h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACULTY OF COMPUTER SCIENCE

University of Applied Sciences Darmstadt

Master Thesis

Multi-Instance Fingerprint Classification based on Global Features

A thesis submitted in fulfillment of the requirements for the degree of Master of Science (M.Sc.)

Author:
Daniel Fischer
daniel.b.fischer@stud.h-da.de

1st Supervisor:
Christoph Busch
christoph.busch@h-da.de

2nd Supervisor:
Pawel Drozdowski
pawel.drozdowski@h-da.de

Thesis Begin: October 11, 2017
Submission: April 23, 2018

Selbständigkeitserklärung

Ich, Daniel Fischer, versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Datum	Unterschrift

Abstract German

Fingerabdrücke sind eines der am häufigsten verwendeten biometrischen Merkmale und finden Verwendung in einer Vielzahl von Systemen. Oft werden sie für die Identifizierung von Personen verwendet, indem deren Fingerabdrücke gegen diejenigen verglichen werden, die zuvor in einer Datenbank gespeichert wurden. Solche Datenbanken werden beispielsweise von Strafverfolgungsbehörden betrieben. Sie enthalten dann z.B. die Fingerabdrücke von registrierten Straftätern oder unbekannte Abdrücke von Tatorten. Die Anzahl der in solchen Datenbanken gespeicherten Fingerabdrücke wächst stetig. Schon jetzt verfügt das FBI über eine solche Datenbank, in der ca. 230 Millionen Abdrücke gespeichert werden. Das suchen eines Abdruckes in einem solchen System kann sehr lange dauern, da im einfachsten Fall alle gespeicherten Fingerabdrücke mit einer entnommenen Probe verglichen werden müssen. Um die Anzahl der Vergleiche zu reduzieren, verwendet man Algorithmen um eine Vorauswahl von Fingerabdrücken zu treffen. In dieser Arbeit geht es darum, mehrere Fingerabdrücke eines Subjektes zu verwenden, um die Subjekte innerhalb der Datenbank anhand der Klassen ihrer Fingerabdrücke zu gruppieren, sodass eine Probe lediglich gegen Subjekte verglichen werden muss, deren Fingerabdrücke dieselben Klassen besitzen. Für die Klassifizierung sollen ausschließlich globale Eigenschaften des Fingerabdruckes, z.B. der Verlauf der Fingerabdrucklinien, verwendet werden. Der erste Teil dieser Arbeit besteht aus einem Survey von Ansätzen, die in diesem Bereich publiziert wurden. In einem nächsten Schritt wurde die öffentlich zugängliche SD9 Datenbank des NIST mit dem Ziel untersucht, die Korrelationen zwischen den Klassen der Fingerabdrücke eines Subjektes zu bestimmen. Mit den daraus gewonnenen Informationen konnte ermittelt werden, dass die Verwendung mehrerer Fingerabdrücke für die Gruppierung theoretisch bis zu 94% weniger Vergleiche notwendig macht. Mithilfe eines im Rahmen dieser Arbeit erstellten Systems und der Verwendung selbst trainierter neuronaler Netze für die Fingerabdruckklassifizierung, konnte gezeigt werden, dass bei Verwendung der Klassen von lediglich 3 Fingern einer Hand, die Anzahl der notwendigen Vergleiche von Subjekten um bis zu 80% reduziert werden kann und selbst die Verwendung von ungenauen Klassifizieren eine Verbesserung möglich macht.

Abstract English

Fingerprints are one of the most used biometric characteristics. They are used in a wide range of applications, as the identification of subjects within a database. Therefore, the fingerprints need to be compared against those stored in the database, what is often done by law enforcement agencies like the FBI. Their database contains around 230 million fingerprints from inter alia recorded criminals or unresolved traces from crime scenes. Identifying a fingerprint within this database can take a long time. Since the naïve approach of comparing each fingerprint subsequently would result in long identification time, often a pre-selection is used to reduce number of comparisons. Within this work, we will use multiple fingerprints of a subject to create a binning of the subjects towards the classes of those fingerprints, so that only subjects whose fingerprints share same classes need to be compared. The classification of the fingerprints should be done using only global information of the fingerprint pattern, like the orientation of the ridgelines. The first part of this work is therefore a survey on the different approaches for fingerprint classification using those features. After that, the NIST SD9 database was analyzed towards the correlations between the fingerprint classes of the subject. With the derived information we can show, that using multiple instances of fingerprints for the binning of the database can result in up to 94% less comparisons for identification assuming perfect classification. The use of the system for multi-instance classification defined in this work, together with the trained neural networks for fingerprint classification enabled us to reduce the number of required comparisons by up to 80%, while using just 3 fingers of a subject. In addition, it was shown, that the use of classifiers with only moderate classification accuracy allowed a reduction of comparisons.

Dedication

I want to thank my supervisors Christoph Busch and Pawel Drozdowski for their support during the work and their comments on the thesis.

I further want to thank my family and my girlfriend Linda for their help and their sympathy and support, especially during the last stressful phase of this thesis.

Special thanks goes to the BKA, which kindly supplied information that I used for some of the statistics that were calculated during this work.

Contents

Acronyms

AFIS	Automatic Fingerprint Identification System
OM	Orientation Map
DI	Directional Image
CCR	Correct Classification Rate
GHM	Gaussian-Hermite Moment
NIST-SD4	NIST Special Database 4
NIST-SD9	NIST Special Database 9
NIST-SD14	NIST Special Database 14
NIST-SD24	NIST Special Database 24
FBI	Federal Bureau of Investigation
EU	European Union
ROI	Region of Interest
BPA	Back Propagation Algorithm
KL-Transformation	Karhunen-Loéve Transformation
SOM	Self-Organizing Map
MLP	Multi-Layer Perceptron
PCASYS	Pattern-Level Classification Automation System
PNN	Probablistic Neural Network
BPN	Back Propagation Network
NN	Nearest Neighbor

FFT	Fast Fourier Transformation
DCT	Discrete Cosine Transformation
RBF	Radial Basis Function
MM	Markov Model
HMM	Hidden Markov Model
SVM	Support Vector Machine
kNN	k-Nearest Neighbor
DFT	Discrete Fourier Transformation
VQ	Vector Quantization

1. Introduction

Since the time, that fingerprints were found to be a proper biometric characteristic for identification tasks, in the early 20th century, they were adapted into a wide range of applications and present in many aspects of our everyday life. Especially for civilian and crime investigation purposes, huge amounts of fingerprints are stored within the databases of fingerprint identification systems. E.g. the United States' FBI fingerprint database already contains ca. 135 million fingerprints [1] of civilians as well as criminals and is therefore one of the biggest datasets of its kind. But as the time goes by, additional fingerprints will be stored to the system and increase a major drawback that comes together with the size of the database: the workload and directly connected, the response time of the systems. Since the naïve approach of comparing each fingerprint subsequently to a captured probe is very inefficient, another approach has to be done to reduce the number of comparisons.

The reduction of the workload is the key motivation for this thesis. Since various approaches exist on the classification of fingerprints that can serve for the task, a further step is done within this thesis towards the application of multiple instance of fingerprints for classification purposes and database binning.

1.1. Research Questions

In the following, the research questions for this work are outlined. They served as the basis for the work packages and allow the definition of milestones for the project. They further imply a basic structure for this work as defined in section 1.2.

RQ1 What are the state-of-the-art classification features for fingerprint

databases?
It is intended to make a deeper related work study in order to find global features (Level-1) of a fingerprint that are used for classification purposes and outline their performance. Further this will give a description of possible classes that can be investigated and utilized.

RQ2 How are the fingerprint classes distributed?
The task is to find proper datasets on which analysis of the class distribution of fingerprints can be done. A further analysis is planned to find possible correlation of the fingerprint classes from adjacent fingers of the subjects. This aims for an understanding of possible error classes and the correlation between those fingers. In addition, existing statistics and works should be mentioned in the related work study.

RQ3 Is there a theoretical improvement for multi-finger classification?
Depending on the findings of the related work study and the distribution of fingerprint classes, a statement on theoretical improvements for multi instance classification should be given.

RQ4 Can multi-finger classification improve fingerprint identification systems?
Is it possible to utilize the classes of multiple fingerprints of a subject for a finer-grained binning of the databases to reduce identification workload without decreasing biometric performance of the overall system? For this task, a baseline system based on neural networks is going to be extended to multiple fingerprints and evaluated.

1.2. Structure of the Work

The structure of this work is oriented towards the order of the research questions stated above.

- Initially chapter 2 will give some fundamental information on fingerprints and biometric systems in order to get a general understanding about the context of this work.

- Chapter 3 then introduces further information that are related to fingerprint classification and presents a survey on works that deal with classification techniques based on global classification features.
- The statistical analysis of fingerprint databases towards their distribution of fingerprint classes is the topic of chapter 4.
- Chapter 5 then presents a possible fingerprint classification system that is based on multiple fingerprint instances and utilizes the findings of the statistics chapter for improved classification performance and workload reduction towards identification scenarios.
- Testing of the proposed system is described in chapter 6 and the results are discussed subsequently together with some ideas for further improvements.
- The conclusion in chapter 7 will sum up the results of the chapters shortly and give an outlook to future work.

2. Fundamentals

The fingerprint is one of the most used biometric characteristics [2], which is evident when looking to Figure 2.1. Its manifold applications reach from end-user authentication on personal devices like smartphones or notebooks and mobile banking, over access control systems for buildings in public and private sector up to the usage within law enforcement and public administration. In this section the fundamental information on fingerprints as a biometric characteristic as well as its biological background is given. The second part of the chapter covers biometric systems in general and outlines their basic workflow and operational modes.

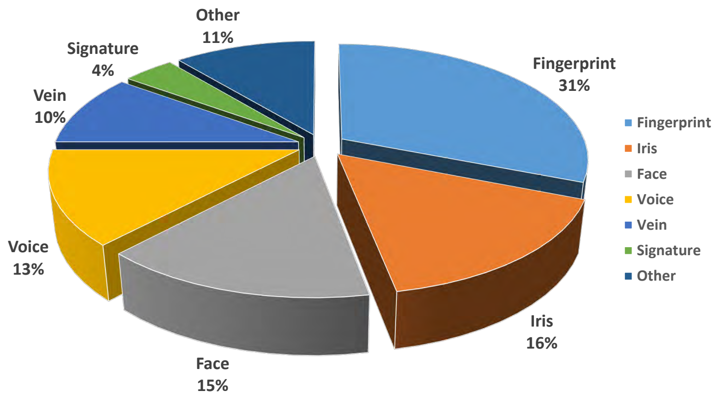


Figure 2.1.: Biometric market share by characteristic [3]

2.1. Fingerprint as a biometric characteristic

Following section will give some fundamental information on fingerprints, their formation and patterns. The basic properties in relation to biometric systems and the suitability as a biometric trait are denoted afterwards.

Figure 2.2.: Example fingerprints from left to right: dry, normal and wet impression (created by SFinGe)

2.1.1. Foundations

The observable unique ridge and valley structure (see Figure 2.2) on the surface of the fingerprint can be used as a biometric characteristic. Figure 2.4 shows a fingerprint with the marked positions of so called minutia points (endings of ridgelines in red and bifurcations in blue). If a sufficient number of minutia points (typically a minimum of 12 minutiae [2]) is available, the combination their positions and directions can be used for the identification of a subject.

Nowadays, the usage of latent fingerprints is very common in crime investigation whereas fingerprints in general serve for a variety of applications like authentication or access control as mentioned before. As the first person that found indications for the uniqueness of fingerprints based on empirical studies, Henry Fauld laid the foundation for the development of the systems, we know today, back in 1880. Later, in the beginning of the twentieth century, the understanding of fingerprints, its formation and individual patterns proceeded so far that their application for identification became widely accepted [2], especially for forensic purposes. This was the beginning for a continuously increasing number of application areas of fingerprints.

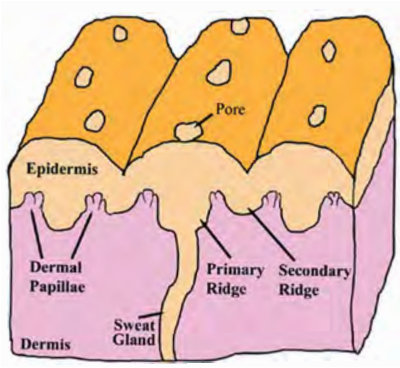


Figure 2.3.: Different layers of tissue within a finger [4]

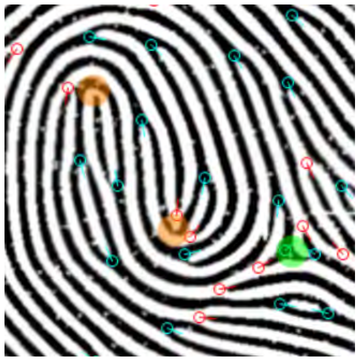


Figure 2.4.: Fingerprint with marked minutia, cores and delta

2.1.2. Fingerprint Formation

The formation of fingerprints starts during the fetal development and is affected by genes as well as diverse environmental conditions within the womb. That is why even monozygotic twins differ in the patterns of their fingerprints, though there seems to be some correlation in their basic structure. [5]

Human fingers consist of different layers of tissue (compare Figure 2.3) that form the visible structure on its surface. The outermost layer that is directly showing up this structure is the epidermis. The underlying dermis is a tissue having the same structure as the epidermis. Its function is to rebuild the outer skin layers in case that they are destroyed, while preserving the pattern.

Taking a high-resolution image of a finger shows small pores on the ridgelines that are the ends of the sweat glands. Beside the ridge and valley structure, these pores can further increase the distinctiveness of the finger. However, since the sweat glands emit fluids to the surface of the finger, they can have impact on the quality of images taken from fingerprints. Especially a very high or very low amount of emitted fluid affects the presentation of fingers to contact scanners (compare Figure 2.2).

2.1.3. Biometric Properties

In terms of biometric systems, the biometric characteristic used by the system, needs to fulfill some basic properties to guarantee that preferably all targeting individuals can use it and are willing to. Other properties are mandatory to make sure that the system is capable of distinguishing the individuals correctly and to describe the security level of the characteristic. Forging the characteristic is a hard task to ensure security.

In the following, these properties (firstly defined in [6]) are outlined for fingerprint patterns as biometric characteristic as described in [2, 7, 8].

Collectability Taking an impression of fingerprints can be done very easy with a variety of different capturing devices and sensing techniques [2]. For most biometric systems that use fingerprints as the characteristic of choice, the quality of the images highly depends on the capturing devices on the one hand and the cooperation of the subject on the other.

Universality The universality of fingerprint is given by natural presence of ten fingers for most people, with exception of those individuals suffering from illnesses that affect their fingerprint patterns or those where amputation of fingers, hands or arms were performed.

Uniqueness Any of a person's ten fingers is claimed to show a unique pattern build by the ridgelines, given a sufficient number of visible minutiae. [9].

Invariability Fingerprint patterns were found not to change during a person's life [10]. While they grow together with the person, no altering in the actual pattern can be seen, so that the invariability due to natural factors is very high. Even damages of the pattern by small cuts or bruises only cause temporary alteration, since the skin is able to heal itself. Only continuous damaging or serious injuries like deep cuts or even amputations would result in permanent alteration.

Performance The performance indicates how well fingerprints suit for development of biometric recognition systems. It takes into account the recognition accuracy, throughput of the system, hard and software requirements as well as dependency on external factors. Fingerprints offer good performance due to the high distinctiveness, good and cheap collectability as well as low resource requirements for its processing.

Circumvention In order to provide good security, it should be hard to forge the biometric characteristic. Since the hands and fingers are our main tool in everyday life, we leave our fingerprints on any surface that we touch, allowing forgers to capture and use these latent prints. Therefore, it is more easy to copy fingerprints then for example iris images. By using countermeasures as liveness detection or other presentation attack detection (PAD) methods, biometric systems try to overcome these drawbacks.

Acceptability Since first approaches of using fingerprints for forensic applications, they were continually linked to crimes. But as our society evolves, the acceptance of fingerprints in our everyday life is still increasing. Our fingerprints are saved on the identity cards and the biometric characteristics replace login credentials or tokens required for online banking. Further attempts to improve the acceptance contain contact-less sensors to reduce interaction and a proceeding integration of biometrics in governmental applications.

Concluding the fingerprint as a biometric characteristic, one can find that it is suitable for a variety of applications. The individuality of its pattern and the ease of collecting impressions with different sensors integrated into most of the latest smartphone and notebook models shows the hight potential of this characteristic.

2.2. Biometric Systems

A biometric system as defined in [11] is a system used for recognizing individuals based on their behavioral and biological characteristics. The following section will give an overview on a reference biometric system, its subsystems and its internal workflows as it is described in the ISO/IEC 19795-1 [12].

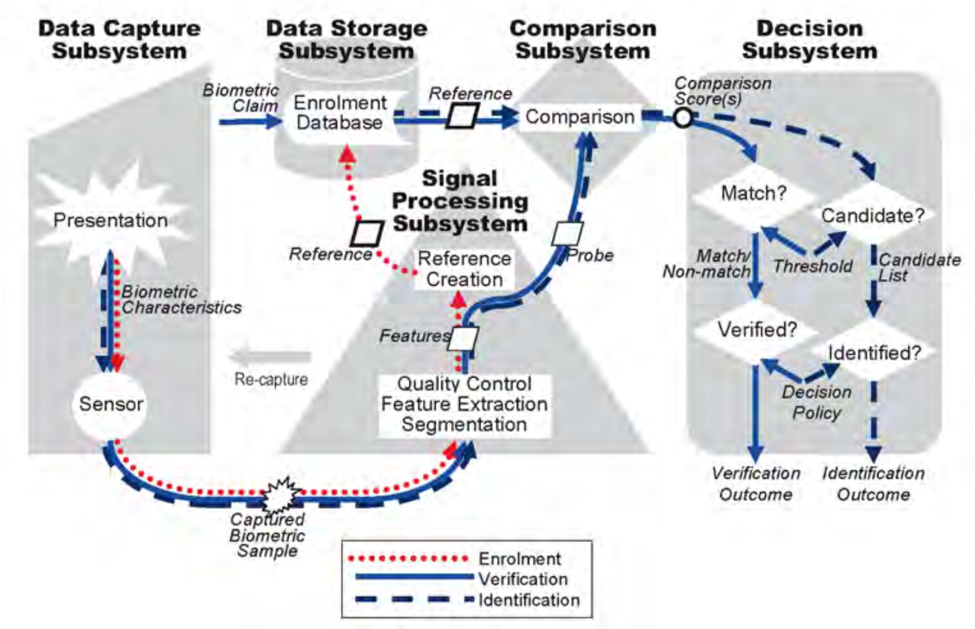


Figure 2.5.: Standard Biometric Recognition System [12]

2.2.1. System Overview

A biometric system, as depicted in Figure 2.5 consists of five major subsystems, described in the following.

Data Capture Subsystem The 'Data Capture Subsystem' is responsible for capturing a biometric sample from a presented biometric characteristic. For this purpose the systems can make use of one or multiple sensors. The captured biometric sample is then processed in the subsequent system.

Signal Processing Subsystem This system fulfills three tasks based on the received biometric sample.

1. During the **segmentation**, the system processes the sample to differentiate between foreground and background areas. The foreground defines the region of interest, that contains the usable information, whereas the background consists of unwanted information (e.g. noise in digital images).
2. In the **feature extraction** phase, the systems extracts biometric features from

the segmented data. These features are later used for the creation of a biometric reference, that might be enrolled to or compared against the enrollment database. 3. Before the actual **reference creation**, a quality control is performed on the extracted features. Its purpose is to detect features of low quality and potentially reject captured samples and initiate a recapture of the characteristic.

Data Storage Subsystem If a captured sample successfully passed the signal processing during the enrollment of a subject, the 'Data Storage Subsystem' is used to store the create reference of the subject together with a reference identifier. It holds the references of all subjects enrolled to the system within an enrollment database and provides them the comparison system, as needed.

Comparison Subsystem This subsystem is intended to compare a probe that was extracted from a biometric sample in the signal processing system, against one reference from the enrollment database for verification or against multiple references for identification. The output then is one or more comparison scores indication the similarity of the probe and the reference(s).

Decision Subsystem Within the 'Decision Subsystem' a final outcome of the biometric system is calculated, based on the comparison scores obtained by the 'Comparison Subsystem', predefined thresholds and further decision policies. In terms of verification, the systems states if a probe matches to the compared reference or not. For identification, the outcome can be the identified subject, a list of most likely candidates or nothing in case no candidates were found.

2.2.2. Biometric Fingerprint Recognition System

Further details on the previously defined subsystems can be given in the context of fingerprint recognition. Within the 'Data Capture Subsystem', different types of sensors can be used to capture one or multiple fingerprint of a subject at a time. These days, different types of sensors based on various technologies are in use to capture fingerprint. The most common sensor types are the optical and capacitive

Figure 2.6.: Fingerprint image from capacitive (left) and optical sensor (right) - created by SFinGe

ones. The former take a visual impression of the finger, while the second ones utilize electrostatic effects to detect the ridge and valley structures in order to create the image (see Figure 2.6).

The 'Signal Processing Subsystem' combines image enhancement and segmentation techniques to extract the biometric features of the sample. The subsequent image quality estimation following the ISO quality specifications [13] and [14], is then used to determine if the extracted sample can be used as a reference that is stored in the database or a probe for comparison purposes. In fingerprint recognition systems, the enhancement often includes brightness and contrast optimization to increase the visibility of ridgelines. Various approaches exist for the segmentation, which searches for the area of interest within the image and discards background information like noise (compare Figure 2.7). Examples would be measurements of pixel gray values, utilizing ridge orientations within pixel blocks and even neural network approaches exist[2]. Afterwards, the actual feature extraction can be performed.

The subsequent quality estimation can be done using the reference algorithms for [14] - NFIQ2. It checks if the processed image and extracted features can fulfill the minimum requirements on the quality to perform subsequent steps or if a recapture of the finger is required.

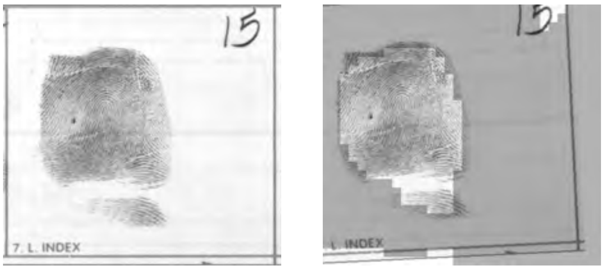


Figure 2.7.: Original (left) and segmented image (right) - from SD9

2.2.3. Operational Modes

Biometric systems can be operated in two different modes, visualized by the blue shaped arrows in Figure 2.5, namely verification and identification. During the verification and identification processes, the ‘Signal Processing Subsystem’ creates a biometric probe of the acquired sample (as described in the previous section) that can be compared to one or multiple stored references, within the ‘Comparison Subsystem’. The comparison score(s) obtained in this step are then utilized in the ‘Decision Subsystem’ to determine either verification or identification outcome.

Verification The intention of both, verification and identification is slightly different. The verification process is used to check a presented biometric sample against the reference sample of a claimed identity. This is denoted as 1:1 comparison whose outcome is either a match or non-match between probe and reference.

An example would be an access control system for a building based on fingerprints with an additional identity claim. The fingerprints of every person with access rights are enrolled to the database and linked to the identity of the subject. When a subject wants to access the building, it makes a claim of its identity and presents its fingerprints to create a probe that can be compared against the reference stored in the enrollment database. If the comparison score exceeds a given threshold, the identity can be verified and the subject is granted access. Otherwise, the access is denied.

Identification The identification of a subject is a more challenging task since no identity claim is given. In this scenario, the probe extracted from a presented biometric sample has to be compared against multiple references within the 'Data Storage Subsystem'.

