theta, Lambda, Gamma, psi);
182     filter2D(image, image, CV_32F, gaborKernel);
183     image.convertTo(image, CV_32F, 1.0 / 255, 0);
184     return image;
185 }
186 
187 void featureClassifierCombo()
188 {
189    //clear vectors for iterations of identification
190     trainFused.clear();
191     testFused.clear();
192 
193     Mat trainDataPCA, trainDataPCA2, testDataPCA, testDataPCA2;
```

```
194     trainDataPCA = formatImagesForPCA(trainModality[0]);
195     testDataPCA = formatImagesForPCA(testModality[0]);
196
197     if (fn_csv.compare(fn_csv2) != 0)
198     {
199         trainDataPCA2 = formatImagesForPCA(trainModality[1]);
200         testDataPCA2 = formatImagesForPCA(testModality[1]);
201     }
202
203     Mat hConTrain, hConTest;
204
205     if ((reductionType == "PCAb" || reductionType == "PCAa"))
206         // do PCA after or before
207
208     else if (reductionType == "Gabor") // Gabor after fusion
209     {
210       gaborFilter(trainFused[i], 0, varRed);
211     }
212
213     //using SVM with logistic regression for probablity scores (open-set identification)
214         if (matcher == "SVM" || matcher == "LR")
215         {
216             saveSVMTrainData(trainFused);
217             saveSVMTestData(testFused);
218         }
219
220         else
221             model->train(trainFused, trainLabels); //facerecognizer eigen,fisher and lbph
222     }
223
224    //if not other feature reduction is performed then weigh biometrics (if applicable),
225
226     else if (reductionType == "None")
227     {
228         for (int i = 0; i < testModality[0].size(); i++)
229         {
230             // horizontal concatenation
231
232             // weighted averaging 50/50
233             if (weigh == 1)
234             {
235                 if (i < trainModality[0].size())
236                 {
237                     trainFused.push_back(modalityWeight2 * trainModality[0][i]
238                         + modalityWeight1 * trainModality[1][i]);
239                 }
```

```
240
241                 testFused.push_back(
242                     modalityWeight2 * testModality[0][i] + modalityWeight1 * testModality
    [1][i]);
243             }
244             else
245             {
246                 if (i < trainModality[0].size())
247                 {
248                     hconcat(trainModality[0][i], trainModality[1][i], hConTrain);
249                     GaussianBlur(hConTrain, hConTrain, Size(3, 3), 0, 0);
250                     resize(hConTrain, hConTrain, Size(hConTrain.cols - 1, hConTrain.rows
    - 1),
251                         resizeType);
252
253                     trainFused.push_back(hConTrain);
254                 }
255
256                 hconcat(testModality[0][i], testModality[1][i], hConTest);
257                 GaussianBlur(hConTest, hConTest, Size(3, 3), 0, 0);
258                 resize(hConTest, hConTest, Size(hConTest.cols - 1, hConTest.rows - 1),
    resizeType);
259
260                 testFused.push_back(hConTest);
261             }
262         }
263         // fusion
264
265         if (matcher == "SVM" || matcher == "LR" && trainingStage = 1)
266             saveSVMTrainData(trainFused);
267         if (matcher == "SVM" || matcher == "LR" && testStage = 1)
268             saveSVMTestData(testFused);
269
270
271         else
272             model->train(trainFused, trainLabels);
273     }
274 }
```

Listing B.10: Main code (masterFusion.cpp)

```
1
2
3 int main(int argc, const char* argv[])
4 {
5     // parameters chosen based on bash scripts
```

```
6
7       fn_csv = string(argv[1]);
8       fn_csv2 = string(argv[2]);
9       samples = atoi(argv[3]);
10      matcher = string(argv[5]);
11      enhanceType = atoi(argv[6]);
12      reductionType = string(argv[7]);
13      varRed = atof(argv[8]);
14      weigh = atoi(argv[9]);
15      finalSize = atoi(argv[10]);
16      noClasses = atoi(argv[11]);
17      modalityTypes = atof(argv[12]);
18
19      if (noClasses == -1 || matcher == "SVM" || matcher == "LR")
20          threshIterations = 1;
21
22      vector<string> paths;
23      vector<string> paths2;
24
25      vector<Mat> images;
26      vector<Mat> images2;
27
28      vector<int> labels;
29      vector<int> labels2;
30
31      // Read in the data and execute modality specific functions where applicable.
32      try
33      {
34          if ((fn_csv.compare(fn_csv2) != 0))
35          {
36              read_csv(fn_csv, images, labels, paths);
37              firstSet = 0;
38              read_csv(fn_csv2, images2, labels2, paths2);
39
40              modalitySpecific();
41              lbphAlignment();
42          }
43          else if ((fn_csv.compare(fn_csv2) == 0))
44              read_csv(fn_csv, images, labels, paths);
45      }
46      catch (cv::Exception& e)
47      {
48          cerr << "Error opening file \"" << fn_csv << "\". Reason: " << e.msg << endl;
49          //        cerr << "Error opening file \"" << fn_csv2 << "\". Reason: " << e.msg <<
     endl;
50          exit(1);
```

```
    }

    if (noClasses == -1) // closed-set identification only
        noClasses = labels[labels.size() - 1] + 1;
    cout << noClasses << endl;
    assignImages(images, labels, paths, samples);

    trainModality[0].swap(trainImages);
    testModality[0].swap(testImages);

    if ((fn_csv.compare(fn_csv2) != 0))
    {
        assignImages(images2, labels2, paths2, samples);

        trainModality[1].swap(trainImages);
        testModality[1].swap(testImages);
    }
    featureClassifierCombo();

    if ((fn_csv.compare(fn_csv2) == 0)) // no fusion
    {
        if (matcher == "SVM" || matcher == "LR" && trainingStage = 1)
            saveSVMTrainData(trainFused);
        if (matcher == "SVM" || matcher == "LR" && testStage = 1)
            saveSVMTestData(testFused);
        else
            model->train(trainModality[0], trainLabels);
    }
    // evaluate

    if (verification = 1)
        evaluateVerfication();
    else
        evaluateIdentification();
}
return 0;
}
```

Appendix C

List of Publications

Brown, D. and Bradshaw, K. Enhanced biometric access control for mobile devices. In *20th Southern Africa Telecommunication Networks and Applications Conference*, pages 118–123. 2017.

Brown, D. and Bradshaw, K. Feature-fusion guidelines for image-based multi-modal biometric fusion. *South African Computer Journal*, 29(1):92–121, 2017.

Brown, D. and Bradshaw, K. Improved Automatic Face Segmentation and Recognition for Applications with Limited Training Data, pages 415–426. Springer International Publishing, 2017. doi:10.1007/978-3-319-58274-0_33.

Brown, D. and Bradshaw, K. Extended feature-fusion guidelines to improve image-based multi-modal biometrics. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*. ACM, 2016.

Brown, D. and Bradshaw, K. A dynamically weighted multi-modal biometric security system. In *19th Southern Africa Telecommunication Networks and Applications Conference*, pages 254–258. 2016.

Brown, D. and Bradshaw, K. An Investigation of Face and Fingerprint Feature-Fusion Guidelines, pages 585–599. Springer International Publishing, 2016. doi:10.1007/978-3-319-34099-9_45.

Brown, D. and Bradshaw, K. Improved fingercode alignment for accurate and compact fingerprint recognition. In *Technologies for Homeland Security (HST)*. IEEE, 2016.

Brown, D. and Bradshaw, K. A multi-biometric feature-fusion framework for improved uni-modal and multi-modal human identification. In *Technologies for Homeland Security (HST)*. IEEE, 2016.