

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

RESEARCH IN FINGERPRINT DAMAGE SIMULATIONS

VÝZKUM V OBLASTI SIMULACÍ POŠKOZENÍ OTISKU PRSTU

DOCTORAL THESIS
DISERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE

Ing. ONDŘEJ KANICH

SUPERVISOR
VEDOUCÍ PRÁCE

prof. Ing., Dipl.-Ing. MARTIN DRAHANSKÝ, Ph.D.

Abstract

The goal of this research is to develop methods for fingerprint damage simulations. In the first part of this thesis the emphasis is placed on a summary of the current knowledge of synthetic fingerprint generation and the damage to these fingerprints. Moreover, general information about fingerprints, fingerprint recognition, and phenomena that damage fingerprints including skin diseases are stated herein. This thesis contains the design and implementation of the SyFDaS application for generation and modular damaging of fingerprints. The next part is a description of methods for damage by swipe mode, narrow sensor, damaged sensor, pressure and moisture, skin distortion, warts, atopic eczema, and psoriasis. Several other types of damage, including fingerprint spoofs, are analysed. Overall, there are 43 basic damages which were visually verified. Due to damage combinations, there are 1,171 types of damage and 348,300 fingerprint images generated, which were evaluated by four different quality measurement methods.

Abstrakt

Cílem této práce je vyvinout metody simulací poškozování otisků prstů. V první části je kladen důraz na shrnutí stávajících znalostí v oblasti generování syntetických otisků prstů a jejich poškozování. Dále jsou uvedeny informace o otiscích prstů obecně, jejich rozpoznávání a vlivy, které otisky poškozují, včetně onemocnění kůže. Práce obsahuje návrh a implementaci aplikace SyFDaS pro generování a modulární poškozování otisků prstů. Další částí je popis metod pro poškozování vlivem průtahového režimu, zúženého snímače, poškozeného snímače, přitlaku a vlhkosti, zkrvení pokožky, bradavic, atopického ekzému a lupénky. Dále je analyzováno několik dalších typů poškození včetně falzifikátů otisků prstů. Celkově je uvedeno 43 základních poškození, která jsou vizuálně verifikována. Díky kombinování poškození je využito 1 171 typů poškození a vygenerováno 348 300 obrázků otisků prstů, které jsou vyhodnoceny čtyřmi různými metodami posuzování kvality.

Keywords

fingerprint, synthetic fingerprint, fingerprint generation, damage simulation, sensors, skin disease, fingerprint spoof, Petri net

Klíčová slova

otisk prstu, syntetický otisk prstu, generování otisku prstu, simulace poškození, senzory, onemocnění kůže, falzifikát otisku prstu, Petriho síť

Citation

Ondřej Kanich: Research in Fingerprint Damage Simulations, doctoral thesis, Brno, FIT BUT, Brno, Czech Republic, 2018

Research in Fingerprint Damage Simulations

Declaration

I hereby declare that this thesis is my original work and has been created under the supervision of prof. Ing., Dipl.-Ing. Martin Drahanský, Ph.D. Some results were achieved in cooperation with bachelor's or master's degree students led by my supervisor or myself. Where other sources of information have been used, they have been duly acknowledged.

.....
Ondřej Kanich
May 30, 2018

Acknowledgements

I wish to thank Martin Drahanský for his support, advice, and his valuable and inspiring consultations during his supervision of this work. I would also like to thank my whole family for their endless support, great hints, strong motivation, and ceaseless patience. I appreciate Eva Březinová for her advice and insight of the medical part of my work. I acknowledge all students that have helped with this work, namely Milan Bárta, Tomáš Oravec, Štěpánka Barotová, and David Košťák. Last but not least, I wish to thank all former and current members of the STRaDe research group for fruitful discussions, advice, and other assistance which was immensely helpful.