With a naïve approach this results in n comparisons of the probe, where n is the number of enrolled subjects. The outcome is a candidate list containing the subjects with the highest comparison scores above a defined threshold. A decision policy is then required to produce the final identification outcome, which might be the identified subject, a list of subjects or nothing.

The workload of such a system during the identification process increases with each additional references enrolled to the database, due to required additional comparisons. For that reason, some of the systems implement steps for workload reduction. Two major approaches aim for reducing the total number of comparisons.

Indexing approaches try to create a logic order of the references in the database. If a probe should be compared to the database, the index of the probe is calculated and all references with similar indices are determined.

The second major approach is the **classification**. It is used to separate the database into multiple bins or sub databases in with all of the containing references are members of the same class. This allows us to reduce the workload, by comparing a probe only against references that are of the same class. In case of fingerprints, these classes are often derived by the observation of different pattern shapes (see chapter 3).

Example of Identification System A very prominent example of such an identification system is the Integrated Automated Fingerprint Identification System (IAFIS), which was established in 1999 and is operated by the US Department of Justice. Later this system was merged to the Next Generation Identification

(NGI) System. Currently 135,093,826 fingerprint (civilian and criminal) are enrolled to the system [1] by January 2018. Thereof around 2,763,222 subject are included within a sub-database called ‘Repository for Individuals of Special Concerns (RISC)’ that is used for ‘Rapid Fingerprint Identification’ and has an average response time of around 8 seconds. The response times for requests of criminal receipts were around 10 minutes for urgent request and ca. 15 minutes for non-urgent (in year 2017).
These facts show, that the high number of subjects enrolled results in high response times for the system. The adaption of the RISC dataset might be a result of the response times, because time critical requests could not be handled otherwise. This approach indicates, that splitting a database into multiple bins and concentrate search to smaller parts of the database will decrease the workload of the system and increase response time, since fewer comparisons are required.

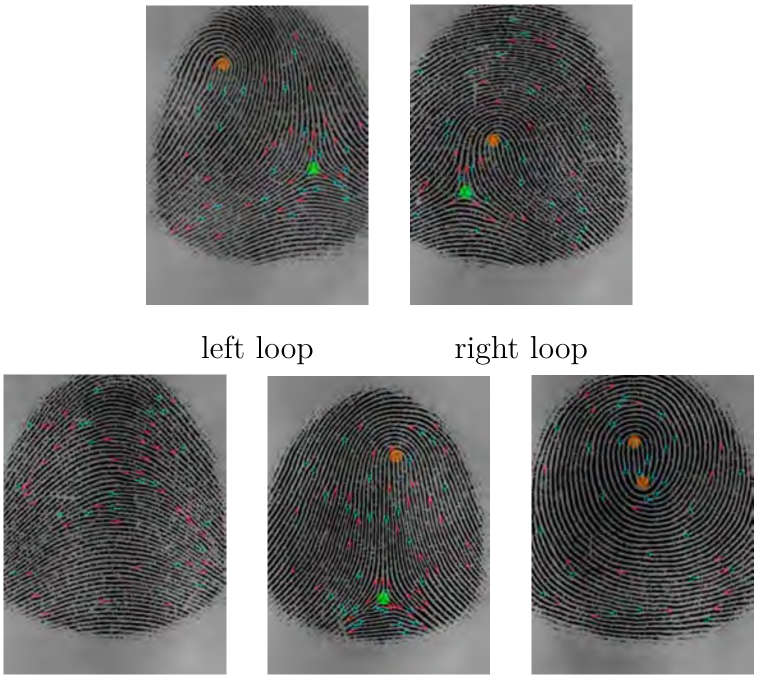
This example well illustrates the context in which the present work was created. Within this work, we follow the classification approach for workload reduction. The related work towards this topic is outlined in the next chapter.

3. Related Work

Within this chapter, the related works regarding the thesis topic are revealed. The focus here is on papers about fingerprint classification using global features. The intention of studying these works and doing a structured analysis of their results is having a good overview of past work on the one hand and finding the most promising approaches that can serve as a foundation of this work on the other hand. The surveys of Galar [15, 16] and Kavati [17] served as a starting point for the study. Since the structure of [15] was found appropriate, the subsequent sections are structured in a similar way.

By analyzing the structure of the fingerprint patterns Galton [18] was able to describe three major classes of fingerprints based on the curvature of the ridgelines, namely: whorls, arches and loops. Henry [19] further refined these classes into whorls, arches, tented arches, right loop and left loop, given a total of 5 separable classes. Most of the subsequent approaches will follow the Henry Scheme but some of them will used an extended scheme like Moayer[20, 21] (Mairs' classification scheme), or reduced schemes where one or more of the classes will be merged into a combined class. An often-used example is the merging of both arch classes because an automatic distinction can be challenging. The different classes defined by Henry as well as further subclasses are depicted in Figure 3.1. This definition allows us to split up fingerprint databases with respect to these classes. When trying to search a fingerprint within the database, we can reduce search time and the amount of fingerprints compared to the searched, by then only determining its class and first perform search for the enrolled subjects sharing the same class. The separation of the database into chunks of fingerprints holding the same class is called binning and the chunk are referred to as bins.

Figure 3.1.: The five most used fingerprint classes (created by SFinGe)



left loop right loop

arch tented arch whorl

3.1. Fingerprint Classification Process

In order to achieve the classification, there are 3 to 5 general steps to be performed after receiving an input image. The number of steps depends on the type of classification approaches. Works using machine learning need to train their classifier before it can be used to classify new images. If multiple classifiers are used, fusion metric is required in order to combine their outcomes. In the following, these basic steps are described.

Preprocessing

The preprocessing is used to prepare the input image for the feature extraction step. While some of the reviewed works lack in information on whether they used preprocessing or not, few papers distinguish only in the algorithm used for preprocessing in order to receive better performance in feature extraction. The preprocessing can consist of different enhancements. For example increasing the contrast of the image for better detection of the ridgelines, filters to reduce the noise or the segmentation of the fingerprint image into a Region of Interest (ROI) and background. Most of the classification algorithms utilize segmentation to remove the background areas that do not contain information on the fingerprint pattern. Further enhancements are skeletonization and binarization.

Feature Extraction

The second step is to extract features out of the (enhanced) fingerprint image that are capable of describing the class of the fingerprint pattern. A first starting point at of nearly all approaches is the creation of the Orientation Map (OM). It is either applied as a feature or used for derivation of other features like singular point. Some of the approaches use a registration point to define a region of interest within the fingerprint. Further information on the features used by the different algorithms are given in Section 3.2.

(Train Classifier)

In case of neural network based classification approaches, the supposed systems need to be trained on real data in order to be able to classify fingerprints. Although it is a good idea to use disjunct data for training and testing to prevent overfitting of the classifier, some of the approaches present results with overlapping training and test data.

Classification

In this step the actual classification of a sample fingerprint is done. Depending on the extracted features, a classifier assigns the fingerprint to the most suitable class. Because not all approaches classify the fingerprints into the same number of classes, we have to take care of this fact when comparing them. To deal with fingerprints that are of low quality or for which no class can be determined, some of the works deploy criteria to reject such fingerprints or classify them as Unknown. The main reason for that is to prevent the algorithm from classifying the fingerprint into wrong classes.

(Classifier Fusion)

To further improve classification performance, several approaches utilize multiple classifiers and features. The outputs of the different classifiers need to be fused to obtain the fingerprint class. Different strategies are used to achieve the fusion, like majority vote, winner takes all or weighted decision.

Classification and Rejection Metrics

To measure the performance of classification algorithms and allow comparison of different approaches, some metrics are defined. The Correct Classification Rate (CCR) describes the relative measure between the correctly classified fingerprints and all fingerprints that were tested. With respect to that, the classification error can be described by the proportion of falsely classified fingerprints over the test samples. When binning is applied to the database regarding the fingerprint classes, the so called Bin-Error-Rate defines the probability, a fingerprint is assigned to a wrong bin. It is an equivalent to the classification error rate. In some papers, a rejection rate is pointed out. It measures the amount of fingerprints that were excluded from the classification process at some point within the algorithm based on predefined rejection rules. The rejection correlates with the CCR insofar as increasing the number of rejected samples will reduce the classification error and increase CCR.

3.2. Global Classification Features

Before facing the different approaches for the actual classification of fingerprint patterns, the underlying features are described in the following.

When looking at a fingerprint image one can observe different types of features within it. They can be described as level 1, level 2 and level 3 features [2]. Level 1 features give a coarse description on the fingerprint pattern, which makes it useful for classification of similar fingerprints. Information like the ridge flow directions and detected singular points (see subsequent paragraphs) are used in this scope. The level two features are given by describing single ridgelines in the form of minutia. These information e.g. bifurcation (splitting of a ridgeline) or ridgeline ends allow a finer description of the fingerprint, which can make it unambiguously identifiable as long as sufficient minutia points can be found. For level 3 features, high resolution images are required to detect the width of ridgelines, sweat glands

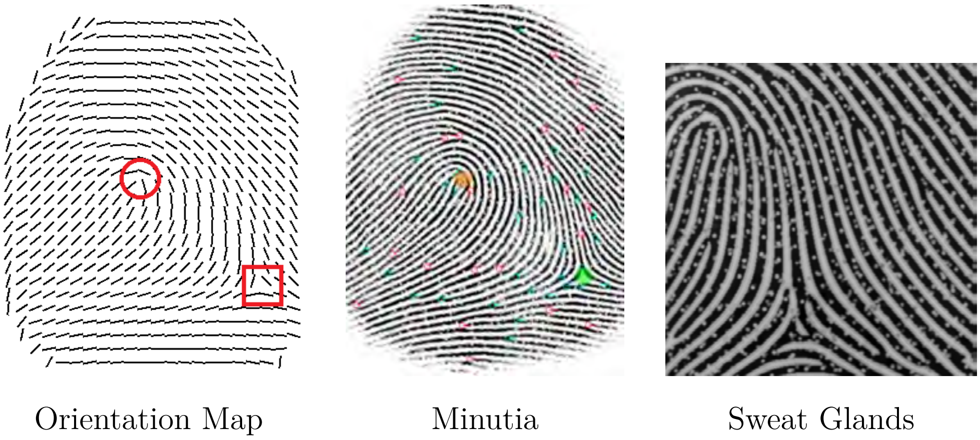


Figure 3.2.: Visualization of the three feature levels

or distance between ridgelines. Especially the sweat glands can serve as highly distinctive feature for fingerprints. In Figure 3.2 a visualization of the different feature levels can be found.

As mentioned before, level 1 features are the means of choice for classification of fingerprints. They are also called global features of the fingerprint and allow a general/coarse description of the fingerprint pattern. Together with the help of classification schemes like Henry's [19], algorithms can be developed that are capable of classifying fingerprints into the different classes.

Reviewing the works on classification approaches and analysis of the used features leads to the following general features.

Orientation Information

The orientation information is derived by the analysis of the local ridge flow within the fingerprint image. This can be done by for instance frequency filters, slit sum [22] or gradient methods. In the most approaches, an OM is calculated block wise, containing the averaged direction, instead of using orientations on a pixel level.

Singular Points

The singular points or singularities are defined as the area of the greatest curvature changes[18]. Two types of singular points can be described: 1. the core point, at which the ridgelines seem to converge 2. the delta point, where ridges tend to diverge. Many approaches make use of these points either as features or alignment of the image. Many extraction algorithms exists for the singularities, that are described in[15]. To mention one of the most used extraction algorithm, the Poincaré method should be mentioned. It uses the orientation information by measure the directions around a given point. A singular point is found, if the directions behave as defined in 1. and 2.

Structural Information

The structural information of fingerprints is derived by tracing the ridgelines. It is used to build representative graphs or derivation of features like ridgeline curvature changes.

Frequency Filter Responses

These features are obtained by applying filter function (like gabor or fast fourier) to the image or OM or parts of it. The main reason for their application is their ability to detect the ridge/valley pattern of the fingerprint. The filter response depicted in figure 3.3 is the so called Fingercode invented by Hain et al.[23].

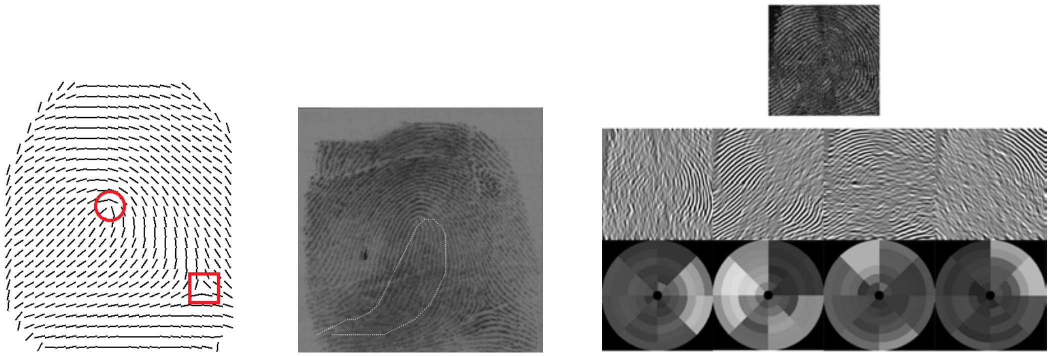
A visualization of the outlined features can be found in Figure 3.3. More detailed information on the approaches to determine these features can be found in [15].

3.3. Classification Approaches

The following section presents an survey of fingerprint classification algorithms based on global features.

3.3.1. Syntactic Classification

The syntactic classification approaches can be found in the early beginning of automatic fingerprint classification[20, 21, 24]. They share the idea of describing the ridge structure by a number of symbols that code different structural information.



Orientation Map & Singular Points

Ridge Flow & Pseudo Ridges

Filter Responses

Figure 3.3.: Visualization of outlined global fingerprint classification features

By analyzing this code with a defined grammar over the symbols, they can make a decision of the fingerprint class.

First approaches were made by Moayer [20] for distinguishing 7 classes. The main idea here is to use directional information from the OM and interpret them block-wise to obtain a feature set of 64 symbols (syntactic coding). This representation allows the classification of the feature set based on a defined grammar for a syntax-free language. In [21], they used more detailed directional information (256 symbols) and statistical syntax-free language to determine subclasses of these used in [20]. For the first approach results are based on very small database (92 images) gave a CCR of 92.5% while rejecting 13% of the images. Results of the second approach were provided for sub classification of one class. Therefore, they are not representative for the test set.

Another approach was made by Rao [24]. They used representative characters to build a string that describes the ridge flow based on the OM. Then a context-free grammar is used to determine the class. Tests were done on a small test set (60 images) only, where 91.66% CCR could be achieved.

The only later approach using a grammar for classification was found in Chang's work [25]. His approach based on ridge distribution derived by extracting ridge patterns. He defined a number of different ridge patterns that can occur within fingerprints, represented by characters. Starting from the bottom of the image, he creates a sequence of the characters depending on the ridge types that were crossed. By utilizing a non-deterministic automate, the fingerprint class is retrieved. An analysis on the NIST Special Database 4 (NIST-SD4) shows 93,4% CCR for 7 classes and 94.84% CCR for 5 classes while rejecting 5.1% of the fingerprints containing error-prone sequences.

3.3.2. Rule-Based / Fixed Classification

The common aspect of the following approaches is their classification utilizing predefined rules for categorization of the features used. Therefore, no further training of the classifier is needed. For instance, the number of cores and deltas is used to define rules for classifying fingerprints. For some cases, the approaches might use a rejection rule to prevent misclassification.

Rao [26] described the first rule based classification algorithm depending on a ridgeline tracer. While working on a thinned and binarized image the line tracer does a counting of 0s and 1s in different image areas. Based on the results of counting, the fingerprint is categorized into one of 4 classes (the loop classes are merged). Since the algorithm was tested only with a few representative images to show its functionality, no further testing results are available.

Most of the rule-based algorithms rely on extracted singular points. Kawagoe [27] used the number of singular point to have a rough classification in a first step and uses a ridge tracing algorithm around the singular points afterwards to do a finer distinction. Test on a small database of 94 fingerprints show a CCR of 91.48% for 6 different classes. With respect to Kawagoe, Zhang [28] makes use of the ridge tracing only in the case that only one core and one delta was found. Thereby he wants to prevent incomplete images, with missing singular points, resulting in wrong classification. CCRs of 92.7% for 4 classes and 84.3% for 5 classes were archived on NIST-SD4. Wang [29] further extended the approach by introducing a new singular point using Gaussian-Hermite Moments (GHMs). They call it the Core-Delta-Pair and it represents a pair of close by core and delta point, which would be ignored using Poincaré method. Test on the NIST-SD4 show an improved performance for tented arches at a total CCR of 88.6% for 5 classes.

Karu's proposed method [30] uses the number of singular points together with their location information. The connection of core and delta point is checked against the local ridge direction within the area and taken into account for final decision. Different tests on NIST-SD4 and NIST Special Database 9 (NIST-SD9) show CCRs of 91.3% and 90.1% for 5 classes and 93.9% and 91.4% for 4 classes respectively. Similar to that, Ballan [31] uses the relative locations of core and delta to do further classification. The given images were self-made and no results were published. Klimanee [32] names the connections between core-core and core-delta principal axes and additionally measures ridge flow directions in that area to distinguish classes. The algorithm was tested on a small individual database (157 images) and had a CCR of 91.3.% for 6 classes with 4.5% rejection rate.

Further rules are defined by Msiza [33] to receive better classification performance for loops, in case of missing delta points by measuring the loop position within different image partitions. Their tests on the FVC2002-1 reveal a CCR 84.5% and 83.5% for of 4 and 5 without rejecting fingerprints. Webb [34] take up the idea of Msiza and formulated further rules for the absence of delta points to improve recognition of loops, whorls and partial fingerprints. They were able to distinguish between 5 classes in FVC2002-1 with 91.1% CCR and 91.8% in FVC2004-1 without any rejection. By introducing measurements like Center-to-Delta Flow (CDF) of Balance Arm Flow (BAF) and an extended decision tree, [35] achieved an overall CCR of 92.74% on the FVC2000, FVC2002 and FVC2004 databases without rejection for the 4 class problem.

Hong [36] uses a combination of singular point number and position together with the recurrence of ridge types, which they detect with their ridge verification algorithm. Results on the NIST-SD4 show a CCR of 87.5% for 5 and 92.3% for 4 classes without rejection. At 20% rejection rate, they can achieve 92.5% for 5 and

97.6% for 4 classes. Later Liu et al.[37] take up the idea and presented a similar approach. They detect the singular points and three different ridgeline types in a first stage for pre-classification. Then they apply a ridge count between detected cores and deltas to get the final class. The approach was tested on the NIST-SD4 and achieved a CCR of 95.6% for the five-class problem.

By analyzing the OM using Poincaré index, Cho [38] selects the candidates for core points. After applying a filter to eliminate spurious core points, the curvature around the core point is detected and used for classification together with the number of cores. On a self-made database containing 6283 fingerprints, a CCR of 92.3% was achieved for 4 classes without rejection.

Dass [39] proposes the use of Orientation Field Flow Curves (OFFC). He detects the number of representative OFFCs for four different classes and uses thresholds to select the fingerprint class. On the NIST-SD4, the algorithm got 94.4% CCR for the four-class problem.

Wang and Xie[40] use the OM to detect the singular points using a self-developed algorithm based on Poincaré. Classification is done by the number of cores and deltas. Besides that, distinction of classes with one core only is done by analyzing ridge flow around the core block. For the five-class problem, the method achieved a CCR of 82% without rejection and 94% when rejecting 14.4% of the fingerprints of the NIST-SD4.

Besides that, Wang et al.[41] propose to use the number of singular points together with the relative position of cores and deltas for further distinction of loop types and tented arch. The method uses a pixel-wise OM as a basis and slightly modified Poincaré method for singular point detection. On a combined database of FVC2002, FVC2004 and Verifinger_Sample_DB (730 fingerprints), the algorithm achieved a CCR of 96.96% distinguishing five classes while rejecting 0.8% of the images where no singular point were extracted.

Fan et al.[42] try to improve the classification accuracy by using an improved algorithm for singular point detection. In a first stage, they use Hough Transformation to detect candidate singular points from OM of foreground blocks. These points are further evaluated using the Poincaré method and class decision relies on the number of cores and deltas and ridgeline curvature originating from the core. A reduced set of the NIST-SD4 (first 100 fingerprints) was used for testing. The algorithm was able to classify arch, loop together with tented arch and whorls at a CCR of 97%.

The approach of Liu et al.[43] is, to use a combined classification based on the number of singular points, predominant directions and a directional pattern analysis. Therefore, they detect the singular points from the OM by clustering the directions into pattern zones of three major directions and finding the points where the zones intersect. For fingerprints, containing only one core-delta pair, further rules are applied that base on directional pattern analysis of the realigned area around the core-delta connection. Fingerprints with missing singular points or for which no reliable statement on the class can be given are rejected. The algorithm was tested on the NIST-SD4 and NIST Special Database 14 (NIST-SD14). CCRs on the NIST-SD4 are 91.62% for 5 classes and 94.38% for four classes respectively, while rejecting 1.55% of the fingerprints. The test on the NIST-SD14 shows a CCR of 89.15% with 3.07% of rejections for distinguishing 5 classes.

In Dorasamy[44], they propose a clustering of the OM based on 3 orientation ranges. They analyze the intersection of the clusters to determine the singular points and process an alignment of the fingerprint. With the help of their defined rules, the algorithm performs well on the FVC2002-1(92.87%) and FVC2004-1(92.2%) with no rejection for 5 classes. The cross validation on NIST-SD4 states 80.51% at a rejection rate of 12%.

The latest rule based approach by Chua [45] makes use of geometric measurements between cores and the corresponding delta point. For classification, a fuzzy-rule based decision system is used to treat with possible uncertainties by detecting tented arches and loops. The systems shows a CCR of 88.33% (5 classes) and 92.13% (4 classes) on the NIST-SD4.

3.3.3. Machine Learning Classification

The Machine Learning approaches are used to analyze high dimensional features vectors and derive probabilities of class affiliations. In most cases, a statistical process like principal component analysis (PCA) [46] or Karhunen-Loéve Transformation (KL-Transformation) is used to reduce the dimensionality of the feature vector to the most discriminative ones and to keep the size of the classifier manageable. Some of the approaches also implement a reject option for the case, that the classifier output is not significant.

Artificial Neural Network

Hughes [47] were using micro patterns for extraction of OM of size 8x8. They propose a neural network with Back Propagation Algorithm (BPA) takes the OM as input vector for learning a classifier. A handmade database of 1600 fingerprints was used for Training, but no results were provided.

The approach of Bowen [48] shows the usage of average directions from OM and a vorticity map as input for 2 neural networks. The outputs of these networks were further used to a train a Back Propagation Network (BPN) for actual classification. Test on a very small test set (47 images, 12 per class) give CCR of 93.6% for 4 classes.

In the work of Kamijo [49] a ridge tracing algorithm is used to find the characteristic ridge of the fingerprint class. Thereof a feature vector with 256 elements is created that serves as input to the network, which consists of 5 subnetworks for each class. The approach can reach CCR of 86% on a database containing 500 equally distributed fingerprints (5 classes). Using continuous classification, including second most probable class, the algorithm achieves 99% CCR.

Geng and Shen [50] introduced a neural network structure, called 'fuzzy - cerebellar model arithmetic computer (CMAC)'. The feature vector is derived from a 20x20 OM, which is further reduced, using KL-Transformation. The classifier was trained to distinguish between arch and whorl class on the NIST-SD4. The results obtained are 98.2% CCR.

Nagaty [51] proposed a combined feed forward neuronal system with one subsystem for each of six classes. After normalization and skeletonization of the input image, a ridgeline-tracing algorithm is used to detect ridge curvature. The different curvatures found are described by alphabetic symbols, similar to a syntactic coding. Afterwards a statistical analysis of the symbols is done and together with the binarized string used as input vector for the neural network. On a sub database of Egyptian Criminal Evidence Database, CCRs of 98.8% (5 classes) and 99% (4 classes) were observed. Some of the fingerprints were assigned to the 'Unknown' class if no proper classification can be done.

The approach of Mohamed [52] includes the usage of singular point information like number, positions and directions as input vector for a fuzzy neural network. The results obtained from the NIST-SD4 show a CCR of 98.5% when distinguishing 5 classes. It should be denoted, that training and testing was done on the same data.

A filter based feature vector is used in Jin's [53] work. By clustering the fingerprint image and applying Discrete Cosine Transformation (DCT) 64 coefficient are derived. These are further processed by Fisher's Discriminant Analysis to obtain training features. Classification is done by a neural network using a Radial Basis Function (RBF). The algorithm is supposed to have a faster running time than other approaches and is able to achieve 91.4% CCR on the NIST Special Database 24 (NIST-SD24) for 5 classes.

The algorithm in [54] includes a Vector Quantization (VQ) approach. Clustered images consisting of 8x8 blocks for each of the 5 classes are used for training. The VQ algorithm creates a codebook of size 4 that is used for classification. By calculation the Euclidean Distance between input vector and stored codebooks, a class decision can be done. On a small test set of only 50 images, the algorithm was able to achieve 80% CCR.

In [55] Wang et al. propose a algorithm based on a Deep Neural Network. The algorithm works without any preprocessing and the OM is used as feature for the network. After training the classifier on four classes, it is able to determine the probability of class memberships for the fingerprint. Test were performed on NIST-SD4, using the first half for training and second for testing. For distinguishing four classes, 91.4% CCR can be achieved without rejection. With a rejection of 1.8%, the CCR improves to 93.1%.

Multi-Layer Perceptron (MLP)

Wilson et al. [22] described a system using the vectors of the OM as features for a MLP network. After the preprocessing with Gabor filters and finding a registration point, the derived vector with 1620 elements was reduced to 96 features using KL-Transformation. The proposed network was able to achieve 89.2% CCR on the NIST-SD4 with 10 percent rejection rate. They further improved the setup of the network [56] and obtained 90.2% CCR at 10% rejection, on NIST-SD4 for 5 classes. In a later work [57], they tested a new sine activation function for the MLP network and were able to further improve the performance. Testing was done on the NIST-SD4 and gave CCR of 92.2% without rejection and 96.57% at 10% rejection.

In the approach of Sarbadhikari [58], the MLP with BPA is used to classify the filter responses of the Fast Fourier Transformation (FFT) of different directional band. For each bands response a histogram is created giving the powers of 256 frequencies which serve as input for the MLP. The algorithm was tested on images that were derived from the training images, by adding some noise. Results up to 100% CCR were achieved, but overfitting effects have to be taken into account, so that they are not meaningful.

Self Organizing Map (SOM) / Kohonen Map

The approach of Moscinska [59] proposed the usage of SOM for the extracted OM. In a first step, image enhancement and skeletonization is done before OM extraction. Then the SOM is trained to find delta patterns and their position within the OM and distinguish between arches, loops and whorls. Around 80% of 100 fingerprints were classified correctly.

Halici [60] extended the idea of Moscinska to the use of Multiple SOMs. The OM and certainty map (derived during segmentation process) are used as feature vectors (64 elements each). After a core based alignment, the features can be used to train the SOMs. Since the proposed systems does a classification to more than the Henry classes, an additional BPN is used to do the mapping to the Henry classes. The proposed method was tested on NIST-SD4 and received 81% CCR for 5 classes.

The SOM algorithm defined in [61] uses extracted singular points from the OM for a correction of the OM in a first step. The used feature space is derived from the 32x32 OM. A deterministic forgery algorithm is applied for clustering based on some prototypes. The proposed SOM is capable of classifying 1600 fingerprints of a NIST database into 4 classes with a CCR of 88%.

A recent approach by Borra et al.[62] proposes extensive preprocessing and an Adaptive Genetic Neural Network for classification. The preprocessing contains a denoising of the input image and morphological transformations for image enhancement. An extracted feature vector (no further information given) is used for training a Feed-Forward Neural Network. With the use a Genetic Algorithm, the weights of the network are adjusted to gain better results. The algorithm was evaluated on the FVC2000 database and gave a CCR of 97.56%. Since they lack in giving information on the used features and the number of classes, the results are not comparable.

Support Vector Machine

Since Support Vector Machine (SVM)[63] is a machine learning approach that is designed for solving two-class problems, its application for fingerprint classification[64] tasks requires a network of multiple SVMs in order to distinguish more than two classes[65]. Multiple types of SVMs are used for these networks. The One-vs-All SVM is used to make a decision if the test vector belongs to one concrete class or any of the other possible classes. With the pairwise SVM, a decision is made between two concrete classes or two disjunct subsets of classes. Combinations of multiple SVMs of these types can be used to design an Error-Correcting Code (ECC) system to lower classification errors[66]. It is further possible to apply an explicit rejection rule for feature vectors depending on the minimum distance.

The use of SVM for fingerprint classification was introduced by Yao [65]. In this approach the FINGERCODE is used as feature for classification as described in [67]. Three combinations of the SVMs were tested on NIST-SD4 to classify into five classes. Using five One-vs-All SVMs, 88% CCR was achieved while rejecting 1.8% of the fingerprints. The combination of ten pairwise SVMs can improve CCR slightly to 88.4%. Best results were obtained by using a combination of both systems and fusing them to an ECC Scheme. The CCR for this scheme is 89.3%.

Min [68, 69] also described the use of FINGERCODE and SVM as classifier. 5 One-vs-All SVMs are used (one for each class) to create so called Decision Templates for each class, using training data. The Decision Templates of each class are then clustered with the help of a SVM to obtain one Decision Profile per class. For classification, the distance of the Decision Profiles to the Decision Template of the probe is computed. The algorithm was tested on the NIST-SD4 and gave a CCR of 90.4% (5 classes) and 94.9% (4 classes). It is notable, that the first impression of fingerprints were used for training and the second for testing, which might result in overfitting.
An alternative approach was given by [70]. While he also uses the FINCERCODE as feature for classification, he also extracted singular point and ridge structure information. The additional information is used within a naïve Bayes classifier to calculate the weights for the output of 5 One-vs-All SVMs. This allows dynamic ordering based on the input fingerprint and improved classification performance. For five classes, a CCR of 90.8% was achieved on the NIST-SD4. In case of ambiguous class labels, all of them were considered correct.

In [71] a lot of preprocessing is applied to the image, containing a reject option for low quality images. Filters are used on the image in order to find candidates for singular points (including the probability to be SP). Responses of complex filters and the information extracted from the candidate singular points are used as features. A predefined SVM System is used for classification. The best performance observed on the NIST-SD4 is 93.5% for five and 95% for four classes.

Nearest Neighbor

The Nearest Neighbor or K-Nearest Neighbor classifier [72] uses the entire training set for its class decision. During the classification, it compares the feature vector of the probe to all the feature vectors of the training data and determines the k training features close to the probe. The different approaches based on k-Nearest Neighbor (kNN) classifier defer in the way, they calculate the distances between vectors and how the class decision is done after obtaining the neighbors.

In Fritz work [73], a feature is used to classify fingerprints with the Nearest Neighbor (NN) method. During the preprocessing, many enhancements are applied to the image and afterwards skeletonization is done. The feature vector is received by using FFT on the image and employ the defined 'wedge-ring' detector, which aggregates FFT responses in predefined areas. The NN algorithm is used to find the nearest reference vector for a given sample. On a small test set consisting of 40 images, a CCR of 85% was achieved for classifying arches, loops and whorls.

Another approach was made by Wang [74], who used directional classification together with kNN. In a first preprocessing step, the image quality is enhanced. After the calculation of the OM, the core point is searched using Poincarémethod. Around the core point, 16x16 blocks of the OM are extracted and their mean is calculated. A k-Means is used for creating the clusters and 3 kNNs are used for classification. On a reduced set of the NIST-SD14 89.4% of the fingerprint were correctly classified into four classes. The CCR on their created database (1200 images) is 91.5%.

Rajanna et al. [75] tested and compared multiple feature extraction algorithms with kNN as classifiers. They used OMs, a generalized representation of it called Orientation Collinearity Maps (OCs), Gabor filter responses and Minutia Maps. Since Minutia Maps depend on level 2 features and results are worst of the tested features, they are not further mentioned here. All tests were performed on the NIST-SD4 to solve the 5 class problem. The CCR using OM was the best (85.53%) compared to the other single classifiers (Gabor with 83.6% and OC with 76.76.5%). Only the combined output of the classifiers using OM an OC could outperform the best single classifier slightly (85.85%).

Luo et al.[76] take up the idea of kNN and use features for the classification. First step of their algorithm is a preprocessing to reduce noise, as well as the detection of a reference point. A Curvelet Transformation is used to create gray-level co-occurrence matrices, from which the feature vector for classification is derived. With help of 10-Nearest Neighbor classifier, a CCR of 94.6% for five classes and 96.8% for four classes was obtained on the NIST-SD4.

Classifier Comparison

Nyongesa et al. [77] did a benchmark of different neural network approaches, namely MLP, RBF and Fuzzy-Neural Network (FNN). They used 7 features derived from the OM and singular points like their relative position and orientations. Results of classifying into five classes were reported for the NIST-SD4. The best CCR of 92.55% was achieved using MLP. CCR for RBF and FNN were 88.85% and 90.75% respectively.

In [78], Kristensen et al. compare different machine learning classifiers using the FINGERCODE as a feature. They use the core of the fingerprint as reference point to extract the feature vector for classification. SVMs as well neural networks based on MLP, Bidirectional Associative Memory (BAM), Hopfield and Kohonen were tested on a small database (512 images) with natural class distribution. Only the MLP and SVM approaches produced noteworthy CCRs of 88.8% and 87% respectively, without rejecting fingerprints.

3.3.4. Graph Matching & Structural Approaches

Maio [79] introduced a new type of approaches by segmenting the OM into areas containing similar slopes. The interconnection of the center points of these areas form a graph that is then further analyzed. Thereby he makes the classification problem to graph matching problem. By defining an arbitrary number of prototype/model graphs for each class that needs to be distinguished this method is very flexible. The classification is done by the calculation of distances between the probe and model graph. No tests on databases were performed to show classification performance.

In the approach of Senior [80] fiducial lines are used on an enhanced and skeletonized image. The classification features are derived by analyzing the intersection of the fiducial lines with the ridgelines and taking some measurements. By using a two dimensional Hidden Markov Model (HMM) up to 90% CCR were achieved on subset of NIST-SD4.

Cappelli et al. [81] defined a graph based classification algorithm, called MASKS. This algorithm utilizes segmentation of the directional image similar to [79] but has some changes in the algorithms that do the work. A genetic algorithm is used to improve the segmentation and to be more robust to local ridge direction changes. Predefined dynamic masks are used to do a cost calculation for the transformation of the template mask to the predefined ones. The lowest costs indicate the class of the fingerprint. Since this approach was primary designed for continuous classification, results for exclusive classification classifying into five classes are lower than other approaches. On the NIST-SD14, 85.7% CCR were achieved and 87.1% on the NIST-SD4.

The proposed method of Jain and Minut [82] uses 'Kernel Fitting' for classification. The OM derived from a fingerprint image is analyzed using predefined kernel functions e.g. unique circle for the loop class. The classifier tries to fit the kernel functions to the OM and calculates scores on which the classification is done. In their test, they were able to achieve a CCR 91.25% without rejection on the NIST-SD4.

An improved feature extraction algorithm for graph matching is presented by Neuhaus and Bunke [83]. Within the OM directional variances are detected and used to create the graphs. Prototype graphs for the different classes are used to calculate the graph edit distance (GED) to the graph extracted from probe. The class is then determined by the lowest GED. The performance was tested on NIST-SD4 and CCR of 80.27% was achieved.

Jung and Lee[84] proposed an algorithm using Markov Model (MM) for classification. After enhancing and applying skeletonization, a ridgeline scanner is used to extract the directional variations of the ridges. The compressed variation information are used as classification features. A MM for each class is used is then used to classify input feature vector. The tests on the second half of a combined set of FVC2000-DB1 and FVC2002-DB1 shows a CCR of 80.1% for distinguishing four classes.

In Liu[85] an approach based on adaboost decision trees is shown. After OM calculation and image segmentation, the singular points are extracted using complex filters. Different measurements of the SPs are used to obtain 16 dimensional feature vector. The outputs of the decision tree are further normalized, so that they can be interpreted as class probabilities. Testing results on the NIST-SD4 show CCRs of 95.7% (4 classes) and 94.1% (5 classes).

3.3.5. Other Classification Approaches

In the approach of Chong [86], a geometric representation of the fingerprint is used for classification. Therefore, they group the ridgelines within the skeletonized image to receive the global geometric shape of the fingerprint. By analyzing its curvature they classify the sample into one of five classes. The algorithm was able to achieve good results (96.6% CCR) on a small database containing 89 images.

A Genetic Algorithm is used in Qi's approach [87] to improve the performance of the stated Probablistic Neural Network (PNN) by using a feedback mechanism. With respect to a registration point, the directions of the OM are used as feature vector for the PNN. With help of the genetic algorithm, the probabilities within the net are updated to improve performance. Tests on the NIST-SD14 give CCRs of 94% for five classes and 99% with 20% rejections. When decision is made between four classes, 94.4% CCR are achieved.

Hu and Xie[88] show a similar approach to [87]. They use genetic programming to optimize the distinctiveness of primitive features derived from the OM. After that, a BPN is used to classify input fingerprints and return the two most probable classes. If the output is reliable the class is directly determined. Otherwise, a SVM is used for the distinction of the two classes. They results on the FVC2004 DB1 and DB2 show good CCRs of 96.4% for distinguishing four classes at 7.2% rejection and 93.6% (with no rejection) as well as 96.2% (with 15% rejection) for five classes.

Park and Park [89] describe an approach using kernel discriminant analysis for classification. With the help of Discrete Fourier Transformation (DFT) and directional filters, the image is preprocessed, before FFT is used to retrieve the OM. The a core point is retrieved and directional vectors around the core point are used for the discriminant analysis and class derivation. The tests were performed on NIST-SD4. For the 4 class problem, the CCRs are 94% and 97.9% (at 20% rejection) and for five classes 90.7% and 95.3% (at 20% rejection).

The approach of Tan et al. [90] uses genetic programming for feature learning. The feature vector consists of the OM itself as well as additional information derived from it. After the learning algorithm finished, a Bayesian classifier is used to do class detection. The NIST-SD4 was used for testing and the classifier achieved 93.3% CCR for four and 91.6% for five classes.

Leung and Leung [91] make use of a simplified FINGERCODE as feature. They use block wise filter responses around the registration point as features that are reduced by applying Fisher's discriminant analysis for creating the test data. A Bayes classifier is used to classify the fingerprints based on the filter responses. The results given are not in the form to be comparable to the other results stated here.

Vitello et al. [92] presented a method based on k-Means and naïve Bayes classifier. The feature for classification is derived from the OM using fuzzy c-means algorithm. The classification is done by applying the c-means algorithm, using the probe and training set as input and use its output to decide class with naïve Bayes classifier. The approach was tested on a subset of PolyU Database (100 images) and received 91% CCR.

Jung and Lee[93] proposed a new algorithm based on statistical analysis of the ridge directions within a detected core block. The OM is derived by clustering the input image and applying FFT to the blocks. Then MM that is trained to detect core points is used to select the reference point for the center of a core block. Orientation histograms are created for 4 regions within the core block and so called local models are created. The classifier than calculates the probabilities of the local models to belong to one of four classes. Results on the databases FVC2000, FVC2002 and FVC2004 show CCR of 97.1%, 97.8% and 97.3% respectively.

3.3.6. Multi Classifier Approaches

Within this section, all approaches are mentioned that use a combination of multiple classifiers in order to improve the classification performance. It is tried to achieve the improvement by using (hopefully) uncorrelated features for different classifiers, so that they produce uncorrelated errors. With that, they can set up decision rules for combining the classifier[94] results like hierarchical/rule based or voting based (majority votes, weighted votes).

A direct successor of [22, 56] is the definition of Pattern-Level Classification Automation System (PCASYS)[95] as a reference system whose source code is freely-available. The system applies image enhancement in the first step, before extracting the OM. KL-Transformation is used to reduce the 1680 directional features to 64. In contrast to [22, 56], they use a PNN for learning the features. An additional line tracing algorithm is taken into account for detecting whorls with a high precision. By combining both, PNN and line tracer output, the system can achieve a CCR of 92.2% without rejection and 96.5% at 10% rejection rate on the NIST-SD14.

Based on the PCASYS approach Lumini [96] extended the idea to a continuous classification system. In contrast to PCASYS, the outcome of the algorithm is no an exclusive class but the ranking of the possibilities of belonging to the different classes. The database search is done incrementally by looking into the batches of different classes depending on the derived probabilities. This allows handling ambiguous fingerprints without misclassifying them. Test show that the database hit rate can be improved on NIST-SD4.

Within [67, 23], Jain et al. described a new representation for fingerprints using filter responses, named FINGERCODE. In a first step, a reference point needs to be found. When this reference point is found, it is set as registration point and the area around this point is divided into 48 sectors. Four different Gabor filters are applied to the point, so that a feature vector of 192 elements is obtained. They tested different classifier together with this feature vector on the NIST-SD4. At first a kNN was testes, which was able to classify 85.4% correctly into 5 and 91.5% into four classes. By applying rejection-based threshold for neighbors and rejecting 19.5% of the fingerprints, the CCRs could be improved to 93.5% and 96.6% respectively. When using a set of 10 pairwise SVMs and kNN for classification and combining their outputs using a multiplexer, 90% (five classes) and 94.8% (four classes) CCR were achieved. A trained neural network was able to classify 86.4% (five classes) and 92.1% (four classes) correctly.

The purpose of [97][98][99] is to combine the MASKS [81] and Multispace Karhunen-Loéve Transformation (MKL) based approach[100] for continuous and multiple MKL based classifiers for exclusive classification. The MKL method utilizes the OM and a registration point as feature input. With that, training data is partitioned regarding the fingerprint classes and subspaces of the feature vectors are created for each partition. New fingerprints are classified by the distance to these subspaces. Therefore, they named the method Subspace-based Pattern

Discrimination (SPD). Two types of classifiers were used in this approach. The MKL-MIN selects the class with the smallest distance and MKL-KNN by determining the nearest subspaces. By combining these classifiers using majority voting system, their test on the NIST-SD14 achieved 94.5% CCR without rejection and 99% when rejecting 17.5% of the fingerprints, with respect to the confidentiality of the distances. The combination with the MASKS was done for continuous classification, where a combined distance to the subspaces served as foundation of classification. The obtained class was searched for the test sample and search space was further extended, until either a match was found or the entire database was searched. In average 3.7% of the database were searched.

Cappelli et al.[101] took up their previous idea of MKL classifier [98] and built a two-stage classifier from that, together with a so called SPD classifier. The MKL is used in a first instance to find the two most likely classes of the probe fingerprint. Based on that, one of the multiple trained SPD classifiers is selected to make the final decision between the two classes. The SPD uses the OM as base for the feature vector and it is reduced by KL-Transformation. To differentiate between two classes, it was trained on the most discriminative features of the two classes. They were able to achieve 96.6% and 95% CCR for four classes and for five classes respectively without rejection on the NIST-SD4.

In Pattichis [102] a similar approach to PCASYS is used. It differs in the way of preprocessing and retrieving the directional information. Preprocessing is done using Amplitude-Modulation(AM) and Frequency-Modulation(FM) technique. The obtained image quality is better than the initial approach and the extraction of FM component gives the features for classification. Using this approach results in less misclassified fingerprints but unfortunately, no detailed results are given.

In [103][104] Senior and Bolle extend the approach using fiducial lines and HMMs [80] by fusing the results with an additional Decision Tree classifier. The features for the second classifier are extracted at salient points on the ridgelines e.g. turning point in the curve. By using this second classifier, it is supposed to include additional uncorrelated features to the classification process. Multiple test were performed on the NIST-SD4. The combination of the two defined classifiers achieved a CCR of 91% for the four-class problem when taking into account the class priors. When further merging the PCASYS [95] classification system, the CCR could be increased to 94.9%.

Marcialis et al.[105] describe a system consisting of a combination of Recursive Neural Network (RNN) output for relational graphs and MLP based on FINGERCODE[67]. The combination of the classifiers is done by using a kNN. Tests were performed on the NIST-SD4, which showed CCR of 87.88% for distinguishing five classes.

The approach of Yao et al.[106] bases on [105]. They propose a combination of RNN and SVM classifiers after comparing different classifiers and features. In this work they contrast the usage of FINGERCODE [67] and relational graphs [81] as features together with RNN, SVM and MLP as classifiers. They found that a combination of RNN extracted features from the relational graphs and FINGERCODE features together in an ECC SVM system produces the best results. The approach was tested on the NIST-SD4. At a rejection rate of 1.8%, CCRs of 94.7% (5 classes) and 90% (4 classes) were achieved. When allowing rejection of 20%, the CCRs will increase to 98.4% for four and 95.6% for five classes respectively. Further measurements for the single classifiers are provided in the paper.

The approach of Han [107] fuses the idea of machine learning with a set of rule. For the machine learning part, a reference point is searched by detecting singular points (using Poincaré) or maximum curvature. 100 blocks of the OM from around this point are then used to create a 300-dimensional feature vector by applying a statistical analysis. The rule-based system is based on the singularity information. When fusing both classifiers, they achieve 93.23% CCR for five-class problem without rejection.

Shah and Sastry[108] designed a line-detector for processing the input image and obtain the skeletonized fingerprint. They used the OM to create 2 feature vectors, one based on the directions around the center of the image and the other one with the direction around a detected center of the fingerprint. The classifiers kNN, Fast-Forward Neural Network (FFN) and SVM were compared and found to produce similar results. SVMs and FFN were set up in a hierarchical structure to separate arches from the loops and whorls and then distinguish four classes. The kNN classifier is improved by using the SVM vectors as prototypes. This decreases the complexity neighbor search. The different setups were tested on the NIST-SD14 and produced CCRs of 97.07% for the SVMs, 99.29% for kNN and 97.96% for the PNN. It should be denoted, that the documented results were achieved with overlapping training and test data.

The combined approach of Cao et al.[109] uses hierarchical combination of rule based, kNN and SVM classifier. In a first stage, complex filter responses for singular point detection are used to find arches and whorls. Secondly, a kNN is used to get the two most likely classes from a combined feature vector consisting of OM and filter response, reduced by principal component analysis. SVMs are then used to distinguish between those classes. For the loop and whorl classes, they further use ridgeline tracing for distinction. The NIST-SD4 was used for testing. CCRs of 95.9% for five-class and 97.2% for four-class problem were achieved without rejection.