This thesis was supported by several projects: *New solutions for multimodal biometrics – enhancement of security and reliability of biometric technologies* (COST LD14013), *Reliability and Security in IT* (FIT-S-14-2486), *IT4Innovations excellence in science* (LQ1602), *Tools and methods for video and image processing to improve effectivity of rescue and security services operations* (VI20172020068), *Secure and Reliable Computer Systems* (FIT-S-17-4014).

Contents

1	Introduction	4
2	State of the Art Fingerprint Technology	6
2.1	Introduction to Biometrics	6
2.2	Fingerprints	7
2.2.1	Ridges (Papillary Lines)	7
2.2.2	Classification of Fingerprints	8
2.2.3	Fingerprint Minutiae	9
2.3	Sensor Technologies for Fingerprint Acquisition	10
2.3.1	Optical Technology	11
2.3.2	Capacitive Technology	12
2.3.3	Thermal Technology	13
2.3.4	Ultrasonic Technology	14
2.3.5	Pressure Sensitive Technology	15
2.3.6	E-field Technology	15
2.3.7	Electro-Optical Technology	15
2.3.8	MEMS Technology	16
2.4	Fingerprint Recognition Process	17
3	Synthetic Fingerprint	18
3.1	Methods for Generating Synthetic Fingerprints	18
3.2	Phenomena Influencing a Fingerprint	19
3.3	SFinGe	21
4	SyFDaS – Synthetic Fingerprint Damage Simulator	23
4.1	Fingerprint Generation Petri Net	23
4.1.1	Design of the Fingerprint Generation Petri Net	24
4.1.2	Examples of the Initial Marking for Various Conditions	29
4.2	SyFDaS Core Design and Graphical User Interface	30
4.3	Enhancement of Fingerprint Generation	32
4.4	Touch-based Sensor Damage Simulation	33
4.4.1	Damaged Sensor	33
4.4.2	Pressure and Moisture	33
4.4.3	Fingerprint Distortion	34

4.5	Database Generation	42
5	Swipe Sensor Damage Simulation	45
5.1	Damage Analysis	45
5.2	Swipe Mode	46
5.2.1	Swipe Sensor Definition in Swipe Mode.....	47
5.3	Damages Exclusive to Swipe Sensors	49
5.3.1	Narrow Sensor	50
5.4	Examples of Damages.....	51
5.4.1	Pressure and Moisture in Swipe Mode	51
5.4.2	Narrow Sensor in Swipe Mode.....	53
5.4.3	Damaged Sensor in Swipe Mode.....	55
5.4.4	Distortion in Swipe Mode.....	56
5.5	Evaluation	57
5.5.1	Synthetic Database Used for Evaluation	58
5.5.2	Methods Used for Quality Measurement.....	58
5.5.3	Evaluation of Generated Narrow Images.....	60
5.5.4	Evaluation of Generated Normal Width Images.....	73
6	Skin Disease Simulation.....	88
6.1	Database of Fingerprints with Skin Diseases.....	89
6.1.1	Database Analysis.....	90
6.1.2	Description of Diseases in the Database.....	92
6.1.3	Skin Diseases Detection	97
6.2	Directly Simulated Diseases	101
6.2.1	Verruca Vulgaris (Warts)	101
6.2.2	Atopic Dermatitis	107
6.3	Simulation Based on Learning from Diseased Images	113
6.3.1	Psoriasis.....	114
6.4	Evaluation	117
6.4.1	Evaluation by the Dermatologist	118
6.4.2	Warts Damage Evaluation	118
6.4.3	Atopic Eczema Damage Evaluation	121
6.4.4	Psoriasis Damage Evaluation	125
7	Other Inspected Damages	128
7.1	Fingerprint Spoofs	128

7.1.1	Spoof Production	128
7.1.2	Spoof Images	129
7.1.3	Spoof Damage Analysis	132
7.2	Detergents or Lotions.....	133
8	Conclusion.....	134
	Research and Development Activities	145