Depending on the survey of Galar et al.[15], they selected a set of approaches([95, 30, 81, 67, 98, 28, 77, 108, 29, 70, 71, 85, 91, 110]) for testing within a defined framework[16]. They not only reimplemented the classification approaches from feature extraction to classifier but also used the feature extraction together with a bunch of reference classifiers, namely kNN, SVM and C4.5[111]. The evaluation of the approaches and different classifiers was done on multiple synthetic databases of different quality, created by SFinGe[112], as well as the NIST-SD4. Since the evaluation is very complex and many combinations of the previous named approaches and classifiers were tested, the original paper should be consulted for detailed information. In the following, only the best performing combination of the classifiers is described. It uses multiple SVMs with RBF kernel for each of the features from Hong 2008[70], Liu 2010[85], Zhang 2004[28] and Cappelli 2002[98]. The output of the four classifiers is combined after each of them presented its desired class. Either consensus vote or majority vote was used for combination. It has to be taken into account, that the consensus mode leads to a rejection if not all of the classifier outputs correspond and therefore a higher amount of fingerprints is rejected in favor of a better CCR. The tests on the different databases show good results for combination using consensus mode (98.45% to 99.5%) but also high rejection rates (ranging from 17.5% up to 43.3%) which could render the classifier combination not useful, depending on the database. Majority mode could improve the rejection rate (3.3% to 21.3%) but would also decrease the CCRs (93.4% to 98%).

Peralta et al.[113] propose a identification system, containing fingerprint classification subsystem based on the findings of [15]. For their system, they tested different combinations of feature extractors together with random forests and SVM as classifier. They came up with a two level classification system in order to prevent rejections. Depending on the first feature that is based on [70], the fingerprint might be rejected or not. If the fingerprint is not rejected, it will be classified using level 1 and level 2 if the feature extraction fails. The major difference of the two levels is the feature set used. Level 1 uses features described by Hong[70], Liu[85], Nyongesa[77] and Leung[91], whereas level 2 leaves out the Hong features. For both of the levels, a random forest is trained to find the most discriminative information for the final feature vector of reduced size. With that, the training data for the SVM classifiers could be extracted. Tests on the databases NIST-SD4, NIST-SD14 and a SFinGe created database (400,000 fingerprints) revealed CCRs of 92.97%, 93.76% and 94.38% respectively, without rejecting fingerprints.

3.4. Summary

After reviewing the above- mentioned works, it can be said that the variety of different approaches makes it hard to compare and rate them. This is due to multiple reasons:
The most important is that the datasets on which the tests were performed differ in size, quality and distribution. Even if the same databases are used, the different partitioning into test and training data affects the algorithms performance and comparability, especially for the machine learning approaches. Besides that,

some of the fingerprints within sensor captured databases can have ambiguities regarding their class if the ground truth is created by multiple human experts and the fingerprint does not clearly belong to one of the defined classes. The authors of the approaches use different strategies to deal with the ambiguous fingerprints like removing these images from database, accepting only one or multiple ground truth classes.
Another major point is the number of classes used for classification. It is obvious, that having an algorithm designed for distinguishing three classes will perform better compared to those classifying into five or more classes. Since for example tented arch and loops can have very similar feature distribution, its distinction is a more challenging task than selecting between whorl and arch. Similar to the findings in [15] it can be said that the authors of the papers offer information on their approaches at different levels of details. While some of them state each possible parameter for configuration, other only give a high-level description with a lack of details. This can lead to problems for reimplementing the stated approaches.

When contrasting the different types of classification approaches some basic observations can be made. Nearly all of the approaches somehow rely on the OM in the first step, except those using the raw or at most the preprocessed image. Preprocessing is another task in most of the proposed methods but it is not clear if papers without information on preprocessing completely skip this step or just not mentioned it. Apart from [25] no recent approach was made for using syntactic classification of fingerprint. Moayer showed that syntactic representation could serve for this purpose but a complex grammar needs to be defined for automatic classification. This complexity might be one of the main reasons why so few works base on syntactic classification to date.
The rule-based classification approaches mentioned in Section 3.3.2 show advantages towards any machine learning approach because they do not necessitate training data and after the feature extraction, a quick decision can be done based on a small set of rules. Since most of them somehow rely on the extraction of singular points from the input fingerprint, this is a crucial step and the classification performance can highly rely on its accuracy. Especially those methods using relative measures between the singular points will suffer from inaccurate feature

extraction.
The machine learning methods are capable of managing more complex feature vectors and making their decision based on those. While training of SVMs and classifying with them can be done very efficiently depending one the learning strategy, the training of neural networks can be a challenging task. Both approaches are fast in classification in contrast to kNN classifier, thats running time during classification is highly dependent on the number of prototypes because distance measurement with the probe has to be done for each prototype. The more prototypes are used, the better the classification performance will be but it results in worse classification speed.
The graph matching approaches from Section 3.3.4 tend to have lower classification accuracy than others. This is because they are primarily designed for continuous classification approaches instead of exclusive classification. The outcome is the similarity to the different classes, which is used to define a order in which database searches are performed.
The most promising works regarding the classification performance make use of the advantages of different classifiers and features by fusing them. However, the combination of different approaches requires a good fusion strategy and classifier selection. Hierarchically arranged classifiers offer opportunities for doing a stepwise classification based on the reliability of the classifiers output. Without the definition of hierarchy, the classifier outputs can be fused by applying different voting schemes like majority or consensus voting which will directly affect the CCR and rejection rates, as can be seen in [16].

Having in mind that the comparison of the approaches is difficult due to the before mentioned reasons and information on the different approaches might be incomplete, is difficult to select possible approaches as a basis for testing within this thesis. Except the PCASYS system of Candela et al.[95] none of the other approaches deliver any sources to their papers, what makes it necessary to have a reimplementation of the approaches. Fortunately Galar et al.[16] performed extensive testing of different feature extraction methods and classification approaches and therefore reimplemented some of the methods. They provided the source code together with their paper to enable others building and testing classification systems with no need for reimplementation. This work will serve as a basis for implementation of the classification system in chapter 5.

All reviewed approaches of the previous sections can be found within Table 3.1. The table states the best testing results as well as the used databases for each of the approaches. Further information is given regarding the class distribution, size of the database and the handling of ambiguous class labels. Due to the FBI requirements of at least 99% CCR at a maximum of 20% rejection rate[30], the presented results are selected with respect to that. If multiple rejection rates were reported, then results with the lowest (including 0%) and the first rejection rate, lower or equal to 20% are presented. The entries are ordered by year and contain information on the type of approaches, databases used as well as the achieved CCRs and corresponding rejection rates.

Table 3.1.: Different classification approaches and their best results on different databases

Ref.	Author	Year	Type	DB	Classes	CCR	Reject
[20]	Moayer	1975	SYN	other	7	92.5%	13%
[21]	Moayer	1976	SYN	-	-	-	-
[26]	Rao	1976	RUL	other	-	-	-
[24]	Rao	1980	SYN	other	7	91.66%	-
[27]	Kawagoe	1984	RUL	other	6	91.48%	-
[47]	Hughes	1991	ANN	other	-	-	-
[48]	Bowen	1992	ANN	other	4	93.6%	-
[22]	Wilson	1992	ANN	SD4	4	81% / 89.3%	- / 10%
[49]	Kamijo	1993	ANN	other	5	86%	-
[59]	Moscinska	1993	ANN	other	3	80%	-
[56]	Wilson	1994	ANN	SD4	5	90.2%	10%
[95]	Candela	1995	MUL	SD14	6	92.22% / 96.5%	- / 10%
[57]	Wilson	1995	ANN	SD4	5	92.2% / 96.57%	- / 10%
[73]	Fitz	1996	NN	other	3	85%	-
[60]	Halici	1996	ANN	SD4	5	81%	-
[30]	Karu	1996	RUL	SD4	4 / 5	93.9% / 91.3%	10%
			RUL	SD9	4 / 5	91.4% / 90.1%	10%
[79]	Maio	1996	STR	-	-	-	-
[31]	Ballan	1997	RUL	-	-	-	-
[86]	Chong	1997	OTH	other [s]	5	96.6%	-

Table 3.1.: Different classification approaches and their best results on different databases

Ref.	Author	Year	Type	DB	Classes	CCR	Reject
[50]	Geng	1997	ANN	SD4	2	98.2%	
[96]	Lumini	1997	MUL	SD4 ^{n}	-	continuous	-
[80]	Senior	1997	STR	SD4 ^{r}	4	90%	-
[87]	Qi	1998	-	SD14	4 / 5	94.8% / 94%	
			-	SD14	5	99%	20%
[58]	Sarbadhikari	1998	MLP	other ^{s}	5	up to 100% ^{o}	
[81]	Cappelli	1999	STR	SD4 ^{n}	5	87.1%	
			STR	SD14 ^{r}	5	85.7%	
[36]	Hong	1999	RUL	SD4	4 / 5	97.6% / 92.3%	20%
[67]	Jain	1999	KNN	SD4	4	91.5% / 96.6%	- / 19.5%
			KNN	SD4	5	85.4% / 93.5%	- / 19.5%
			ANN	SD4	4 / 5	92.1% / 86.4%	-
			MUL	SD4	4 / 5	94.8% / 90%	-
[97]	Cappelli	2000	MUL	SD14	5	94.4% / 99%	- / 17.5%
[38]	Cho	2000	RUL	other	4	92.3%	-
[61]	Bernard	2001	SOM	NIST •	4	88%	-
[105]	Marcialis	2001	ANN	SD4	5	87.88%	-
[51]	Nagaty	2001	ANN	other	4 / 5	99%/98.8% ^{b}	-
[102]	Pattichis	2001	MUL	SD4	-	-	-
[103]	Senior	2001	MUL	SD4	4	94.9%	-
[65]	Yao	2001	SVM	SD4	5	89.3%	-
[98]	Cappelli	2002	see Cappelli 2000 [97]				
[25]	Chang	2002	SYN	SD4	5 / 7	93.4% / 94.8%	5.1%
[82]	Jain	2002	STR	SD4	4	91.25% ^{b}	-
[52]	Mohamed	2002	ANN	SD4	5	98.5% ^{o}	-
[74]	Wang	2002	KNN	SD14 ^{r}	4	89.4%	-
			KNN	other	4	91.5%	
[101]	Cappelli	2003	MUL	SD4	4 / 5	96.3% / 95.2%	-
[106]	Yao	2003	MUL	SD4	4	94.7% / 98.4%	1.8% / 20%
			MUL	SD4	5	90% / 95.6%	1.8% / 20%
[99]	Cappelli	2004	see Cappelli 2000 [97]				
[39]	Dass	2004	RUL	SD4	4	94.4% *	-
[32]	Klimanee	2004	RUL	other	6	91.3%	4.5%
[77]	Nyongesa	2004	ANN	SD4	5	92.55%	-
[89]	Park	2004	-	SD4	4	94% / 97.9%	- / 20%
			-	SD4	5	90.7% / 95.3%	- / 20%

Table 3.1.: Different classification approaches and their best results on different databases

Ref.	Author	Year	Type	DB	Classes	CCR	Reject
[104]	Senior	2004	see Senior 2001 [103]				
[108]	Shah	2004	SVM	SD14 ^{r}	4	97.07% ^{o}	-
			KNN	SD14 ^{r}	4	99.29% ^{o}	-
			ANN	SD14 ^{r}	4	97.96% ^{o}	-
[40]	Wang	2004	RUL	SD4	5	82% / 94% ^{b}	- / 14.4%
[28]	Zhang	2004	RUL	SD4	4 / 5	92.7% / 84.3%	-
[107]	Han	2005	MUL	SD4	5	93.23%	-
[83]	Neuhaus	2005	STR	SD4	5	80.27%	-
[90]	Tan	2005	-	SD4	4 / 5	93.3% / 91.6%	-
[68]	Min	2006	SVM	SD4	4 / 5	94.9% / 90.4% ^{o}	
[78]	Kristensen	2007	SVM	Other $^{s\ n}$	4	87%	-
			MLP	Other $^{s\ n}$	4	88.8%	-
[29]	Wang	2007	RUL	SD4	5	88.6%	-
[42]	Fan	2008	RUL	SD4 $^{r\ s}$	3	97%	-
[70]	Hong	2008	SVM	SD4	4 / 5	94.9% / 90.8% ^{f}	-
[71]	Li	2008	SVM	SD4	4 / 5	95% / 93.5%	1.8%
[43]	Liu	2008	RUL	SD4	4 / 5	94.38% / 91.62%	1.55%
			RUL	SD14	5	89.15%	3.07%
[53]	Jin	2009	ANN	SD24	5	91.4%	-
[84]	Jung	2009	STR	FVC00/02 ^{s}	4	80.1%	-
[37]	Liu	2009	RUL	SD4	5	95.6%	-
[33]	Msiza	2009	RUL	FVC02-1	4 / 5	84.5% / 83.5%	-
[41]	Wang	2009	RUL	FVC02/04	5	96.96%	0.8%
[88]	Hu	2010	MUL	FVC04	4	96.4%	7.2%
			MUL	FVC04	5	93.6% / 96.2%	- / 15%
[85]	Liu	2010	STR	SD4	4 / 5	95.7% / 94.1%	-
[69]	Min	2010	see Min 2006 [68]				
[75]	Rajanna	2010	KNN	SD4	5	85.85%	-
[91]	Leung	2011	-	FVC00/02 ^{s}	-	-	-
[54]	Thepade	2012	ANN	other ^{s}	5	80%	
[109]	Cao	2013	MUL	SD4	4 / 5	97.2% / 95.9% ^{f}	-
[35]	Guo	2014	RUL	FVC00/02/04	4	92.74%	-
[76]	Luo	2014	KNN	SD4	4 / 5	96.8% / 94.6%	-
[92]	Vitello	2014	-	PolyU ^{s}	4	91%	-
[34]	Webb	2014	RUL	FVC02-1	5	91.1%	-
			RUL	FVC04-1	5	91.8%	-

Table 3.1.: Different classification approaches and their best results on different databases

Ref.	Author	Year	Type	DB	Classes	CCR	Reject
[44]	Dorasamy	2015	RUL	SD4	5	80.51%	12%
			RUL	FVC02-1	5	90.11%	-
			RUL	FVC04-1	5	88.98%	-
[93]	Jung	2015	-	FVC00	4	91.1%	-
			-	FVC02	4	97.8%	-
			-	FVC04	4	97.3%	-
[45]	Chua	2016	RUL	SD4	4 / 5	88.3% / 92.13%	-
[55]	Wang	2016	ANN	SD4	4	91.4% / 93.1%	-
[62]	Borra	2017	ANN	FVC2000	-	97.56%	-
[113]	Peralta	2017	MUL	SD4	5	92.97%	-
			MUL	SD14	5	93.76%	-
			MUL	SFinGe	5	94.38%	-

• no details on the database
^{n} natural distribution
^{f} for ambiguous labels, only first label assumed correct
^{b} for ambiguous labels, both labels assumed correct
^{o} over-fitting due to overlapping training and test data
^{s} small test set
^{r} reduced database set

4. Data Analysis

Before setting up a fingerprint classification system based on the previously mentioned features and methods, we need to determine if there could be any possible performance gain for using multiple finger's classes for database binning. Within this chapter, the analysis of the fingerprint databases regarding the distribution of fingerprint classes as well as the correlation of classes between the fingerprints of a hand is described.

4.1. Datasets

The following section will outline the process of database selection for the planned analysis. Therefore, the requirements on the fingerprint databases will be outlined and the available databases are evaluated against these requirements.

4.1.1. Requirements

In order to have a proper selection of the data used for the planned analysis, some basic requirements were defined. On the one hand, the requirements offer the possibility to compare the available databases and they also serve as arguments for or against their usage within the analysis.

The following requirements were defined:

Ground Truth for fingerprint classes
: The most important requirement on the data is the presence of ground truth information on the fingerprint classes in each image. Absence of this information renders the dataset useless for the planned analysis since no statistics can be produced.

Natural Class Distribution

A further important point is the class distribution within the database. The data should represent an extract from a real world database, without any pre-selection or filtering of the data regarding the fingerprint classes. If this is not the case, any statistics calculate from the data is useless since it does not represent natural data.

Number of subjects

In order to receive statistically significant results, the underlying dataset for the collection of the statistical data should have a large number of subjects.

Multiple adjacent fingers

To be able to analyze the correlation of fingerprint classes between multiple fingers of a person, the dataset should include fingerprints of more than one finger per hand.

Both hands

For the comparison of statistics from right and left hand, fingerprints of both hands of a subject should be included in the dataset.

Image Quality

The images of the database should be of a good quality for later evaluation purposes. This means that the ridgelines are clearly visible, since the feature extraction algorithms of proposed system in the next chapter should be able to work with the images from the dataset.

4.1.2. Dataset Selection

The related work study as well as some additional research led to a list of candidate fingerprint databases, often used for scientific research. An overview of the databases can be found in Table 4.1. It has to be mentioned that during the creation of this work, the publicly available databases provided by National Institute for Standards and Technology (NIST) have been withdrawn and are no longer available to download.

Even though most of the database looked very promising with regard to the num-

Table 4.1.: Candidate fingerprint databases for statistics calculation

Database	Images	Subjects	Fingers	Impres.	Res.	Format	DPI	Quality	Distrib.	Cls.
CASIA FPV5	20000	500	8	5	328x356	bmp	-	plain	-	-
NIST 4	4000	2000	1(mixed)	2	512x512	png	500	scanned/ rolled ink	equal	5
NIST 9	54000	2700	10	2	832x768	png	500	scanned/ rolled ink	natural	5
NIST 14			same as SD9			wsq		see SD9		
NIST 10	5520	552	10	1	832x768	png	500	scanned/ rolled ink	selected classes	5
MCYT	79200	330	10	12	256x400 300x300	bmp	500	plain	-	-
SFinGe 1	1500000	10000	10	15	416x560	png	500	plain	natural	5
SFinGe 2	1500000	10000	10	15	416x560	png	500	plain	equal	5
NIST 29	4320	216	10	2		png	500	rolled/ plain	-	-
NIST 30			see SD29				1000	see SD29		

ber of subjects or contained fingers, they lack in providing the required ground truth for the fingerprint classes. These were namely the CASIA FPV5 [114], MCYT [115] as well as both NIST databases SD29 and SD30. Therefore none of them was suitable for the context of this thesis. Further, the author refrained from the usage of the NIST SD10 database. The problem with this database is the pre-selection of subjects that own specific fingerprint classes, since it was created as a supplementary database for the NIST SD9.

The databases named SFinGe consist of algorithmically generated fingerprints using the 'Synthetic Fingerprint Generator' of the University of Bologna [112]. It is obvious that artificial fingerprints are not suitable for a statistical analysis because their class distribution is either equal for all classes or based a given statistical distribution. These databases might serve for later evaluation within this work.

The NIST SD4 database contains one fingerprint of 2000 subjects in 2 impression. As the database was designed to have equal class distribution to five finger classes it is not suitable for a statistical analysis but since it offers preprocessed/cropped images, it can be used for training a machine learning classifier within the proposed system. The equal class distribution allows us to train a classifier without any bias towards some class.

Only the NIST databases SD9 and SD14, which are the same datasets with different image formats (png and wsq), remain out of the determined list. The database consists of the scans from 5400 ten-print cards of the United States' FBI. 2 cards from 2700 subjects provide 27,000 different fingerprints, without any known pre-selection. Its class labels were assigned by examines. That makes it the only suitable database for analysis. For the statistic calculation, some preparation had to be done towards to following remarks:

- Since the database contains scans of two ten-print cards, only the first cards were used for statistics. The second card would be redundant information.
- Few of the fingerprints are classified as Scratches (S), which means they were not able to assign it to one of the valid classes. Therefore, the fingerprints of all hands containing such a S class were removed from the statistics.

  removed right hands of subjects: 172, 297, 328, 966, 1238, 1479, 1805, 1839, 2591

  removed left hands of subjects: 141, 212, 475, 832, 1053, 1253, 1849, 1881, 2525, 2612

Within the scope of an upcoming project of the da/sec Group at HDA, the 'Federal Criminal Police Office' of Germany, one of the project partners, kindly provided an excerpt of information taken from their fingerprint database. At this point, the author wants to point out that none of the received data contains any sensitive or person related information.
The received data consist of the class labels from all ten fingerprints of the subjects within their database. It has to be mentioned, that in contrast to the NIST SD9 database only 4 fingerprint classes are distinguished. Both of the Arch classes (Arch and Tented Arch) were combined into one class. As the classes were automatically determined by algorithm, they further provided multiple class labels with decreasing probability for the case that the detected class is ambiguous. For our analysis, only the first class label for each finger was used in order to get comparable results to the findings of the SD9 database. The presented tables base on

the class labels of ca. 26,000 subjects.

The results will be presented within the subsequent sections.

4.2. Class Distributions

The subsequent Tables 4.2 and 4.3 provide the information on class distribution among the NIST SD9 database and the BKA data respectively. The tables show a relative measurement of each class among all instances of the specified finger. The values for hand and all fingers represent the combined probabilities of related fingers. For a more visual comparison of values within the two before mentioned tables, compare Table 4.4. It consists of a heatmap, which contrasts their values with different color encodings for higher and lower values. This allows a fast capturing of the major differences.

Table 4.2.: Fingerprint Class Distribution for SD9

	A	L	R	T	W
right thumb	3.49%	0.71%	**48.94%**	0.22%	46.64%
right index	5.61%	14.72%	**39.43%**	7.06%	33.18%
right middle	4.76%	1.30%	**69.49%**	2.94%	21.52%
right ring	1.19%	1.41%	**49.61%**	1.19%	46.60%
right small	0.93%	0.19%	**79.41%**	0.82%	18.65%
right hand	3.20%	3.66%	**57.38%**	2.45%	33.32%
left thumb	5.50%	**53.31%**	0.93%	0.48%	39.78%
left index	5.84%	**37.73%**	15.17%	9.63%	31.64%
left middle	5.61%	**67.36%**	1.49%	5.06%	20.48%
left ring	1.90%	**58.66%**	0.48%	1.67%	37.29%
left small	1.26%	**83.94%**	0.22%	1.08%	13.49%
left hand	4.02%	**60.20%**	3.66%	3.58%	28.54%
all fingers	3.61%	31.93%	30.52%	3.01%	30.93%

The first obvious observation, when looking on Table 4.2 is that right loop (30.52%), left loop (31.93%) and whorls (30.93%) are the most common classes for the SD9 dataset, where as the both arch classes -arch (3.61%) and tented arch (3.01%)- have lower occurrence. Similar results can be found for the left loop

(31.11%) and whorl (31.22%) of the BKA data with the difference that the missing tented arch classes leads to higher values for right loop (35.77%) and less arches (1.91%).

Arches The two arch classes have by far the lowest probability among all classes (∼2-3.5% each), both in BKA and SD9 data. Table 4.2 shows only few occurrences of tented arches within thumb, ring and middle finger of both hands and higher values for index and middle finger, whereas the arch has few occurrences only in ring and small fingers. The total distribution of the arch class is lower within the BKA data but ratios between fingers are similar.

Whorl Class When comparing the class distributions for whorl class in both datasets, one can see that for each finger, very similar probabilities exist with a maximum difference of ∼3% for the index fingers. The combined probabilities for the left hand, right hand and all fingers differ for only 0.2%-0.6%.

Loop Classes The comparison of the loop classes creates a different impression regarding the similarity of the datasets in terms of class distribution. As Table 4.2 show, the probabilities for right classes are much higher for fingers of the right hand than they are for the left hand. Left loops have only low occurrence in right hands (<2%), except for the right index finger(∼15%). The opposite behavior can be observed for fingers of the left hand. When comparing this finding to Table 4.3, it can be seen that the distributions of the loops are somehow swapped. The communication with the distributor revealed, that its very likely an issue of the system which makes incorrect use of the class labels. In this case, the class distributions would match to those presented in Table 4.2 with minor differences for the right loop (∼3%) and

Table 4.3.: Fingerprint Class Distribution for BKA data

	A	L	R	W
right thumb	1.49%	**49.02%**	0.57%	48.91%
right index	4.23%	**36.62%**	22.41%	36.74%
right middle	2.17%	**74.24%**	2.66%	20.92%
right ring	0.65%	**50.84%**	1.56%	46.95%
right small	0.38%	**83.47%**	0.56%	15.59%
right hand	1.79%	**58.84%**	5.55%	33.82%
left thumb	2.60%	0.42%	**57.75%**	39.23%
left index	3.60%	16.35%	**45.48%**	34.57%
left middle	2.66%	1.70%	**74.01%**	21.63%
left ring	0.81%	0.62%	**62.00%**	36.57%
left small	0.47%	0.19%	**88.08%**	11.27%
left hand	2.03%	3.86%	**65.46%**	28.65%
all fingers	1.91%	31.11%	35.77%	31.22%

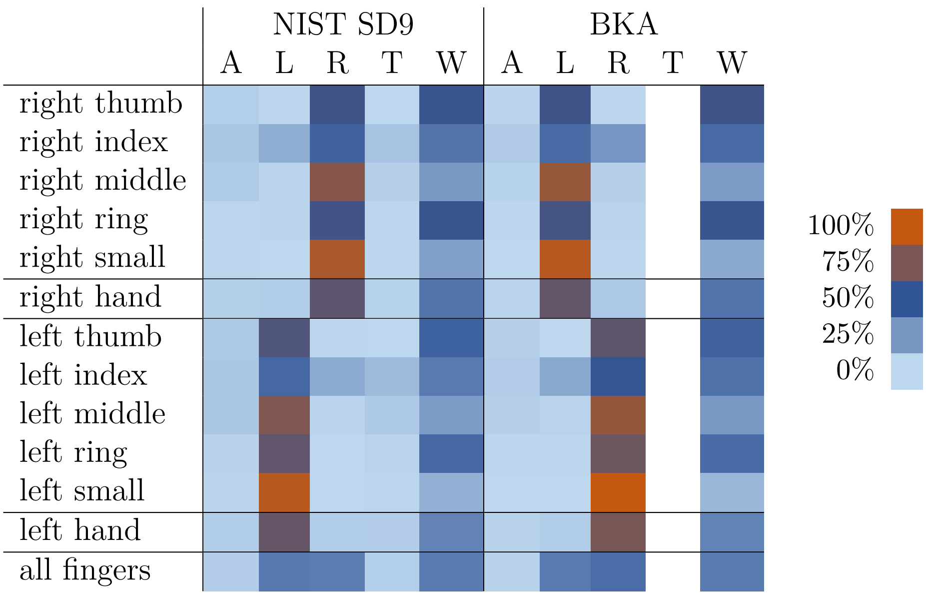
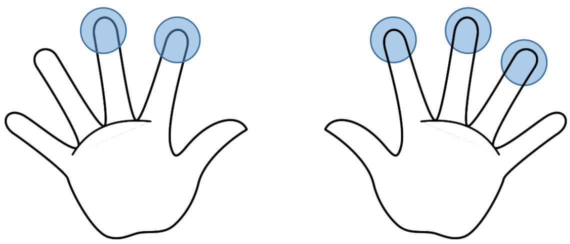


Table 4.4.: Heat Map for Class Distribution of SD9 and BKA data

4.3. Correlation Analysis and Workload Reduction

This section provides the statistics of the correlation analysis of the NIST SD9 and presents the possible workload reductions based on that. For this analysis, we had a look on the different combinations of class labels within the different subjects' hands and created statistics on its occurrences. Each table describes the occurrences of combinations for a defined number of adjacent fingers within either right or left hand. Grouping of the table is done by the used adjacent fingers. For example, Table 4.6 shows the correlation of two fingers classes on the right hand. The first probabilities were calculated for the right thumb and index finger, whereas the second column of probabilities describes right index and middle finger correlation. The first combination is always starting with the class of thumb together with its adjacent finger(s). Compare Figure 4.1 for a visualization of adjacent fingers.



a b

Figure 4.1.: The above figures show the adjacent fingers, right index and middle (a) and left index, middle and ring finger (b)

The size of tables was reduces for the sake of clarity, so that only the top ten class combinations are included within the tables of this section[1]. All tables shown here, base on the information of the SD9 dataset and its 5 fingerprint classes. Further tables were created by fusing its arch classes and by an analysis on the basis of provided BKA data. These secondly mentioned tables can be found in the appendix B and C of this thesis, since they show similar distributions as the following tables do.

[1]The complete tables will be provided on a CD together with this thesis

4.3.1. Possible Workload Reduction

For a naïve approach, the average number of comparisons required to identify a probe within a closed set scenario can be described by

$$m * \frac{n * (n+1)}{2}$$

, where n is the number of enrolled subject and m denotes the number of fingerprints to be compared for identification. The function fits if we stop comparison as soon as a comparison scores above a given threshold is retrieved. As an example, we take the database SD9. If we define one potential finger that should be utilized to identify the subject within the database (e.g. right index), then all the references of all enrolled subjects will be compared subsequently and at some time a match is found and the search can be stopped. Since the closed set scenario considers all attempting probes to be enrolled to the database [12], the stopping point will occur as soon as the index at which the subject is saved within the database is reached. If we search for each subject within the database, those indices will be $1, 2, 3, ..., n$. For the SD9 database and searching for one finger, the formula looks as follows:

$$1 * \frac{2700 * (2700+1)}{2} = 3,646,350$$

This would be the average number of comparisons done to search all 2700 subjects for a given fingerprint.

The classification system approach, defined in section 5.3, allows an extension of this formula by taking into account the binning of the database. Instead of searching the entire database for each attempt, we only search within the database bins, ordered by their probability regarding classification outcome. For determination of the maximum possible workload reduction, we assume a perfect classification and matching algorithm. That means, after the classification step, we will always search in the correct bin. And due to the perfect matcher, we can stop the search as soon as we found the subject within that bin.

Therefore, we can take the previous formula to define the average search times within each bin and sum up all these search times. Note, that now n_i defines the size of the i-th bin and N the number of bins.

$$\sum_{i=1}^{N}(m * \frac{n_i * (n_i + 1)}{2})$$

The following example will show the possible workload reduction for the case that the right index finger will be used for comparison and classification is done using right index and middle finger. The size of bins for the SD9 database regarding this two fingers is given in Table 4.5[2]. If we use the formula above, we get

AA	66	LA	13	RA	26	TA	21	WA	2
AL	0	LL	22	RL	5	TL	3	WL	5
AR	65	LR	278	RR	946	TR	153	WR	430
AT	17	LT	17	RT	27	TT	11	WT	7
AW	3	LW	66	RW	58	TW	3	WW	449

Table 4.5.: Bins for right index and middle finger on SD9

the number of 702,216 comparisons. In relation to the naïve approach, these are ~80% less comparisons. Since these values require perfect classification they do not represent actual workload reduction, but it can be used to have a lower bound for comparison with later tests using real classifiers.

The described method can be used to analyze the possible workload reductions for all combination of adjacent fingers for the SD9. In the following subsections, the correlation information is stated together with an analysis of possible workload reduction using the 5 classes. For the calculations, the above formulas are used. The naïve approach as a reference and the sum over the average workloads for searching within the bins as

[2]Note that the values wont sum up to 2700 since 7 finger combinations contained fingers labeled as scars. Within a real system, those could be handled by an additional bin for others.

4.3.2. Workload Reduction for 2 Fingers

If we have a look on Table 4.6 and Table 4.7, we can see that the correlation of right loops and whorls is very high for right hands. The same applies for left loops and whorls for the left hand. Other class combinations especially those containing one of the arch classes, left loops in right hands or right loops in left hands, are very rare.

Table 4.6.: Two Finger Correlations for right hand fingers on SD9

thumb index		index middle		middle ring		ring small	
WW	26.53%	RR	35.12%	RR	41.66%	RR	46.93%
RR	25.94%	WW	16.69%	RW	25.75%	WR	29.73%
WR	11.82%	WR	15.98%	WW	19.47%	WW	16.76%
RL	8.21%	LR	10.33%	AR	2.90%	RW	1.67%
RW	6.43%	TR	5.65%	TR	2.30%	TR	1.11%
WL	6.02%	LW	2.45%	WR	2.04%	LR	1.04%
RT	5.13%	AA	2.45%	RL	1.15%	AR	0.59%
RA	3.23%	AR	2.42%	AA	0.93%	RT	0.52%
WT	1.64%	RW	2.16%	RT	0.74%	AA	0.41%
AA	1.45%	RT	1.00%	LR	0.71%	RA	0.37%
rest	3.60%	rest	5.76%	rest	2.34%	rest	0.85%

As the tables show, the distribution of subjects among the different bins is very uneven, what has an impact to the possible workload reduction achieved if the respective fingers are used for binning. Table 4.8 states the possible workload reductions for the combinations of fingers as described in the above tables. The calculation was done using the extended formula from subsection 4.3.1 for the SD9 database. When looking on the results for possible workload reduction, one can find that the lowest workload is achieved using thumb and index finger of either of the hands, while the worst reduction is achieved for ring and small finger.

If we compare those results to both correlation tables, then we notice that those finger combinations showing a high correlation between their finger classes, can achieve less workload reduction than those with lower correlation (thumb and index, index and middle finger).

Table 4.7.: Two Finger Correlations for left hand fingers on SD9

thumb index		index middle		middle ring		ring small	
LL	24.05%	LL	32.42%	LL	46.51%	LL	55.58%
WW	21.56%	WW	16.17%	LW	19.41%	WL	25.69%
WL	11.34%	WL	14.91%	WW	17.10%	WW	11.38%
LR	9.59%	RL	10.71%	TL	4.13%	LW	2.04%
LW	9.48%	TL	7.21%	AL	3.42%	TL	1.49%
LT	7.10%	AA	2.94%	WL	3.38%	AL	0.93%
WR	4.91%	LW	2.64%	AA	1.38%	AA	0.71%
LA	3.09%	AL	2.12%	RL	1.23%	LT	0.48%
AA	2.04%	RW	1.45%	LT	0.71%	LA	0.45%
AL	1.93%	RT	1.45%	AT	0.56%	AT	0.26%
rest	4.91%	rest	7.99%	rest	2.19%	rest	1.00%

Table 4.8.: Possible Workload Reduction for 2 Fingers

first finger	Comparisons	% of naïve
left thumb	528,524	**15%**
left index	629,451	17%
left middle	1,044,008	29%
left ring	1415,647	39%
right thumb	619,573	**17%**
right index	702,216	19%
right middle	1,022,995	28%
right ring	1,230,814	34%

4.3.3. Workload Reduction for 3 Fingers

Similar to the findings for combinations of two fingerprint classes, we can see that also for three fingers, some classes seem to strongly correlate. For both hands, these classes are the whorls as well as the appropriate loop (right loop for right hand and left loop for left hand). The distributions shown in Table 4.9 and Table 4.10 reveal that the average size of the bins decrease, while there are still some very probable class combinations.

Table 4.9.: Three Finger Correlations for right hand fingers on SD9

thumb, index middle		index, middle ring		middle, ring small	
RRR	23.78%	RRR	24.19%	RRR	39.69%
WWW	15.01%	WWW	15.46%	RWR	18.06%
WWR	11.11%	WRW	10.89%	WWR	10.78%
WRR	10.00%	RRW	9.92%	WWW	8.70%
RLR	6.61%	LRR	6.54%	RWW	7.69%
RWR	4.72%	WRR	5.02%	ARR	2.42%
RTR	4.27%	TRR	4.46%	TRR	2.19%
WLR	3.42%	LRW	3.49%	WRR	1.97%
WLW	2.04%	LWW	2.04%	RRW	1.52%
RWW	1.60%	RWW	1.82%	RLR	0.82%
rest	17.43%	rest	16.16%	rest	6.17%

The results for the workload reductions in Table 4.11 show that as the average bin sizes are getting smaller and more bins of similar size exist, the possible workload reduction increases. This is the case for the two first finger combinations, while the combination of middle, ring and small finger leads to less workload reduction, due to the three major bins holding around 75% of the subjects.

Table 4.10.: Three Finger Correlations for left hand fingers on SD9

thumb, index middle		index, middle ring		middle, ring small	
LLL	21.26%	LLL	23.35%	LLL	44.24%
WWW	12.08%	WWW	14.13%	LWL	14.28%
WLL	9.37%	RLL	8.85%	WWL	10.97%
WWL	9.11%	LLW	8.62%	WWW	6.13%
LRL	7.10%	WLW	8.40%	LWW	4.94%
LWL	5.43%	WLL	6.51%	TLL	3.98%
LTL	5.32%	TLL	5.99%	WLL	3.20%
LWW	3.87%	LWW	2.12%	ALL	3.01%
WRL	3.20%	WWL	2.04%	LLW	1.71%
ALL	1.52%	ALL	1.82%	RLL	1.15%
rest	21.75%	rest	18.18%	rest	6.39%

Table 4.11.: Possible Workload Reduction for 3 Fingers

first finger	Comparisons	% of naïve
left thumb	335,227	9%
left index	389,312	11%
left middle	865,634	24%
right thumb	411,156	11%
right index	421,217	12%
right middle	793,815	22%

4.3.4. Workload Reduction 4 and 5 Fingers

The analysis for the combination of 4 and 5 fingers for database binning shows, that the trend towards lower average bin sizes when using additional fingers continues. The distributions in Table 4.12 and Table 4.13 show that less bins of big size exist, while the majority of bins have similar, smaller size. Again the increasing number of bins of lower size correlate to the possible workload reductions (see Table 4.14).

Table 4.12.: Four Finger Correlations for right hand fingers on SD9

thumb, index middle, ring		index, middle ring, small	
RRRR	18.99%	RRRR	23.19%
WWWW	14.12%	WWWR	8.18%
WWRW	8.10%	WWWW	7.28%
WRRW	5.46%	RRWR	6.91%
RLRR	5.02%	WRWR	6.76%
WRRR	4.31%	LRRR	6.21%
RRRW	4.12%	WRRR	4.50%
RTRR	3.53%	TRRR	4.38%
WWRR	3.01%	WRWW	4.12%
RWRW	2.71%	LRWR	3.08%
rest	30.62%	rest	25.38%

Although the size of bins is further decreasing when using 5 fingers (tables 4.15 and 4.16), the impact on the workload reduction is less strong. The number of possible bins is 5 times the number of bins for 4 fingers but only 1-2% of additional workload reduction can be achieved (compare Table 4.17). This shows that a higher number of fingers used for binning not necessarily leads to lower workload.

Table 4.13.: Four Finger Correlations for left hand fingers on SD9

thumb, index middle, ring		index, middle ring, small	
LLLL	16.51%	LLLL	22.08%
WWWW	10.86%	WWWL	8.85%
LRLL	6.32%	RLLL	8.70%
WWLW	5.87%	LLWL	6.39%
WLLL	5.61%	WLLL	6.10%
LTLL	4.65%	WLWL	5.91%
LLLW	4.54%	TLLL	5.76%
WLLW	3.64%	WWWW	5.28%
WWLL	3.23%	WLWW	2.45%
LWWW	3.12%	LLWW	2.16%
rest	35.65%	rest	26.32%

Table 4.14.: Possible Workload Reduction for 4 Fingers

first finger	Comparisons	% of naïve
left thumb	220,893	6%
left index	308,007	8%
right thumb	280,130	8%
right index	319,559	9%

Table 4.15.: Five Finger Correlations for right hand fingers on SD9

thumb, index, middle ring, small	
RRRRR	18.43%
WWWWR	7.28%
WWWWW	6.84%
WWRWR	4.76%
RLRRR	4.76%
WRRRR	3.86%
RTRRR	3.53%
RRRWR	3.42%
WWRWW	3.34%
WRRWR	3.23%
rest	40.54%

Table 4.16.: Five Finger Correlations for left hand fingers on SD9

thumb, index, middle ring, small	
LLLLL	16.06%
WWWWL	6.51%
LRLLL	6.25%
WLLLL	4.87%
LTLLL	4.42%
WWWWW	4.35%
WWLWL	3.90%
LLLWL	3.61%
WWLLL	2.94%
LWLLL	2.94%
rest	44.16%

Table 4.17.: Possible Workload Reduction for 5 Fingers

first finger	Comparisons	% of naïve
left thumb	177,018	5%
right thumb	211,996	6%

4.4. Summary

Within this chapter, we have seen the correlation statistics of fingerprint classes among adjacent fingers of left and right hands. Starting with the selection of the datasets in section 4.1, an analysis of the class distribution was created for NIST SD9 as well as the data provided by BKA in section 4.2. The following remarks can be made regarding the class distributions of SD9:

- The class distribution for all fingers in SD9 lays at around 31% for whorls and both loop classes while the arch classes have lower probability of ∼3% each.
- The right hands contain way more right loops (57.38%) than left loops (3.66%), while the opposite is true for left hands (3.66% right loops and 60.2% left loops)
- Because of the uneven class distribution among right and left hands, separate statistics have to be created if they should be utilized for classification purposes.

Except from the swapped loop class distribution on the BKA data, which is very likely an issue of the classification system, the distributions support the findings of SD9 database. Consequently, this would allow generalizing the findings also for other databases.

The presented statistics in section 4.3 allow some conclusions for the following chapters, where this information is going to be used for proposing a new classification system with classification error correction:

- The probabilities of different class combination allow a correction of poor classifier outputs for single fingers.
- For fingers, where classification fails completely, the most probable class can be determined with help of the statistics.
- The ranked lists allow finding the most promising setup for multi-instance classification. The binning of the database works best if the bins are of similar size.

- By using the formula described in section 4.3, we can calculate the possible workload reductions for different setups using different combinations of adjacent fingers. An overview on the possible workload reductions using 5 classes was given within the previous subsections. For a compact overview as well as results using only 4 classes, the tables of Appendix A can be consulted.

- We can see a connection between the distribution of the bins and the possible workload reductions. If we have few bins holding the majority of subjects, the reduction is lower than for those distributions where the majority of subjects is spread over more bins.

In the previous section we have seen, how we can measure the possible workload reduction for multi instance fingerprint classification.

5. Proposed Classification System

Within this chapter, a classification system is proposed, that utilizes the statistic information to reduce the required comparisons for identification tasks. This includes the used features, the classifiers and the developed binning algorithm based on multiple fingerprint classes. A coarse grained overview of the system is depicted in Figure 5.1.

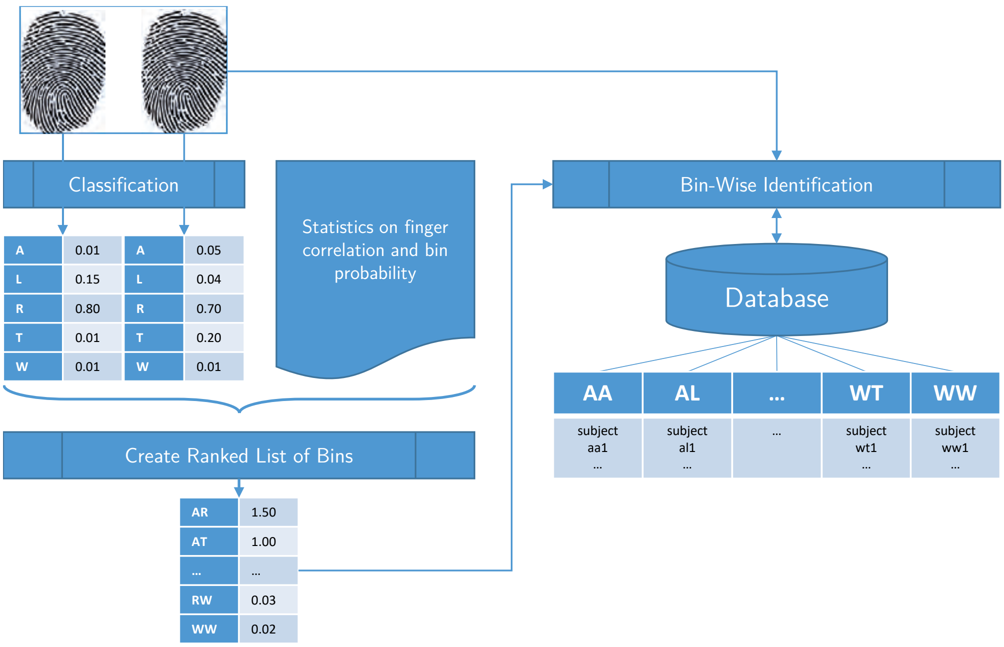


Figure 5.1.: Overview on the Proposed System

5.1. Classification Features

Following section describes the features proposed for testing purposes and states why they were selected. Instead of reimplementing the feature extraction methods from scratch, we made use of the software created in the context of the survey done by Galar et al. [16]. They did a reference implementation for some of the feature extraction methods proposed in chapter 3, which also includes the Fingercode and the PCASYS approach.

5.1.1. FingerCode

The features proposed for this system is the Fingercode, which was described first by Jain et al. [23] for identification purposes and also for classification [67], in a coarse grained variant. Other classification approaches as [70] are based on this feature in combination with varying classifiers (compare chapter 3). The basic workflow of the feature extraction is depicted in Figure 5.2.

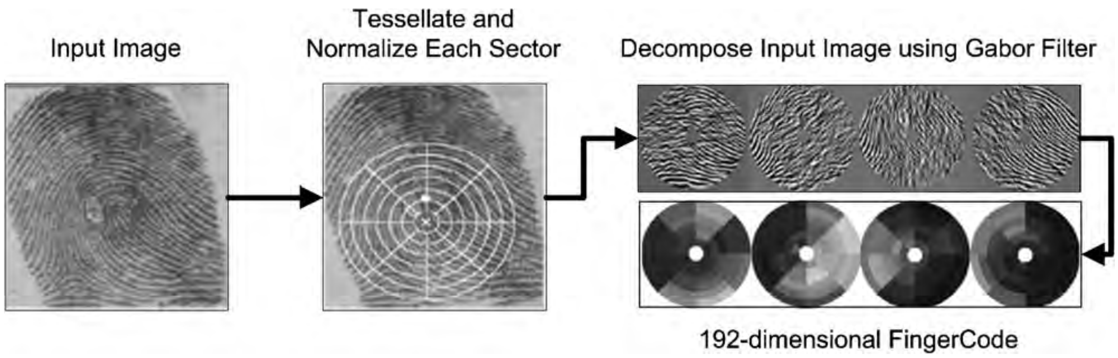


Figure 5.2.: Extracting the Fingercode feature from a fingerprint image [70]

The key idea behind the algorithm is to find a global representation of the ridge curvature around a determined registration by applying Gabor filter[116] of different directions to the area around the registration point. To find a registration point within the image, at a first stage the orientation map is calculated to find the point of highest curvature changes, the core point. This point then serves as the center for circular area to which the Gabor filters for four directions (0deg, 45°, 90°, 135°) are applied. The area is divided into 48 sectors for which the standard deviations of gray values is then calculated.

The final feature vector includes the standard deviations of all sectors and for all gabor filter responses, which makes a total of 192 features.

The reason for the selection of this feature was that the applied filtering and the use of the gray value variance instead of gray value distribution promised a good operation even on databases with lower image quality, which is the case for the SD9 database. A probable drawback of this method is its rejection rule. In case that the registration point cannot be set properly, the respective fingerprint will be rejected and no features will be extracted.

5.1.2. PCASYS Feature - Reduced Orientation Features

As an additional feature for testing, the approach of Candela et al. [95] was used. Within their work, they also used a neural network for classification based on this feature, as it is intended for this work. The actual feature is calculated after the segmentation and enhancement of the input image. A ridge-valley detector is used to determine the orientation map of the image and a registration point is calculated on this basis. The point is located approximately at the position of the core point and is used to rearrange the orientation image. This process is depicted within Figure 5.3.