1 Introduction

In the past decade, fingerprint technology has experienced an incredible boom. They moved from sci-fi movies to just about every personal device. Nowadays, almost every smartphone has a fingerprint reader and their placement in laptops now comes standard. The usage of these technologies in civil areas, like access control or security systems, is now a reality. With this massive expansion, however, there are problems that emerge. Mobile devices are focusing on minimalistic solutions. That usually means the cheapest (for example, the sensor has to be as small as possible), but still workable solutions. On the other hand, security and access control systems are focusing on the highest level of security. [1]

Keeping the performance with the smaller sensors means that algorithms must use every possible information in the sensing area. Cracking these devices (usually smartphones) is a prestigious thing. Producers of biometric systems have to react with new or better liveness detection subsystems. Algorithms that extract features then have to work with liveness detection as well. As a result, algorithms are becoming more sophisticated and complex. This leads to larger demands on testing and testing requires fingerprint database – large databases with not only many fingerprints from one finger but also many fingers. That means that many people (volunteers) are involved in the creation of various databases. The capture of so many fingerprints is a very time-consuming operation. It might seem like that when the database is finished, it can be used everywhere and everything is solved, but that is not true. And that is because fingerprints are considered as personal data and as such they are protected by various laws. The details of these laws can differ from country to country, so generally it can be said that usage of these databases is difficult. [1]

If only there would be a way to get huge databases without these legal concerns, with a lot of challenging fingerprints, and so on. There is one possibility and that is a synthetic fingerprint database. There are already ways to generate a synthetic fingerprint. It is not connected to any real person, thus it is not protected by legislation. The only problem is that they are usually perfect or only slightly damaged. What is needed is challenging fingerprints – damaged ones, and not only with some damage but with a specific damage. The challenge for mobile usage is small sensors (small sensing area); for security and access systems it could be skin disease. When someone has a skin disease that influences the fingerprint, the situation can occur where this person cannot use the access system or cannot get past the security. Fingerprint spoofing is also a problem for all applications. The potential damage done by successful spoofing to break into a smartphone or a highly secure building is different, but it is the same problem.

There are fingerprints that are not so common in the population and these should be in the databases as well. The situation where some kind of fingerprint has not been tested because it just did not appear in the database is unthinkable. This topic is closely related to the so-called Doddington's zoo [2] [3], which stated that the difficulty of comparing two biometric traits is not the same. The fingerprint on a user's left thumb could be easy to compare and the fingerprint on the user's right thumb could wreak havoc for the algorithms. Synthetic database could be prepared so it only contains the worst from the worst. This challenging database could be very beneficial for all types of testing. The usage of synthetic databases is not constricted only to test the algorithms – they can also be used as an educational

tool. Police experts on dactyloscopy can learn what diseased fingerprints looks like, developers of new systems can see the most challenging fingerprints in advance, etc.

This work focuses on how to specifically damage the perfect synthetic fingerprint so it can be used in these exemplary applications. The main aim is to describe the present technology in generating synthetic fingerprints with an emphasis placed on the simulation of a damaged fingerprint and to design and implement methods that take the perfect fingerprint and transform it into a more realistic damaged representation. These methods take in the input from various types of sensors as well as other phenomena in order to simulate a very specific damage done to a real fingerprint when it is acquired. This way it cannot only simulate a specific damage but also generate a fingerprint exposed to different environments.

In the second chapter, the current state of the art is described. There is information about biometrics, fingerprints, the process of fingerprint acquirement, and the sensor technologies associated with it. The third chapter is dedicated to synthetic fingerprints and everything connected with it, i.e. a way of generating a synthetic fingerprint, various data that can be used as inputs for the generator, the current available generators, and their functionality. The phenomena that influences a real fingerprint during the capturing process are also described. In the fourth chapter there is the design and implementation of the SyFDaS application. Starting with the theoretical Petri net background to the core design of the application, there are some basic touch-based damages and database generation methods listed. The fifth chapter is dedicated to damage simulations for swipe sensors. It shows the way in which the swipe sensor influences the phenomena created for touch sensors and describes new phenomena that are specific only to swipe sensors. Methods for implementation of these phenomena are included as well as examples and their evaluation. This chapter also includes an extensive introduction of evaluation methods. In the sixth chapter, skin diseases that influence fingerprints are described. The available database is analysed as well as the method of detecting skin disease. This chapter also contains information about the simulations of some diseases and their examples and evaluations. The seventh chapter is an introduction to other potential damages. A big part is dedicated to fingerprint spoofs. The last chapter is the conclusion, which sums up all essential information.