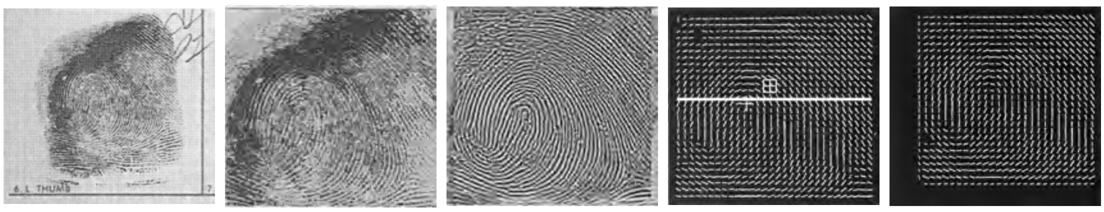


image segmented enhanced registered arranged

Figure 5.3.: Process of feature extraction for PCASYS feature [95]

After determination of the final registered orientation image, a feature vector consisting of 1680 elements representing the directions is created out of it. With the help of Principal Component Analysis, this feature vector can be reduced to a size of 128 elements. This is the actual feature uses during the classification. Since there is no rejection of images during the feature extraction, this feature this feature can be seen as a counterpart towards the Fingercode feature.

5.2. Neural Network Classifier

Neural Networks are a family of machine learning algorithms that are gaining more and more attention in a growing field of applications. Some approaches regarding fingerprint classification have been proposed with a neuronal network as classifier while achieving good classification results (compare Table 3.1). Therefore we decided to train neural networks for the tests in this thesis.

A neural network consists of multiple layers of interconnected nodes, the neurons. The basic layers of such a network are the input layer and output layer. The number of nodes within the input layer corresponds to the number of elements of the feature vector, while the output layer contains the nodes for all class labels. All nodes of these two layers are interconnected and during the training of the network using classified data, it weights the connections between the nodes in order to make a correct class prediction. Often so-called hidden layers are established between the input and output layer. They can be used to separate those two layers and reduce the number of neurons directly affecting the output by cumulating the outputs of the input layer.

Setting up such a network is no easy task, since it depends on a variety of parameters that influence the learning capabilities of the network. Therefore preliminary test were performed to find suitable combinations of the parameters and network architectures. The concrete parameters are described in the test setup.

5.3. Multi-Instance Binning Algorithm

In the following section, one of the core elements of this thesis is described. The binning algorithm to create an ordered lists of database bins based on the classifier output for multiple fingerprints. The aim of this algorithm is to enable an identification system to reduce the number of references to which a captured probe has to be compared and therefore reduce the systems workload.

5.3.1. Retrieval Strategies for Database Bins

According to Maltoni [2, Chapter 5], there exist three major strategies, when the class of a fingerprint was determined. The strategies describe how the information on the class label should be utilized for traversing the database.

Class Exclusive The idea is to use the obtained class label and do an exclusive search among the database bin for this class label. As soon as a matching reference is found, the search can be stopped. Depending on the system configuration, it might be desirable to extend the search to the rest of the database. Without extension to the entire database, this method would require a perfect classifier, which is not very likely.

Fixed Search Order In this approach, predefined orders are used to traverse a database regarding the fingerprint class. The obtained class label is used to select one of the fix traversal orders that were defined for the system. The main intention is to follow a search order that takes the confusion between fingerprint classes into account. Especially structural similarities between classes can be used to previously define fixed search orders.

Variable Search Order This traversal technique can be used if the classifier outputs the probabilities for each class. The search order can then be changed with respect to that probabilities and search can be stopped if the appropriate fingerprint was found.

5.3.2. Binning Algorithm

The retrieval strategy within this work follows the approach of an variable search order, since the bin probabilities are predicted for each request individually. It would be unfavorable to use the bin probabilities provided in chapter 4 for establishing a fixed order for searches. If we think of a combination of class labels that has a low overall probability regarding the statistics than the application of fixed search rule will force the algorithm to search within all bins having a higher statistical probability, even if the classifier generated a certain output for the class labels.

The aim is to combine the class predictions of the fingers in such a way, that we retrieve a list of all database bins in a ranked order, with the most probable bin in the beginning. A further intention is to overcome the problem of uncertain classifier outputs by using the statistical information on fingerprint distribution and bin probabilities to correct these uncertainties and gain values that are more probable. An overview on the algorithm can be found in Algorithm 1.

The algorithm consists of two major steps:

1. Calculating all possible combinations of bins while preserving the single class probabilities

2. Checking the probabilities regarding the statistics and do adjustment

Combine Classifier Outputs The first step can be done easily by creating the Cartesian product of the set of class predictions. This will create all the different bins including the corresponding class probabilities (see Figure 5.4).

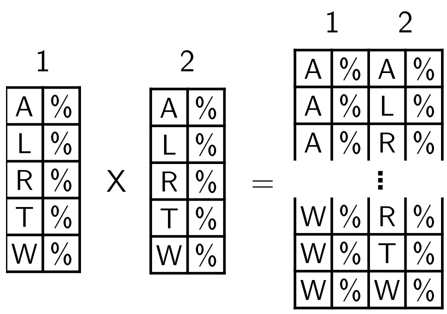


Figure 5.4.: Cartesian product for two sets of class predictions

Adjust Probabilities To adjust a class probability within the bin, the statistic is called to retrieve the class probability under the assumption that the labels of the other fingers are as is. We can further set up a threshold that establishes an upper bound for the adjustment of the probabilities. That means for instance, if the probabilities for the classes, created by the classifier are A-95%, T-4%, R-0.5%, L-0.4%, W-0.1% and we set our threshold = 0.5%, then the adjustment will only applied to the two classes with a predicted probability below the threshold. This is done to honor the results of the classifier and prevent a completely override of predicted values. In case that the class prediction failed and no probabilities are present, this method would adjust all class probabilities for the finger as they lay below the threshold.

Apply Correction Since the adjustment of the class label could result in high probabilities depending on the statistics for the specific bin, some kind of justification could be done in order prefer high class probabilities obtained by the classifier over those calculated from the statistics. An example would be the case that the classifier correctly extracted the class of the first finger but only low probabilities for the second finger were achieved. Then the adjusted probabilities of the second finger could overwhelm those of the first one and statistics are weighted higher than classification outputs.

Therefore, we can think of different values for adjusting the determined probability from statistics.

0 fixed scale to 50%

1 highest prediction probability within current finger

2 defined threshold for upper bound

3 1 - value of [2]

4 class a priori probability from statistics for the given finger

5 highest prediction probability within all given fingers

6 1 / number of fingers used for binning

Algorithm 1: Determine Ranked List of Bins

```
Input: Statistics, threshold, ClassificationResults
Result: Ranked List of Bins
bins = createCartesianProduct(ClassificationResults);
ranked_list = ();
foreach bin in bins do
    foreach classprediction in bin do
        if classprediction.probability < threshold then
            classprediction.probability = Statistics.ClassProbabilityForBin();
            classprediction.probability *= maxPredictionProbability;
        end
    end
    ranked_list.add(getBinName, calculateSumOfProbabilities(bin));
end
sortDescending(ranked_list);
return ranked_list;
```

5.4. Test Setup

This section outlines the test setup, which includes the data used for training and testing the classifier, information on the parameters used for training the classifiers as well description of different tests that were performed.

5.4.1. Selected Datasets

All available databases that fit in the context of this work were described before, in section 4.1. Out of those databases, only the SD4 and SD9 from NIST were chosen for the test provision. Example images can be found in Figure 5.5.

NIST SD4 The database was selected because the included images are already cropped and only few images have greater amounts of background information left. The majority of the images is of good quality and since the class distribution is equal towards all five classes, it is well suited for classifier training.

NIST SD9 The SD9 database is the only large scale database, containing real fingerprint data for which a ground truth of class labels exist. Further, there is no

SD4 512*512 SD9 832*768 SD4 832*768

Figure 5.5.: Example images from the used databases

preselection of class types so that a natural class distribution is expected. In terms of the workload reduction tests, only the first impressions of the finger prints were used, since having a subject twice enrolled, would not be appropriate.

5.4.2. Processing Chain

Resizing of SD9 images In order to use the proposed reference implementation for the feature extraction algorithms by Galar et al. [16], it was necessary to resize the images to at max 800 pixels of width, since the software fail at loading the images.

Feature Extraction As stated in section 5.1, Fingercode and PCASYS features were selected. The extraction was done with the mentioned software provided by [16]

Fingerprint Classification The classification was done on basis of the extracted features, using self trained neural networks. The networks were implemented in Python using the Keras framework for the Tensorflow machine learning library from Google.

Database Binning In order to test the possible workload reduction for the proposed system on the SD9 database, the subject identifier were put into lists representing the database bins. These bins were created for different starting fingers and number of adjacent fingers.

5.4.3. Feature Extraction Errors

As stated in the section about the Fingercode feature, the extraction algorithm provides a rejection rule for fingerprints that fail at detecting a reference point. This is the case for around 5191 images, which is a proportion of nearly 10% of all images (54,000). Since possible workload reduction is expected to highly demand on the classifier performance, it might be interesting to preserve a classifier based on this feature for comparison towards the classifier based on non rejecting feature extraction.

5.4.4. Experiments

Determine Baseline Performance for trained Classifiers Before conducting the actual tests for workload reduction, the baseline performance of the classifiers is tested for both 4 classes and 5 classes.

Tests different approaches for probability correction As described in section 5.3, multiple approaches for the correction of the probability adjustments were proposed. In this test, it is intended to compare the effect of these measurements towards workload reduction under different thresholds for the upper bound.

Test Possible Workload Reduction The last test combines the findings of the previous ones in order measure possible workload reduction when applying proposed classifiers and most promising probability correction techniques. Therefore, multiple classification attempts are done for different combinations of adjacent fingers from left and right hand.

6. System Evaluation

Within this chapter the evaluation of the proposed system of chapter 5 is done. The observed results are described and a discussion is done afterward.

6.1. Metrics

In the following, some metrics are presented, that help us to describe the performance of the classifiers on the one hand and the possible workload reduction for identification scenarios on the other hand.

6.1.1. Confusion Matrix

The confusion matrix is used to show an accumulated result for classification tests. For each possible class, there is one column and on row within the matrix for each class label. The rows denote the actual classes of the tested fingerprint, while the values in the columns show how often fingerprint of this class have been confused to the class of the column label. If the fingerprint is classified correctly, the value of the corresponding row/column for the class is increased, otherwise the value in the column to which it was confused.

6.1.2. Correct Classification Rate

The correct classification rate is the measurement of accuracy for a classifier. It can be calculated with the help of the confusion matrix. After running all classification tests, we can calculate the sum of each row, which is the number of fingerprints tested, that contain the class of the row label. If we divide the value of the row/column for the actual class by the calculated row sum, we will receive the correct classification rate for the given class of the row label [2].

6.2. Classifier Baseline Performance

Before we were able to run the tests towards workload reduction, some classifiers had to be trained and tested. Therefore, different aspects were taken into account to have a diverse selection of classifiers. This includes the feature used by the classifier, the database it was trained on as well as the number of classes, the classifier has to distinguish. Especially the training database is an interesting point here because one might be interested to train classifier on a smaller database, containing images of higher quality and later use that classifier for another database. In the following, the different classifiers and their performance measurements are presented.

The baseline performance of the trained classifiers was measured against a set of test data after the training phase. During the 10-fold cross-validation, the database was randomly separated into training and testing set for ten times. This was done using a defined random seed for each split, which is the same for all classifiers. Then the classifier was trained and tested for all combinations and the confusion matrices for the tests were created. With that, an average measure of the classifiers accuracy can be given, by calculating the confidence interval [12] at confidence value of 95%. The interval is calculated for the single class prediction accuracies among all ten confusion matrices for one classifier, which enables a finer view on the per class accuracy of the classifier. Its outcome is the mean accuracy over all tests, as well as the mean-interval and mean+interval.

6.2.1. Fingercode Feature Classifier 1 - Trained/Tested on SD4

The first trained classifier used Fingercode as a feature. It was trained and tested on the SD4 database for the four class problem (combined arches). Table 6.1 shows the confidence interval for the ten training and testing attempts. One can see that the average CCR for the different classes lays between 88% and 90% while the upper and lower bound of the confidence interval state, that the classification rate is relatively stable among the tests. One has to note, that the feature extractor has rejected 724 fingerprints (18.1% of all images), for that, no classification can be done. Those are also not part of training and testing. To measure the certainty

Table 6.1.: Confidence Interval - Classifier 1

Class	Mean	Lower Bound	Upper Bound
A	88.23 %	86.41 %	90.04 %
L	88.83 %	87.84 %	89.82 %
R	87.66 %	86.09 %	89.22 %
W	89.93 %	88.66 %	91.19 %

of the classifier for a database that is different to the training database, the resulting classifier was then tested on all fingerprints of the SD9 database and the corresponding confusion matrix as well as CCRs were calculated (see Table 6.2). We can see that the correct classification rates drop for all classes for at least 15%. Further, the feature extractor rejected 6763 fingerprints (12.5%) of the SD9 database, which might be problematic for workload reduction tests, besides low CCRs.

Table 6.2.: Confusion Matrix - Classifier 1: Test on SD9

	A	L	R	W	Accuracy	
A	**837**	248	308	290	49.73%	
L	978	**9395**	1248	3767	61.05%	
R	793	725	**9884**	3352	66.99%	
W	191	1516	2012	**11693**	75.87%	
Rejected						6763(12.5%)

Because of the high rejection rate for fingerprints of the SD9 database (see Table 6.2) as well as the low CCRs, the classifier was later used in comparison to another classifier, having better CCRs.

6.2.2. Pcasys Feature Classifier 2 - Trained/Tested on SD4

The subsequent classifiers including the one described below are all based on the Pcasys feature, which has no rejection rule and therefore allows to classify all tested fingerprints.

The classifier described here was trained and tested on the SD4 database. The intention is to show the possible CCRs that the classifier is able to achieve in contrast to the following classifiers tested on the SD4.
As we can see in Table 6.3, the selected feature seems to be well suited for classification. The CCRs are between 86% and 90% and the confidence interval states, that the values are quiet stable among multiple tests. In contrast to the Classifier 1, the accuracy represents the actual accuracy of the classifier, since no fingerprints were rejected during feature extraction.

Table 6.3.: Confidence Interval - Classifier 2

Class	Mean	Lower Bound	Upper Bound
A	90.37%	88.00%	92.73%
L	88.32%	86.68%	89.97%
R	88.90%	87.69%	90.11%
W	86.60%	85.34%	87.86%

6.2.3. Pcasys Feature Classifier 3 - Trained/Tested on SD9

Classifier 3 was trained and tested on the SD9 database. The database split for each test was 20% of the fingerprints for training and 80% for testing. The confidence interval in Table 6.4 reveals that the classification accuracy of loops (90%-91%) and whorls (∼87%) is quite accurate and stable among the training and testing attempts. It is striking, that the CCR for arches is much lower than those of the other classes and also the interval boundaries deviate more strongly. Which raises the question why this is the case. The most obvious assumption is, that the natural distribution of the fingerprint classes (compare section 4.2) entails that only few arches can be randomly selected for the training sets and the classifier becomes biased towards the recognition of the other classes.

Table 6.4.: Confidence Interval - Classifier 3

Class	Mean	Lower Bound	Upper Bound
A	63.50%	61.02%	65.98%
L	90.95%	90.11%	91.78%
R	90.19%	89.50%	90.88%
W	86.73%	85.69%	87.78%

6.2.4. Pcasys Feature Classifier 4 - Trained on SD4/Tested on SD9

Similar to the Classifier 1 using the Fingercode features, the intention of this analyze the impact of training the classifier on the SD4 database, which has equal class distribution and do the testing on SD9. In contrast to Classifier 1, we now evaluate the classification performance using the Pcasys feature. Another intention is to evaluate the impact of unbiased training data towards the classification accuracy, since Classifier 3 seems to suffer from unequal class distribution.

Table 6.5 shows the confidence interval created for this classifier. We can see, that in comparison to Classifier 2 (trained/tested on SD4), all CCRs except those for the whorls are lower and also the confidence interval is wider. When comparing it to Classifier 3, we can see that not only whorls have similar CCR but also the mean CCRs for arches are close together. This leads to two assumptions: 1. The feature seem to be stable for recognizing whorls even if databases for training and testing are different. 2. The recognition performance of the other classes is influenced largely by training data and probably also their quality.

Table 6.5.: Confidence Interval - Classifier 4

Class	Mean	Lower Bound	Upper Bound
A	59.39%	56.43%	62.35%
L	75.28%	73.30%	77.25%
R	62.03%	58.91%	65.16%
W	86.34%	85.44%	87.25%

Pcasys Feature Classifier 5 (5 classes) - Trained/Tested on SD9

After training some models for four classes, an attempt was made to train a neural network similar Classifier 3, while using the Pcasys feature.

The findings for this classifier tests (see Table 6.6) overlap somehow with those from Candela et al. [95]. They also used the Pcasys feature and a neural network for classification and stated low classification accuracy towards tented arches for tests in SD9/14. In contrast to their results, also the CCR for arches is extremely low, and the confidence interval shows that none of the training and testing attempts seem to have good CCRs for the arch classes, whereas accuracy for loops is slightly higher than for the other tested classifiers.

Table 6.6.: Confidence Interval - Classifier 5

Class	Mean	Lower Bound	Upper Bound
A	0.36%	0%	1.16%
L	93.10%	92.23%	93.98%
R	92.20%	91.53%	92.86%
T	0.03%	0%	0.09%
W	85.28%	84.38%	86.18%

6.2.5. Classifier Discussion

All the before mentioned tests seem to supply the theory, that the classifiers have better CCRs on the databases they were trained on, in contrast to varying testing databases. But as mentioned in the test setup, the image quality of the used databases for training and testing, was not consistently good, so that this could have had some impact to the quality of the extracted classification features. If we have a closer look to the Fingercode features, the relatively high rejection rates of fingerprints during feature extraction for SD4 (18.1%) and SD9 (12.5%) might support the supposed suboptimal image quality.

Another interesting observation are the very similar whorl CCRs among the classifiers 2-5. In any case, the values are constantly around 85%-86% even if the accuracies for the other classes diverge among the classifiers. Either the Pcasys features allows describe whorls independently from the dataset with a constant CCR or the findings are by chance, which seems less probable due to the 40 tests of the 4 classifiers.

Unfortunately, the expected better classification accuracy when training on the unbiased SD4 database was not given. The most probable reason for that seems to be the diverging image quality, so that features of same classes are not very comparable among databases of different image quality. Even the Fingercode feature was sensitive to that, although its usage of filter responses and gray value variances was expected to be more robust.

Reviewing the CCRs of the different trained classifiers leads to the conclusion, that the classifier with the best performance on the SD9 database, which is used for workload reduction tests, is Classifier 3. Therefore, this classifier will be used within various tests towards workload reduction. The classifier 1 with the high rejection rate and moderate accuracy will be used to test the different adjustment approaches for correcting the classifier output.

6.3. Workload Reduction

Within the following section, some results regarding the workload reduction for identification scenarios are presented. At first, we will enlarge upon the results for testing different approaches for adjusting class probabilities obtained from the statistics. This is followed by some examples contrasting the probable workload reduction while using classifier 1 and 3. For each tests, the bins of the database were created with respect to the class of the first used finger together with the class(es) of its adjacent finger(s).
The workload reduction is measured by comparing the number of visited subjects against the naïve approach mentioned in subsection 4.3.1. For workload reduction tests, an additional column was entered, which shows the maximum possible reduction for the respective setup, calculated in subsection 4.3.1.

6.3.1. Probability Correction

In section 5.3 different approaches for the correction of predicted class probabilities were proposed. The results in here will show the impact of those values on the number of subjects, that have to be visited, when we are searching for all the subjects in the database once. Four examples for these impacts are presented in Table 6.7 - created using Classifier 1- and Table 6.8 -created using Classifier 3-. The tables contain the evaluation for the index and middle finger of the right and left hand.

Table 6.7.: Influence of correction value on the number of visited subjects - Classifier 1

	visited batches	visited subjects	% of naive	correction value
right	8,215	2,206,864	60.5%	-1
index &	8,262	2,170,183	59.5%	0
middle	8,533	1,833,457	**50.3%**	1
	8,379	2,148,842	58.9%	2
	8,375	2,197,449	60.3%	3
	8,276	2,169,374	59.5%	4
	8,546	2,111,850	57.9%	5
	8,262	2,170,183	59.5%	6
left	8,977	2,188,406	60.0%	-1
index &	9,012	2,136,673	58.6%	0
middle	8,850	1,711,740	**46.9%**	1
	9,119	2,094,487	57.4%	2
	9,119	2,177,769	59.7%	3
	9,078	2,156,219	59.1%	4
	9,124	2,040,612	56.0%	5
	9,012	2,136,673	58.6%	6

The depicted data is sufficient to show that the correction of the adjusted class probabilities can have an influence (see bold values) to the number subjects that were visited during the search. Table 6.7 shows that using the highest prediction probability within a finger (correction value 1) to adjust the changed class probabilities, can lower the number of visited subjects by around ten percent over the unadjusted probabilities (value -1) for the right hand and up to 13% for the left hand. These values were obtained by using the Classifier 1 with a lot of rejections and moderate classification accuracy.

The same was done for Classifier 3, which has better classification accuracy and no rejection. In contrast to the findings of Classifier 1, no major improvements were achieved by an additional adjustment of the probabilities.

Table 6.8.: Influence of correction value on the number of visited subjects - Classifier 3

	visited batches	visited subjects	% of naive	correction value
right	4,003	1,038,096	28,5%	-1
index &	4,009	1,036,241	28.4%	0
middle	4,013	1,037,944	28.5%	1
	4,025	1,036,029	28.4%	2
	4,014	1,036,011	28.4%	3
	4,007	1,035,427	28.4%	4
	4,010	1,037,611	28.5%	5
	4,009	1,036,241	28.4%	6
left	4,202	1,064,290	29.2%	-1
index &	4,204	1,062,095	29.1%	0
middle	4,200	1,062,748	29.1%	1
	4,212	1,060,186	29.1%	2
	4,212	1,061,954	29.1%	3
	4,202	1,060,939	29.1%	4
	4,202	1,063,828	29.2%	5
	4,204	1,062,095	29.1%	6

6.3.2. Workload reduction - Classifier 3

For the workload reduction tests, the determined parameter from the section before was used for both, Classifier 1 and 3. The subsequent tables show the number of visited bins and subject when using classifier 1 within the multi-instance classification system. The workload reduction is denoted by giving the proportion of visited subjects for this test and the naïve search strategy.

Tables 6.9, 6.10, 6.11 and 6.12 show the results for using the classes of two, three, four and five fingers respectively.

We can see that for two fingers, the workload reduction for the combination of right hand fingers tends to be better than for those of the left hand, although the possible workload reduction suggests the opposite. We can reduce workload to at least 51.6% of naïve approach and at max to 25.5% using classifier 1. If we compare the achieved reduction to the possible reduction, we can see that we are able to come as close as around 11% or better to what is possible.

Table 6.9.: Workload Reduction for 2 Fingers - Classifier 3

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	3,973	943,363	25.9%	17.9%
right index	4,026	1,039,215	28.5%	20.2%
right middle	3,726	1,348,713	37.0%	28.3%
right ring	3,602	1,577,563	43.3%	33.8%
left thumb	4,060	931,553	25.5%	15.4%
left index	4,202	1,059,422	29.1%	18.3%
left middle	3,947	1,472,067	40.4%	29.4%
left ring	3,874	1,881,160	51.6%	39.1%

Using 3 adjacent fingers not only increases the theoretical workload reduction, but also the achieved reduction is higher than for two fingers. Table 6.10 states, that we can reach a workload of only 18.8% of the naïve and the reduction using right index, middle and ring finger differs only for around 7% from possible workload reduction.