2 State of the Art Fingerprint Technology

This chapter describes the general information needed to understand the rest of the work that was done in this research. The main goal of this thesis is closely related to the fingerprints used in biometrics, thus the basic knowledge of biometrics with an emphasis on fingerprints and methods to acquiring them is covered. An integral part is also the way the recognition of fingerprints works, i.e. the processes that are necessary to acquire a fingerprint. All terms related to biometrics are consulted with [4]. [1]

2.1 Introduction to Biometrics

Before going further, an explanation is required as to what “biometrics” is. This term has a different meaning in information technology and in biology. *Biometrics* [4] [5] in the context of this work is an automated recognition of people based on their characteristic physiological and behavioural features. There are three basic approaches on how one can prove his or her electronic identity: (i) reveal something only he/she knows (knowledge), (ii) something he/she possesses (possession), and (iii) something he/she is (biometrics). In that order, the level of comfort and safety ascends when using these approaches. [1] [5]

The main advantage of biometric systems is that the biometric characteristic that is used to identify an individual cannot be lost or forgotten. This fact is also the greatest disadvantage of biometric characteristics. Whenever it is revealed, there is no way to change or delete it. Some biometric characteristics can also tell a lot about the individual’s health condition, so it violates one's privacy. [1] [5]

There are a few concepts that are important to biometrics. One is *inter-* and *intra-class variability*. Inter-class variability tells us how big the difference is between traits from different classes (people). On the contrary, intra-class variability tells us how big the difference is between traits from the same class (individual). When biometric characteristics are compared, there are nine basic properties [5] [6]:

- *Universality*, i.e. everyone should have this trait.
- *Uniqueness*, i.e. two persons should not have the same trait.
- *Permanence*, i.e. this trait should not vary over time.
- *Measurability*, i.e. this trait should be easy to acquire.
- *Performance*, i.e. this trait should not be changed or altered.
- *Acceptability*, i.e. the willingness of people to capture this trait.
- *Circumvention*, i.e. how difficult it is to falsify.
- *Price*, i.e. how much it costs to deploy a biometric system with this trait.
- *Maintenance*, i.e. how much it costs to maintain a biometric system with this trait.

There is no perfect biometric characteristic. Each one has its advantages and disadvantages based on these properties. [1] [5] [7] [8]

2.2 Fingerprints

This work is mainly focused on *fingerprints*, therefore, this subchapter studies them in more detail. The fingerprint, since 1880, is one of the biometric characteristics that has been used to identify people. Almost a hundred years earlier, it was already known that fingerprints are unique. Francis Galton counted the likelihood of two fingerprints being the same as 1 in 64 billion. That is one of the reasons why it is one of the most basic and widespread biometric characteristics that can be seen in everyday life. [1] [5]

In comparison with other biometric characteristics, its main advantages are its uniqueness, permanence, performance, circumvention, and price. It is pretty decent in other properties as well, but there are better characteristics for that (e.g. retina, DNA). One of them is acceptability. People very narrowly connect fingerprint scanning with criminal television shows and movies. When it comes to a larger acquirement processes, many of them have the inner feelings that they have done something really bad. Some people are also afraid of the forgery of their fingerprints, so they want to go through acquirement process only when serious crime is investigated. These are some reasons why fingerprints do not have a great acceptability. Nowadays, the position of fingerprint technology is getting better and people are more willing to accept this technology because of its everyday use. However, it is still very difficult to acquire a fingerprint database. [1] [5]