Table 6.10.: Workload Reduction for 3 Fingers - Classifier 3

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	5,726	707,142	19.4%	12.1%
right index	5,772	718,312	19.7%	12.4%
right middle	4,962	1,133,061	31.1%	22.0%
left thumb	5,847	686,249	18.8%	10.0%
left index	6,358	781,817	21.4%	11.6%
left middle	5,646	1,332,235	36.5%	24.4%

Tables 6.11 and 6.12 show that taking additional fingers into account can not only reduce the theoretical workload further, but also the actual reduction can be improved. In comparison to the results for three fingers, we can improve for at maximum 5% to ca. 15% of the workload from the naive approach. The step towards five fingers is much smaller. The additional finger leads to only 2% less workload for both actual and possible.

Table 6.11.: Workload Reduction for 4 Fingers - Classifier 3

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	10,298	537,659	14.7%	8.5%
right index	9,955	613,202	16.8%	9.5%
left thumb	11,029	533,667	14.6%	7.0%
left index	11,371	705,933	19.4%	9.3%

Table 6.12.: Workload Reduction for 5 Fingers - Classifier 3

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	22,558	463,427	12.7%	6.6%
left thumb	25,420	493,455	13.5%	5.8%

A general observation of the presented tables is that also for real classifiers, an improvement of workload can be achieved. Since the classifier accuracy is far from perfect, none of the achieved workload reductions matches to the possible values. Nevertheless, additional fingers used for binning imply less visited subjects during the identification. But with each additional finger the differences in the received workload reductions decrease. The means using 3 fingers instead of 2 produces a higher reduction than going from 3 to 4 fingers.

6.3.3. Workload reduction - Classifier 1

After we have seen the workload reduction for the well working classifier, we want to have a look on what improvements regarding workload reduction can be achieved, if we only use a classifier of moderate accuracy. Therefore, in the following the results for classifier 1 are presented in the same manner as for classifier 3.

As we can see in the tables 6.13, 6.14, 6.15, 6.16, the worse classification accuracy has a big impact on the workload reduction compared. Because the correct bins for searching cannot be reliably determined, more subjects have to be visited before the matching subject was found. In comparison to the results obtained with classifier 3, the workload is at least 20% higher.

Table 6.13.: Workload Reduction for 2 Fingers - Classifier 1

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	8,363	1,685,599	46.2%	17.9%
right index	9,007	1,807,461	49.6%	20.2%
right middle	8,787	2,193,993	60.2%	28.3%
right ring	10,532	2,610,202	71.6%	33.8%
left thumb	8,128	1,512,571	41.5%	15.4%
left index	9,189	1,677,837	46.0%	18.3%
left middle	9,320	2,121,008	58.2%	29.4%
left ring	10,388	2,560,526	70.2%	39.1%

Comparing Table 6.13 and Table 6.14, we can see that the additional finger leads to lower workload. In comparison to classifier 1, the improvement in similar. This holds also when we use even more fingers. But none of the resulting workload reductions is comparable to those with the better classifier.

An interesting observation made for tables 6.15 and 6.16 is, that going from 4 to 5 fingers will not improve workload but instead it will be impaired slightly. Another interesting point is the maximum achieved reduction. It is 33.9% of the

Table 6.14.: Workload Reduction for 3 Fingers - Classifier 1

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	21,074	1,457,818	40.0%	12.1%
right index	24,760	1,666,411	45.7%	12.4%
right middle	27,012	2,146,190	58.9%	22.0%
left thumb	23,080	1,323,417	36.3%	10.0%
left index	26,527	1,534,187	42.1%	11.6%
left middle	29,405	2,070,302	56.8%	24.4%

workload for naive approach and achieved using 4 fingers. The better classifier reaches similar results with only 2 fingers. This findings show, that for worse classifiers, we will need more fingers to reach similar workload reductions. But at a certain point, when more fingers are used, the high rate of false classifications seems to make the prediction of the correct bins much harder, so that no further improvement can be achieved or like in this example the workload is higher for more fingers.

result in no more improvements

Table 6.15.: Workload Reduction for 4 Fingers - Classifier 1

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	65,878	1,376,783	37.8%	8.5%
right index	85,893	1,666,676	45.7%	9.5%
left thumb	75,494	1,237,612	33.9%	7.0%
left index	95,358	1,526,894	41.9%	9.3%

Table 6.16.: Workload Reduction for 5 Fingers - Classifier 1

first finger	visited batches	visited subjects	% of naive	best possible
right thumb	244,091	1,384,941	38.0%	6.6%
left thumb	285,293	1,243,026	34.1%	5.8%

6.3.4. Discussion

In the following, we will cover some discussion points, that arose during the project and that might be of interest, when thinking of a real world application.

Classifier Accuracy If we compare the results obtained in the previous section, we can see that the accuracy of a classifier has a huge impact on the possible workload reductions. While we achieved the best result for classifier 1 when using 4 fingers, similar workload reduction can be achieved for only 2 fingers, when using classifier 3. And also better results are possible depending on the used fingers.

Correction of statistically derived class probabilities In subsection 6.3.1, we saw that the use of a correction value for the class probabilities which have been set by the system based on statistical data can have a noticeable impact on the workload reduction. Since this was only the case, when the moderate Classifier 1 was used and not for the results of Classifier 2, one can conclude that a bad classifier can benefit from that correction, while a better classifier stays unaffected.

Applicability Multi-instance classification is capable of decreasing the number of subjects to be visited in identification scenarios, in the event that the search is stopped as soon as the matching algorithm finds a subject exceeding a given threshold. If the system follows another strategy, e.g. searching the subjects with the ten highest matching scores, then multi-instance classification would not bring any benefit, since in this case all subjects in the database have to be visited for the list creation.

Possible Modifications In identification systems that stop searching at a given threshold for matching scores, the number of fingerprints used for the matching could be increased in order to tighten security. If the database for example is constructed using the classes of two or three fingers and subjects are enrolled with four fingers of a hand starting with the index finger then it is possible to use all of the fingers for identification to get higher level of certainty that the subject where the algorithm stops searching is a correct match.

Used Fingers In real world scenarios, could think about the used fingers for multi-instance classification. Since the evaluation shows, that it might be a good idea to use the thumb and its adjacent fingers for database binning, this is maybe not the case from a practical point of view. Because the displacement of the thumb in contrast to the index finger is really high, the user of the system is either forced to present his fingers one by one to the system or he has to make unnatural posture in order to present his plain fingertips to the sensor. A better idea in this case would be the usage of a combination of the other fingers. Since they are positioned side-by-side, it would be more convenient to present those fingers but we might have to take a loss of workload reduction into account.

7. Conclusion and Future Work

As the introductory example of the large scale FBI biometric database shows, the impact of steadily growing databases sizes and increasing number of subjects enrolled to the biometric systems is very high in terms of the response times of those systems. Without a proper method to improve the number of comparisons during identification tasks, it is a problem for time critical request to be processed in time. Therefore, this thesis targets to contribute a method for workload reduction for fingerprint recognition systems.

One approach for reducing the number of comparisons within biometric fingerprint systems is the use of classification techniques, which allow the separation of fingerprints within the database into bins according to predefined classes. With respect to **RQ1** defined in chapter 1, chapter 3 presents a state of the art survey on classification approaches based on various features and classifiers. Since no common framework was used to test the approaches, they often differ in the databases used for testing, definition of testing and training data as well as the handling of ambiguous fingerprint classes. Besides the descriptions within the chapter, Table 3.1 was created to allow a quick overview on the specifics of each approach.

In chapter 4, an analysis regarding fingerprint class distribution was performed. In the context of **RQ2**, a statistical analysis was done for the NIST SD9, which turned out to be the only free available database of sufficient size, that fulfill all the given requirements from section 4.1. While class distribution among all fingers of both hands reveal a similar probability of right loops, left loops and whorls of around ∼30% as well as ∼3% for both of the arch classes, the obtained results show that the distribution of loops highly differs between right and left hand. This is important, as soon as statistics on class distributions and correlation should be used for classification systems because they need to respect those differences when operating on a specific hand.

To answer **RQ3**, database binning was done several times on the basis of different combinations of fingers and their class labels. An analysis of the possible workload reductions for those setups is given in section 4.3 of the Analysis chapter. It shows that the possible workload reduction depends not only on the number of fingers used for binning, but also the distribution of the subjects among that bins. Considering the usage of two fingers for binning, it can be shown that, if only few bins include the majority of subjects, the possible workload reduction is smaller ($< 70\%$) as for distributions where they are spread over higher number of bins (up to 83%).

Since the previously described workload reduction was calculated under the assumption of a perfect classification, chapter 5 presents a proposal for a classification system that utilizes multiple fingerprints for classification. The system is intended to enable multi-instance classification for any classifier and takes into account classification errors. In the context of **RQ4**, we trained some neural networks for classification, which showed different classification accuracies. This allowed us to analyze and compare different workload reductions based on those classifiers. The results of chapter 6 show, that for instance the usage of three fingers for database binning can lead to lower workload ($\sim 20\%$ of the naïve approach) when using a classifier with average classification accuracy. And even with a classifier of moderate accuracy, the workload could be reduced to 36% of the naïve approach.

The results show that fusing the classes of multiple adjacent fingers can be used to improve the workload for identification scenarios by applying database binning. Further, the presented findings and approaches can be utilized for some future work. This could target to further improvements for the binning algorithm used or testing on additional databases of better quality than the SD9.

Appendix

A. Tables: Workload Prediction assuming Perfect Classifier

The workload prediction was done by ignoring class combinations that contain the 'scar' class, which exists in the database to indicate fingerprints where no class label could be assigned. This was done because subject containing those combinations are stored in an 'others' bin that is searched after all regular bins were searched. Including those combination would have increased number of comparisons by 2700 per containing subject.

Table 1.: Workload Prediction for 4 Classes (arch and tended arch combined)

first finger	adjacent fingers	Comparisons	% of naïve
left thumb	2	508,879	17%
left index	2	607,601	22%
left middle	2	1,037,344	33%
left ring	2	1,414,683	41%
right thumb	2	608,952	20%
right index	2	689,414	23%
right middle	2	1,020,711	30%
right ring	2	1,230,209	35%
left thumb	3	322,634	12%
left index	3	374,146	14%
left middle	3	859,439	28%
right thumb	3	403,329	14%
right index	3	412,804	14%
right middle	3	791,697	24%
left thumb	4	211,428	8%
left index	4	293,982	11%
right thumb	4	274,673	9%
right index	4	311,502	11%
left thumb	5	168,391	6%
right thumb	5	206,635	7%

Table 2.: Workload Prediction for 5 Classes

first finger	adjacent fingers	Comparisons	% of naïve
left thumb	2	528,524	15%
left index	2	629,451	17%
left middle	2	1,044,008	29%
left ring	2	1415,647	39%
right thumb	2	619,573	17%
right index	2	702,216	19%
right middle	2	1,022,995	28%
right ring	2	1,230,814	34%
left thumb	3	335,227	9%
left index	3	389,312	11%
left middle	3	865,634	24%
right thumb	3	411,156	11%
right index	3	421,217	12%
right middle	3	793,815	22%
left thumb	4	220,893	6%
left index	4	308,007	8%
right thumb	4	280,130	8%
right index	4	319,559	9%
left thumb	5	177,018	5%
right thumb	5	211,996	6%

B. Correlation Tables BKA

Table 3.: Two Finger Correlations for right hand fingers on BKA data

thumb index		index middle		middle ring		ring small	
LL	26.31%	LL	36.66%	LL	46.58%	LL	49.70%
WW	25.35%	WL	19.14%	LW	26.00%	WL	30.93%
WL	12.33%	WW	17.04%	WW	18.74%	WW	14.69%
LR	11.57%	RL	14.89%	WL	1.81%	LW	1.47%
LW	11.11%	AL	3.01%	RL	1.79%	RL	1.13%
WR	7.14%	RW	2.07%	AL	1.76%	AL	0.54%
LA	2.64%	RR	1.62%	RW	1.02%	LR	0.40%
WA	0.99%	LW	1.53%	LR	0.94%	LA	0.37%
AL	0.88%	AA	1.28%	AA	0.54%	WR	0.28%
AA	0.71%	LR	1.02%	RR	0.31%	RR	0.20%
rest	0.96%	rest	1.76%	rest	0.51%	rest	0.28%

Table 4.: Two Finger Correlations for left hand fingers on BKA data

thumb index		index middle		middle ring		ring small	
RR	32.50%	RR	40.77%	RR	52.45%	RR	58.05%
WW	20.83%	WW	17.17%	RW	18.74%	WR	27.18%
RW	13.13%	WR	16.60%	WW	18.34%	WW	10.41%
WR	11.99%	LR	12.53%	WR	4.01%	RW	1.60%
RL	9.44%	RW	3.32%	AR	2.69%	LR	0.83%
WL	5.69%	AR	2.32%	LR	1.09%	AR	0.63%
RA	2.58%	LW	1.95%	RL	0.74%	RA	0.43%
AR	1.32%	AA	1.83%	AA	0.69%	AA	0.31%
AA	0.86%	RA	1.29%	LW	0.34%	LW	0.20%
WA	0.72%	LL	0.69%	RA	0.29%	RL	0.14%
rest	0.94%	rest	1.55%	rest	0.63%	rest	0.20%

Table 5.: Three Finger Correlations for right hand fingers on BKA data

thumb, index middle		index, middle ring		middle, ring small	
LLL	24.47%	LLL	25.43%	LLL	44.77%
WWW	14.09%	WWW	15.76%	LWL	19.05%
WLL	11.34%	WLW	11.14%	WWL	10.94%
WWL	11.00%	RLL	10.86%	WWW	7.77%
LRL	9.84%	LLW	10.63%	LWW	6.72%
LWL	8.00%	WLL	7.91%	WLL	1.67%
WRL	4.82%	RLW	3.74%	RLL	1.64%
LWW	2.89%	ALL	2.38%	ALL	1.62%
LAL	1.90%	RWW	1.70%	LLW	1.30%
WRW	1.36%	LWW	1.28%	RWL	0.79%
rest	10.29%	rest	9.16%	rest	3.71%

Table 6.: Three Finger Correlations for left hand fingers on BKA data

thumb, index middle		index, middle ring		middle, ring small	
RRR	29.64%	RRR	31.50%	RRR	50.82%
WWW	12.36%	WWW	14.54%	RWR	14.85%
WRR	9.99%	LRR	10.27%	WWR	11.87%
WWR	8.27%	RRW	8.67%	WWW	6.47%
RWR	8.13%	WRR	8.64%	RWW	3.83%
RLR	8.07%	WRW	7.87%	WRR	3.75%
RWW	4.81%	WWR	2.55%	ARR	2.46%
WLR	4.18%	RWW	2.40%	RRW	1.34%
WRW	1.66%	LRW	2.06%	LRR	1.03%
RRW	1.57%	ARR	2.03%	RLR	0.63%
rest	11.33%	rest	9.47%	rest	2.95%

Table 7.: Four Finger Correlations for right hand fingers on BKA data

thumb, index middle, ring		index, middle ring, small	
LLLL	18.34%	LLLL	24.35%
WWWW	13.21%	RLLL	10.69%
LRLL	7.80%	WWWL	8.70%
WWLW	6.92%	LLWL	7.97%
WLLL	6.44%	WLWL	7.57%
LLLW	5.61%	WLLL	7.54%
WLLW	4.82%	WWWW	7.03%
LWLW	4.14%	WLWW	3.52%
WWLL	4.05%	RLWL	3.09%
LWLL	3.80%	LLWW	2.58%
rest	24.87%	rest	16.95%

Table 8.: Four Finger Correlations for left hand fingers on BKA data

thumb, index middle, ring		index, middle ring, small	
RRRR	24.15%	RRRR	30.70%
WWWW	10.79%	LRRR	9.99%
RLRR	6.87%	WWWR	9.36%
WRRR	6.44%	WRRR	8.21%
RRRW	5.04%	RRWR	7.18%
RWRR	4.58%	WRWR	5.89%
WWRW	4.29%	WWWW	5.18%
WWRR	3.95%	WWRR	2.43%
RWWW	3.75%	WRWW	1.95%
WRRW	3.49%	ARRR	1.92%
rest	26.67%	rest	17.20%

Table 9.: Five Finger Correlations for right hand fingers on BKA data

thumb, index, middle ring, small	
LLLLL	17.75%
LRLLL	7.68%
WWWWL	6.97%
WWWWW	6.21%
WLLLL	6.10%
WWLWL	4.28%
LLLWL	4.20%
WWLLL	3.83%
LWLLL	3.66%
WLLWL	3.57%
rest	35.75%

Table 10.: Five Finger Correlations for left hand fingers on BKA data

thumb, index, middle ring, small	
RRRRR	23.58%
RLRRR	6.72%
WWWWR	6.61%
WRRRR	6.21%
RWRRR	4.41%
RRRWR	4.29%
WWWWW	4.18%
WWRRR	3.69%
WWRWR	3.29%
WLRRR	3.00%
rest	34.02%

C. Correlation Tables SD9 4 Classes

Table 11.: Two Finger Correlations for right hand fingers on SD9 (4 classes)

thumb index		index middle		middle ring		ring small	
WW	26.53%	RR	35.12%	RR	41.66%	RR	46.93%
RR	25.94%	WW	16.69%	RW	25.75%	WR	29.73%
WR	11.82%	WR	15.98%	WW	19.47%	WW	16.76%
RA	8.36%	LR	10.33%	AR	5.20%	AR	1.71%
RL	8.21%	AR	8.06%	WR	2.04%	RW	1.67%
RW	6.43%	AA	4.27%	AA	1.45%	LR	1.04%
WL	6.02%	LW	2.45%	RL	1.15%	RA	0.89%
WA	2.27%	RW	2.16%	RA	0.93%	AA	0.67%
AA	1.75%	RA	1.97%	AW	0.85%	LW	0.22%
AR	1.49%	LA	1.11%	LR	0.71%	RL	0.11%
rest	1.19%	rest	1.86%	rest	0.78%	rest	0.26%

Table 12.: Two Finger Correlations for left hand fingers on SD9 (4 classes)

thumb index		index middle		middle ring		ring small	
LL	24.05%	LL	32.42%	LL	46.51%	LL	55.58%
WW	21.56%	WW	16.17%	LW	19.41%	WL	25.69%
WL	11.34%	WL	14.91%	WW	17.10%	WW	11.38%
LA	10.19%	RL	10.71%	AL	7.55%	AL	2.42%
LR	9.59%	AL	9.33%	WL	3.38%	LW	2.04%
LW	9.48%	AA	5.65%	AA	2.38%	AA	1.15%
WR	4.91%	LW	2.64%	RL	1.23%	LA	0.93%
AA	3.09%	RA	2.30%	LA	1.12%	RL	0.26%
AL	2.16%	LA	2.27%	AW	0.63%	RA	0.15%
WA	1.97%	RW	1.45%	LR	0.33%	WA	0.11%
rest	1.67%	rest	2.16%	rest	0.37%	rest	0.30%

Table 13.: Three Finger Correlations for right hand fingers on SD9 (4 classes)

thumb, index middle		index, middle ring		middle, ring small	
RRR	23.78%	RRR	24.19%	RRR	39.69%
WWW	15.01%	WWW	15.46%	RWR	18.06%
WWR	11.11%	WRW	10.89%	WWR	10.78%
WRR	10.00%	RRW	9.92%	WWW	8.70%
RLR	6.61%	LRR	6.54%	RWW	7.69%
RAR	5.69%	ARR	5.91%	ARR	4.61%
RWR	4.72%	WRR	5.02%	WRR	1.97%
WLR	3.42%	LRW	3.49%	RRW	1.52%
RAA	2.53%	AAR	2.60%	AAR	0.93%
WLW	2.04%	LWW	2.04%	RLR	0.82%
rest	15.09%	rest	13.94%	rest	5.24%

Table 14.: Three Finger Correlations for left hand fingers on SD9 (4 classes)

thumb, index middle		index, middle ring		middle, ring small	
LLL	21.26%	LLL	23.35%	LLL	44.24%
WWW	12.08%	WWW	14.13%	LWL	14.28%
WLL	9.37%	RLL	8.85%	WWL	10.97%
WWL	9.11%	LLW	8.62%	ALL	6.99%
LRL	7.10%	WLW	8.40%	WWW	6.13%
LAL	6.54%	ALL	7.81%	LWW	4.94%
LWL	5.43%	WLL	6.51%	WLL	3.20%
LWW	3.87%	AAL	3.75%	LLW	1.71%
LAA	3.35%	LWW	2.12%	AAL	1.45%
WRL	3.20%	WWL	2.04%	RLL	1.15%
rest	18.70%	rest	14.42%	rest	4.94%

Table 15.: Four Finger Correlations for right hand fingers on SD9 (4 classes)

thumb, index middle, ring		index, middle ring, small	
RRRR	18.99%	RRRR	23.19%
WWWW	14.12%	WWWR	8.18%
WWRW	8.10%	WWWW	7.28%
WRRW	5.46%	RRWR	6.91%
RLRR	5.02%	WRWR	6.76%
RARR	4.38%	LRRR	6.21%
WRRR	4.31%	ARRR	5.80%
RRRW	4.12%	WRRR	4.50%
WWRR	3.01%	WRWW	4.12%
RWRW	2.71%	LRWR	3.08%
rest	29.77%	rest	23.97%

Table 16.: Four Finger Correlations for left hand fingers on SD9 (4 classes)

thumb, index middle, ring		index, middle ring, small	
LLLL	16.51%	LLLL	22.08%
WWWW	10.86%	WWWL	8.85%
LRLL	6.32%	RLLL	8.70%
WWLW	5.87%	ALLL	7.36%
LALL	5.69%	LLWL	6.39%
WLLL	5.61%	WLLL	6.10%
LLLW	4.54%	WLWL	5.91%
WLLW	3.64%	WWWW	5.28%
WWLL	3.23%	AALL	3.35%
LWWW	3.12%	WLWW	2.45%
rest	34.61%	rest	23.53%

Table 17.: Five Finger Correlations for right hand fingers on SD9 (4 classes)

thumb, index, middle ring, small	
RRRRR	18.43%
WWWWR	7.28%
WWWWW	6.84%
WWRWR	4.76%
RLRRR	4.76%
RARRR	4.38%
WRRRR	3.86%
RRRWR	3.42%
WWRWW	3.34%
WRRWR	3.23%
rest	39.69%

Table 18.: Five Finger Correlations for left hand fingers on SD9 (4 classes)

thumb, index, middle ring, small	
LLLLL	16.06%
WWWWL	6.51%
LRLLL	6.25%
LALLL	5.46%
WLLLL	4.87%
WWWWW	4.35%
WWLWL	3.90%
LLLWL	3.61%
WWLLL	2.94%
LWLLL	2.94%
rest	43.12%

D. Results - Correction of Probabilities

In the following, you can find the complete table for testing different correction measurement. The table shows the results for the right index finger and different numbers of adjacent fingers.