2.2.1 Ridges (Papillary Lines)

A fingerprint is created by capturing *ridges (papillary lines)*¹, [5] [6] [9] [10] which are protrusions in the internal side of hands (and feet as well). In Figure 2.1 the structure of the top side of the skin can be seen. In the *epidermis* portion, some types of minutiae and sweat pores are shown, which are described in Subchapter 2.2.3. The curvatures of the ridges are formed in the deeper layer – the *dermis*. The real ridges in the epidermis, which can be seen and captured as a fingerprint, are just a projection from the deeper layer (for example, wrinkles are formed in the same layer). This means that one cannot alter or delete the fingerprint by damaging the epidermis, for instance by a burn, abrasion, or cut. If damage like that is done, it will regenerate with the growth of skin in the surface of the finger. The only way to change ridges is by damaging the dermis. This will permanently alter that part of the ridges, thus creating new unique pattern. [1] [3] [5] [10] [11] [12] [13] [14] [15] [16] [17]

Ridges are created in the fourth month of a baby's development, and for the rest of their life these ridges will remain relatively the same [11]. It is assumed that there were no major injuries. Small injuries, wrinkles, and other effects interrupt ridges, but their continuity and minutiae will remain unchanged. Physiologically, ridges are responsible for better sensitivity to touch and also for a better grip of objects. The height of a ridge ranges from 0.1 to 0.4 mm and their width is around 0.2 to 0.6 mm. [1] [5] [9] [11] [12] [13] [14] [15] [16] [17]

¹ According to Harmonized Biometric Vocabulary (<http://www.christoph-busch.de/standards.html>)

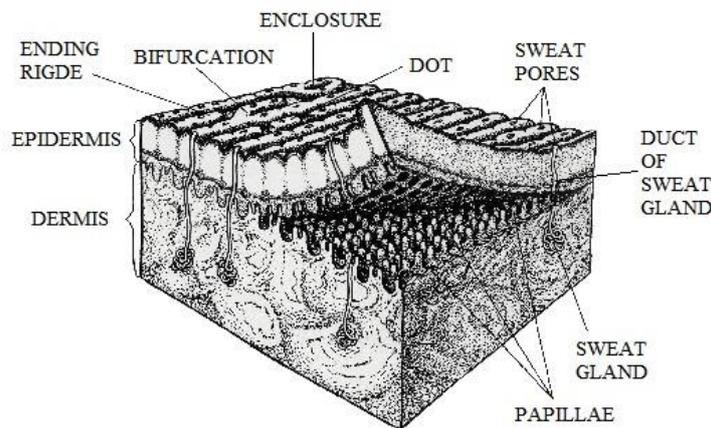


Figure 2.1: Skin Structure (taken and modified from [5]).

2.2.2 Classification of Fingerprints

By simply comparing two images, the identification would be a difficult task, therefore fingerprints can be divided into several particular classes. Using this classification system, it is possible to quickly reject fingerprints from another class, which greatly accelerates the identification. This is necessary in big databases, such as those that the FBI uses. Their system, IAFIS (Integrated Automated Fingerprint Identification System), [18] uses the *Henry's classification system* [5], which contains three classes. These are *arch*, *loop*, and *whorl*. Nowadays, extended versions, where these three classes are split into more specific ones, are used. In Figure 2.3 two subclasses for every class can be seen. In [3], [14], etc., it is possible to find more subclasses usually derived from the whorl class. All these classes are not equally frequent in fingers. Arches are the most unique ones with a probability of around 6.6 %. In the middle, there are whorls in 27.9 % of fingers. The most frequent are loops which can be found in almost two-thirds of all fingers (65.5 %). [1] [5] [9] [11] [19] [A1]