Table 19.: Results for different probability correction approaches and two adjacent fingers

adjacent fingers	searches	hit db bins	hit subjects	% of avg.	threshold	correction type
2	2700	8387	1582026	43%	0.01	-1
2	2700	8385	1581014	43%	0.01	0
2	2700	8386	1581960	43%	0.01	1
2	2700	8388	1580494	43%	0.01	2
2	2700	8386	1580459	43%	0.01	3
2	2700	8385	1581894	43%	0.01	4
2	2700	8386	1581960	43%	0.01	5
2	2700	8385	1581014	43%	0.01	6
2	2700	8370	1585897	43%	0.025	-1
2	2700	8372	1584155	43%	0.025	0
2	2700	8373	1585492	43%	0.025	1
2	2700	8376	1583045	43%	0.025	2
2	2700	8375	1583982	43%	0.025	3
2	2700	8370	1585122	43%	0.025	4
2	2700	8371	1585040	43%	0.025	5
2	2700	8372	1584155	43%	0.025	6

Table 20.: Results for different probability correction approaches and three adjacent fingers

adjacent fingers	searches	hit db bins	hit subjects	% of avg.	threshold	correction type
3	2700	24350	1343831	37%	0.01	-1
3	2700	24339	1343763	37%	0.01	0
3	2700	24347	1343874	37%	0.01	1
3	2700	24343	1342504	37%	0.01	2
3	2700	24343	1342927	37%	0.01	3
3	2700	24340	1343873	37%	0.01	4
3	2700	24346	1343798	37%	0.01	5
3	2700	24337	1343801	37%	0.01	6
3	2700	24214	1344698	37%	0.025	-1
3	2700	24235	1347054	37%	0.025	0
3	2700	24223	1345092	37%	0.025	1
3	2700	24260	1343629	37%	0.025	2
3	2700	24240	1345163	37%	0.025	3
3	2700	24237	1347436	37%	0.025	4
3	2700	24211	1344551	37%	0.025	5
3	2700	24240	1346799	37%	0.025	6

Table 21.: Results for different probability correction approaches and four adjacent fingers

adjacent fingers	searches	hit db bins	hit subjects	% of avg.	threshold	correction type
4	2700	87661	1285028	35%	0.01	-1
4	2700	87653	1283947	35%	0.01	0
4	2700	87647	1285011	35%	0.01	1
4	2700	87693	1282117	35%	0.01	2
4	2700	87695	1282334	35%	0.01	3
4	2700	87672	1284426	35%	0.01	4
4	2700	87656	1285031	35%	0.01	5
4	2700	87667	1282915	35%	0.01	6
4	2700	87258	1288761	35%	0.025	-1
4	2700	87341	1287779	35%	0.025	0
4	2700	87306	1288412	35%	0.025	1
4	2700	87493	1283823	35%	0.025	2
4	2700	87443	1284870	35%	0.025	3
4	2700	87378	1288317	35%	0.025	4
4	2700	87269	1288742	35%	0.025	5
4	2700	87428	1286463	35%	0.025	6

Bibliography

[1] Federal Bureau of Investigation, "Ngi sytsem fact sheet," Federal Bureau of Investigation, Tech. Rep., 2018. [Online]. Available: https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet/view

[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer London, 2009.

[3] A. Mani and M. Nadeski, "Processing solutions for biometric systems," Texas Instruments, Tech. Rep., 2015. [Online]. Available: http://www.ti.com/lit/wp/spry289/spry289.pdf

[4] U. D. of Justice and N. I. of Justice, *The Fingerprint Sourcebook*. CreateSpace Independent Publishing Platform, 2012.

[5] M. Kücken and A. C. Newell, "Fingerprint formation," *Journal of Theoretical Biology*, vol. 235, no. 1, pp. 71 – 83, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519304006198

[6] A. K. Jain, R. Bolle, and S. Pankanti, Eds., *Biometrics: Personal Identification in Networked Society*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[7] H. C. Lee, R. Ramotowski, and R. E. Gaensslen, *Advances in Fingerprint Technology, Second Edition*. CRC Press, 2001.

[8] A. K. Jain, A. A. Ross, and K. Nandakumar, "Fingerprint recognition," in *Introduction to Biometrics*. Springer US, 2011, ch. Fingerprint Recognition, pp. 51–96.

[9] S. Pankanti, S. Prabhakar, and A. Jain, "On the individuality fingerprints," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* IEEE Comput. Soc, 2001.

[10] S. Yoon and A. K. Jain, "Longitudinal study of fingerprint recognition," *Proceedings of the National Academy of Sciences*, vol. 112, no. 28, pp. 8555–8560, jun 2015.

[11] ISO/IEC TC JTC1 SC37 Biometrics, "ISO/IEC 2382-37:2017 Information technology – Vocabulary – Part 37: Biometrics (Preview)," *International Organization for Standardization and International Electrotechnical Committee*, 2017. [Online]. Available: https://www.iso.org/standard/66693.html

[12] ——, "ISO/IEC 19795-1:2006 Information Technology - Biometric Performance Testing and Reporting - Part 1: Principles and Framework," *International Organization for Standardization and International Electrotechnical Committee*, 2006. [Online]. Available: https://www.iso.org/standard/41447.html

[13] ——, "ISO/IEC 19794-2:2011 Information Technology - Biometric Data Interchange Formats - Part 2: Finger Minutiae Data," *International Organization for Standardization and International Electrotechnical Committee*, 2011. [Online]. Available: https://www.iso.org/standard/41447.html

[14] ——, "ISO/IEC 29794-4:2017 Information technology – Biometric sample quality – Part 4: Finger image data," *International Organization for Standardization and International Electrotechnical Committee*, 2017. [Online]. Available: https://www.iso.org/standard/41447.html

[15] M. Galar, J. Derrac, D. Peralta, I. Triguero, D. Paternain, C. Lopez-Molina, S. García, J. M. Benítez, M. Pagola, E. Barrenechea, H. Bustince, and F. Herrera, "A survey of fingerprint classification part i: Taxonomies on feature extraction methods and learning models," *Knowledge-Based Systems*, vol. 81, no. Supplement C, pp. 76–97, jun 2015.

[16] ——, "A survey of fingerprint classification part ii: Experimental analysis and ensemble proposal," *Knowledge-Based Systems*, vol. 81, no. Supplement C, pp. 98–116, jun 2015.

[17] I. Kavati, M. V. N. K. Prasad, and C. Bhagvati, "Search space reduction in biometric databases: A review," in *Developing Next-Generation Countermeasures for Homeland Security Threat Prevention*. Hershey, PA, USA: IGI Global, 2017, pp. 236–262.

[18] F. Galton, *Finger Prints*. Macmillan and Co., 1892. [Online]. Available: http://galton.org/books/finger-prints/galton-1892-fingerprints-1up.pdf

[19] E. Henry, *Classification and uses of finger prints*. George Routledge and Sons, Limited, 1900. [Online]. Available: http://resource.nlm.nih.gov/1306026

[20] B. Moayer and K. Fu, "A syntactic approach to fingerprint pattern recognition," *Pattern Recognition*, vol. 7, no. 1-2, pp. 1–23, 1975.

[21] ——, "An application of stochastic languages to fingerprint pattern recognition," *Pattern Recognition*, vol. 8, no. 3, pp. 173–179, 1976.

[22] C. L. Wilson, G. T. Candela, P. J. Grother, C. I. Watson, and R. A. Wilkinson, "Massively parallel neural network fingerprint classification system," NIST, techreport 4880, 1992. [Online]. Available: https://archive.org/stream/massivelyparalle4880wils#page/60/mode/2up

[23] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "FingerCode: a filterbank for fingerprint representation and matching," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2. IEEE Comput. Soc, 1999, p. 193 Vol. 2.

[24] K. Rao and K. Balck, "Type classification of fingerprints: A syntactic approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 3, pp. 223–231, 1980.

[25] J.-H. Chang and K.-C. Fan, "A new model for fingerprint classification by ridge distribution sequences," *Pattern Recognition*, vol. 35, no. 6, pp. 1209–1223, 2002.

[26] T. M. Rao, "Feature extraction for fingerprint classification," *Pattern Recognition*, vol. 8, no. 3, pp. 181–192, 1976.

[27] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognition*, vol. 17, no. 3, pp. 295–303, Jan. 1984.

[28] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, vol. 37, no. 11, pp. 2233–2243, 2004.

[29] L. Wang and M. Dai, "Application of a new type of singular points in fingerprint classification," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1640–1650, 2007, verbesserung von Zang_04.

[30] K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389–404, Mar. 1996.

[31] M. Ballan, F. Sakarya, and B. Evans, "A fingerprint classification technique using directional images," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, vol. 1. IEEE Comput. Soc, 1997, pp. 101–104 vol.1.

[32] C. Klimanee and D. T. Nguyen, "Classification of fingerprints using singular points and their principal axes," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 2. IEEE, 2004, pp. 849–852 Vol.2.

[33] I. S. Msiza, B. Leke-Betechuoh, F. V. Nelwamondo, and N. Msimang, "A fingerprint pattern classification approach based on the coordinate geometry of singularities," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2009, pp. 510–517.

[34] L. Webb and M. Mathekga, "Towards a complete rule-based classification approach for flat fingerprints," in *2014 Second International Symposium on Computing and Networking*. IEEE, dec 2014, pp. 549–555.

[35] J.-M. Guo, Y.-F. Liu, J.-Y. Chang, and J.-D. Lee, "Fingerprint classification based on decision tree from singular points and orientation field," *Expert Systems with Applications*, vol. 41, no. 2, pp. 752–764, feb 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413006003

[36] L. Hong and A. Jain, "Classification of fingerprint images," in *PROCEEDINGS OF THE 11TH SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS, KANGERLUSSUAQ*, 1999, pp. 7–11.

[37] W. Liu, Z. Ye, H. Chen, and H. Li, "Ridgeline based 2-layer classifier in fingerprint classification," in *2009 International Workshop on Intelligent Systems and Applications*. IEEE, may 2009, pp. 1–4.

[38] B.-H. Cho, J.-S. Kim, J.-H. Bae, I.-G. Bae, and K.-Y. Yoo, "Core-based fingerprint image classification," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, 2000, pp. 859–862 vol.2.

[39] S. C. Dass and A. K. Jain, "Fingerprint classification using orientation field flow curves," in *In Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, 2004, pp. 650–655.

[40] X. Wang and M. Xie, *Fingerprint Classification: An Approach Based on Singularities and Analysis of Fingerprint Structure*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 324–329. [Online]. Available: https://doi.org/10.1007/978-3-540-25948-0_45

[41] X. Wang, F. Wang, J. Fan, and J. Wang, "Fingerprint classification based on continuous orientation field and singular points," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, Nov 2009, pp. 189–193.

[42] L. Fan, S. Wang, H. Wang, and T. Guo, "Singular points detection based on zero-pole model in fingerprint images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 929–940, 2008.

[43] L.-m. Liu, C.-y. Huang, and D. C. D. Hung, “A directional approach to fingerprint classification,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 02, pp. 347–365, mar 2008.

[44] K. Dorasamy, L. Webb, J. Tapamo, and N. P. Khanyile, “Fingerprint classification using a simplified rule-set based on directional patterns and singularity features,” in *2015 International Conference on Biometrics (ICB)*. IEEE, may 2015, pp. 400–407.

[45] S. C. Chua, E. Wong, and A. Tan, “A fuzzy rule-based fingerprint image classification,” *International Journal of Applied Engineering Research*, vol. 11, pp. 7920–7925, 01 2016. [Online]. Available: https://www.researchgate.net/publication/306226059_A_fuzzy_rule-based_fingerprint_image_classification

[46] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, nov 1901.

[47] P. A. Hughes and A. D. P. Green, “The use of neural networks for fingerprint classification,” in *1991 Second International Conference on Artificial Neural Networks*, Nov 1991, pp. 79–81.

[48] J. D. Bowen, “The home office automatic fingerprint pattern classification project,” in *IEE Colloquium on Neural Networks for Image Processing Applications*, Oct 1992, pp. 1/1–1/5. [Online]. Available: http://ieeexplore.ieee.org/document/193709/

[49] M. Kamijo, “Classifying fingerprint images using neural network: deriving the classification state,” in *IEEE International Conference on Neural Networks*. IEEE, 1993, pp. 1932–1937 vol.3.

[50] Z. J. Geng and W. Shen, “Fingerprinting classification using fuzzy cerebellar model arithmetic computer neural networks,” *Journal of Electronic Imaging*, vol. 6, no. 3, p. 311, jul 1997.

[51] K. A. Nagaty, "Fingerprints classification using artificial neural networks: a combined structural and statistical approach," *Neural Networks*, vol. 14, no. 9, pp. 1293–1305, 2001.

[52] S. M. Mohamed and H. Nyongesa, "Automatic fingerprint classification system using fuzzy neural techniques," in *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*, vol. 1. IEEE, 2002, pp. 358–362.

[53] C. Jin and P. Jin, "Fingerprint classification in dct domain using rbf neural networks," *Journal of Information Science and Engineering*, vol. 25, pp. 1955–1962, 11 2009. [Online]. Available: https://www.researchgate.net/publication/220587154_Fingerprint_Classification_in_DCT_Domain_using_RBF_Neural_Networks

[54] S. Thepade, D. Parekh, and V. Tiwari, "LBG algorithm for fingerprint classification," in *International Journal of Advances in Engineering & Technology*, Nov. 2012. [Online]. Available: https://www.semanticscholar.org/paper/Lbg-Algorithm-for-Fingerprint-Classification-Thepade-Parekh/f14323f5d09d1e974a631806f17969520f363325?tab=abstract

[55] R. Wang, C. Han, and T. Guo, "A novel fingerprint classification method based on deep learning," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, dec 2016, pp. 931–936.

[56] C. L. Wilson, G. T. Candela, and C. I. Watson, "Neural network fingerprint classification," *Journal of Artificial Neural Networks*, vol. 1, no. 2, pp. 203–228, 1994.

[57] C. L. Wilson and J. L. Blue, "Improving neural network performance for character and fingerprint classification by altering network dynamics," NIST, techreport 5695, 1995.

[58] S. N. Sarbadhikari, J. Basak, S. K. Pal, and M. K. Kundu, "Noisy fingerprints classification with directional fft based features using mlp," *Neural Computing & Applications*, vol. 7, no. 2, pp. 180–191, Jun 1998.

[59] K. Moscinska and G. Tyma, "Neural network based fingerprint classification," in *1993 Third International Conference on Artificial Neural Networks*, May 1993, pp. 229–232. [Online]. Available: http://ieeexplore.ieee.org/document/263221/

[60] U. Halici and G. Ongun, "Fingerprint classification through self-organizing feature maps modified to treat uncertainties," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1497–1512, Oct 1996.

[61] S. Bernard, N. Boujemaa, D. Vitale, and C. Bricot, "Fingerprint classification using kohonen topologic map," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 3. IEEE, 2001, pp. 230–233 vol.3.

[62] S. R. Borra, G. J. Reddy, and E. S. Reddy, "Classification of fingerprint images with the aid of morphological operation and agnn classifier," *Applied Computing and Informatics*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210832717301576

[63] V. N. Vapnik, *Statistical Learning Theory*. JOHN WILEY & SONS INC, 1998. [Online]. Available: http://www.ebook.de/de/product/3602628/vladimir_naumovich_vapnik_statistical_learning_theory.html

[64] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun 1998.

[65] Y. Yao, P. Frasconi, and M. Pontil, "Fingerprint classification with combinations of support vector machines," in *Audio- and Video-Based Biometric Person Authentication: Third International Conference, AVBPA 2001 Halmstad, Sweden, June 6–8, 2001 Proceedings*, J. Bigun and F. Smeraldi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 253–258. [Online]. Available: https://doi.org/10.1007/3-540-45344-X_37

[66] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Int. Res.*, vol. 2, no. 1, pp. 263–286,

Jan. 1995. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622826.1622834

[67] A. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.

[68] J.-K. Min, J.-H. Hong, and S.-B. Cho, "Effective fingerprint classification by localized models of support vector machines," in *Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5-7, 2006. Proceedings*, D. Zhang and A. K. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 287–293.

[69] ——, "Fingerprint classification based on subclass analysis using multiple templates of support vector machines," *Intelligent Data Analysis*, vol. 14, pp. 369–384, 01 2010. [Online]. Available: https://www.researchgate.net/publication/220571687_Fingerprint_classification_based_on_subclass_analysis_using_multiple_templates_of_support_vector_machines

[70] J.-H. Hong, J.-K. Min, U.-K. Cho, and S.-B. Cho, "Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive bayes classifiers," *Pattern Recognition*, vol. 41, no. 2, pp. 662–671, 2008.

[71] J. Li, W.-Y. Yau, and H. Wang, "Combining singular points and orientation image information for fingerprint classification," *Pattern Recognition*, vol. 41, no. 1, pp. 353–366, 2008.

[72] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, aug 1992.

[73] A. Fitz and R. Green, "Fingerprint classification using a hexagonal fast fourier transform," *Pattern Recognition*, vol. 29, no. 10, pp. 1587 – 1597, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320396000180

[74] S. Wang, W. W. Zhang, and Y. S. Wang, "Fingerprint classification by directional fields," in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, 2002, pp. 395–399.

[75] U. Rajanna, A. Erol, and G. Bebis, "A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion," *Pattern Analysis and Applications*, vol. 13, no. 3, pp. 263–272, 2010.

[76] J. Luo, D. Song, C. Xiu, S. Geng, and T. Dong, "Fingerprint classification combining curvelet transform and gray-level cooccurrence matrix," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–15, 2014.

[77] H. O. Nyongesa, S. Al-Khayatt, S. M. Mohamed, and M. Mahmoud, "Fast robust fingerprint feature extraction and classification," *Journal of Intelligent and Robotic Systems*, vol. 40, no. 1, pp. 103–112, 2004.

[78] T. Kristensen, J. Borthen, and K. Fyllingsnes, "Comparison of neural network based fingerprint classification techniques," in *2007 International Joint Conference on Neural Networks*. IEEE, 2007, pp. 1043–1048.

[79] D. Maio and D. Maltoni, "A structural approach to fingerprint classification," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 3. IEEE, 1996, pp. 578–585 vol.3.

[80] A. Senior, "A hidden markov model fingerprint classifier," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, vol. 1. IEEE Comput. Soc, 1997, pp. 306–310 vol.1.

[81] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint classification by directional image partitioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 402–421, 1999.

[82] A. K. Jain and S. Minut, "Hierarchical kernel fitting for fingerprint classification and alignment," in *Object recognition supported by user interaction for service robots*, vol. 2. IEEE Comput. Soc, 2002, pp. 469–473 vol.2.

[83] M. Neuhaus and H. Bunke, *A Graph Matching Based Approach to Fingerprint Classification Using Directional Variance.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 191–200.

[84] H. W. Jung and J. H. Lee, "Fingerprint classification using the stochastic approach of ridge direction information," in *2009 IEEE International Conference on Fuzzy Systems.* IEEE, 2009, pp. 169–174.

[85] M. Liu, "Fingerprint classification based on adaboost learning from singularity features," *Pattern Recognition*, vol. 43, no. 3, pp. 1062–1070, 2010.

[86] M. M. Chong, H. N. Tan, L. Jun, and R. K. Gay, "Geometric framework for fingerprint image classification," *Pattern Recognition*, vol. 30, no. 9, pp. 1475–1488, 1997.

[87] Y. Qi, J. Tian, and R.-W. Dai, "Fingerprint classification system with feedback mechanism based on genetic algorithm," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, vol. 1. IEEE Comput. Soc, 1998, pp. 163–165 vol.1.

[88] J. Hu and M. Xie, "Fingerprint classification based on genetic programming," in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 6. IEEE, 2010, pp. 193–196.

[89] C. H. Park and H. Park, "Fingerprint classification using fast fourier transform and nonlinear discriminant analysis," *Pattern Recognition*, vol. 38, no. 4, pp. 495–503, 2004.

[90] X. Tan, B. Bhanu, and Y. Lin, "Fingerprint classification based on learned features," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 3, pp. 287–300, 2005.

[91] K. C. Leung and C. H. Leung, "Improvement of fingerprint retrieval by a statistical classifier," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 59–69, 2011.

[92] G. Vitello, F. Sorbello, G. I. M. Migliore, V. Conti, and S. Vitabile, "A novel technique for fingerprint classification based on fuzzy c-means and naive bayes classifier," in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, jul 2014, pp. 155–161.

[93] H.-W. Jung and J.-H. Lee, "Noisy and incomplete fingerprint classification using local ridge distribution models," *Pattern Recognition*, vol. 48, no. 2, pp. 473–484, 2015.

[94] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar 1998.

[95] G. T. Candela, P. J. Grother, C. I. Watson, R. A. Wilkinson, and C. Wilson, "Pcasys - a pattern-level classification automation system for fingerprints," NIST Research Library, techreport 5647, 1995. [Online]. Available: https://www.nist.gov/publications/pcasys-pattern-level-classification-automation-system-fingerprints

[96] A. Lumini, D. Maio, and D. Maltoni, "Continuous versus exclusive classification for fingerprint retrieval," *Pattern Recognition Letters*, vol. 18, no. 10, pp. 1027–1034, oct 1997.

[97] R. Cappelli, D. Maio, and D. Maltoni, "Combining fingerprint classifiers," in *Multiple Classifier Systems*. Springer Berlin Heidelberg, 2000, pp. 351–361.

[98] ——, "A multi-classifier approach to fingerprint classification," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 136–144, 2002.

[99] R. Cappelli and D. Maio, *The State of the Art in Fingerprint Classification*. New York, NY: Springer New York, 2004, pp. 183–205. [Online]. Available: https://doi.org/10.1007/0-387-21685-5_9

[100] R. Cappelli and D. Maltoni, "Multispace KL for pattern representation and classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 977–996, 2001.

[101] R. Cappelli, D. Maio, D. Maltoni, and L. Nanni, "A two-stage fingerprint classification system," in *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications - WBMA 03*, ser. WBMA '03. New York, NY, USA: ACM, 2003, pp. 95–99.

[102] M. S. Pattichis, G. Panayi, A. C. Bovik, and S.-P. Hsu, "Fingerprint classification using an AM-FM model," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 951–954, 2001.

[103] A. Senior, "A combination fingerprint classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1165–1174, 2001.

[104] A. Senior and R. Bolle, *Fingerprint Classification by Decision Fusion.* New York, NY: Springer New York, 2004, pp. 207–227.

[105] G. L. Marcialis, F. Roli, and P. Frasconi, *Fingerprint Classification by Combination of Flat and Structural Approaches.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 241–246. [Online]. Available: https://doi.org/10.1007/3-540-45344-X_35

[106] Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi, and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Pattern Recognition*, vol. 36, no. 2, pp. 397–406, 2003, biometrics.

[107] Z. Han and C.-P. Liu, "Fingerprint classification based on statistical features and singular point information," in *Advances in Biometric Person Authentication: International Wokshop on Biometric Recognition Systems, IWBRS 2005, Beijing, China, October 22-23, 2005. Proceedings*, S. Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, and D. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 119–126. [Online]. Available: https://doi.org/10.1007/11569947_15

[108] S. Shah and P. Sastry, "Fingerprint classification using a feedback-based line detector," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 85–94, 2004.

[109] K. Cao, L. Pang, J. Liang, and J. Tian, "Fingerprint classification by a hierarchical classifier," *Pattern Recognition*, vol. 46, no. 12, pp. 3186–3197, 2013.

[110] T. H. Le and H. T. Van, "Fingerprint reference point detection for image retrieval based on symmetry and variation," *Pattern Recognition*, vol. 45, no. 9, pp. 3360–3372, 2012, best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320312000957

[111] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[112] R. Cappelli, "SFinGe: an Approach to Synthetic Fingerprint Generation," 01 2004. [Online]. Available: http://biolab.csr.unibo.it/research.asp?organize=Activities&select=&selObj=12&pathSubj=111%7C%7C12&Req=&

[113] D. Peralta, I. Triguero, S. García, Y. Saeys, J. M. Benitez, and F. Herrera, "Distributed incremental fingerprint identification with reduced database penetration rate using a hierarchical classification based on feature fusion and selection," *Knowledge-Based Systems*, vol. 126, no. Supplement C, pp. 91 – 103, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095070511730134X

[114] Chinese Academy of Sciences' Institute of Automation (CASIA), "Casia-fingerprintv5," Nov. 2016, last Accessed: 2017-10-12. [Online]. Available: http://english.ia.cas.cn/db/201611/t20161101_169922.html

[115] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J. J. Igarza, C. Vivaracho, D. Escudero, and Q. I. Moro, "Mcyt baseline corpus: a bimodal biometric database," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, Dec 2003.

[116] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.