To understand how these classes can be distinguished, it is necessary to define some objects of interest. The first of them is *delta* [5] [6]. It is a place where ridges run in three different directions; it forms a triangular shape. The second of them is *core* [5] [6]. Core is the centre of the fingerprint and it can be found in the innermost loop or in the middle of the spiral in the whorl class. In Figure 2.2 the core is marked with a blue colour, the direction of the core is marked with a red arrow, and the delta is marked with a green triangle. Six classes in Figure 2.3 all differ in the quantity of cores and deltas or in

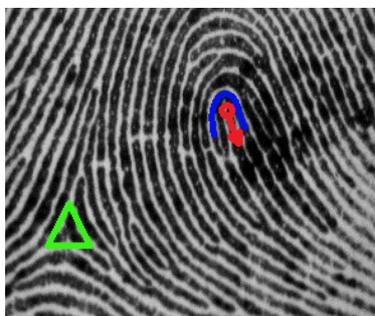


Figure 2.2: Singular points – core and delta.

the direction of the cores. An arch does not have any cores or deltas. A tented arch has one core, one delta, and the direction of the core points to the delta. Loops are like the tented arch, but with a different direction that specifies them. In general, whorls have two deltas and one or two cores. With all this information, the fingerprint in Figure 2.2 can be classified as a right loop. [1] [5] [6] [9] [11] [20] [A1]

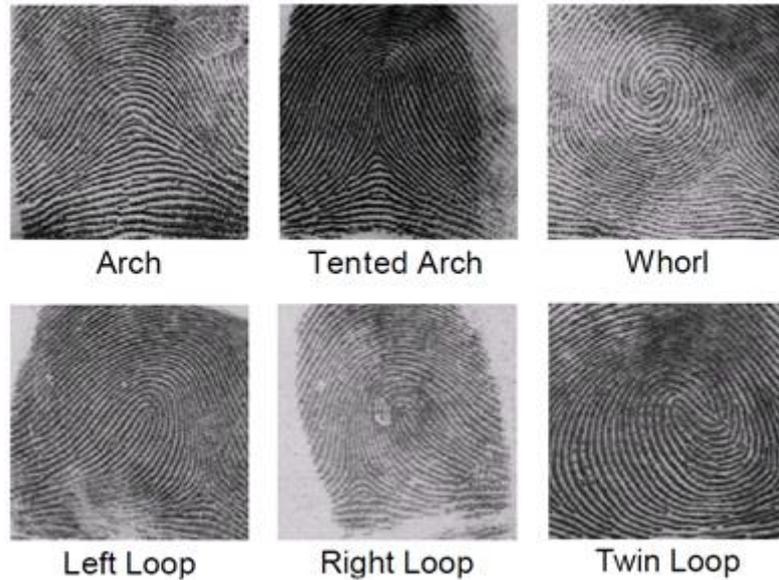


Figure 2.3: Classes of fingerprints (taken and modified from [19]).

2.2.3 Fingerprint Minutiae

Classes alone are not sufficient enough to identify a person. The characteristic that is detailed enough to distinguish every finger in the world is the *fingerprint minutia*. Minutia [5] is a special formation created by ridges. In dactyloscopy huge amounts of these formations are distinguished. Some of them can be seen in Figure 2.4. From left to the right it is [5]: ridge ending, bifurcation, double bifurcation, triple bifurcation, spur (or hook), ridge crossing, opposed bifurcation (or side contact), dot, island (short ridge), enclosure (or single whorl), double whorl, bridge, twin bridge, through line. Each type of minutia has a different likelihood of appearance in the fingerprint. [1] [5] [9] [11] [21] [A1]

Computers can find and save all these types of minutiae to recognize fingerprints, but it is very demanding. The recognition of these complicated patterns will only prolong the fingerprint acquirement. Unlike people, computers do not have a problem with saving greater numbers of minutiae or their location and orientation. For these reasons, in automated processing only two basic types of minutiae are recognized: *ridge ending* and *bifurcation* (in Figure 2.4 marked with a red frame). There are specialized dactyloscopic tools which can recognize more types of minutiae. [1] [5] [9] [11] [22] [A1]

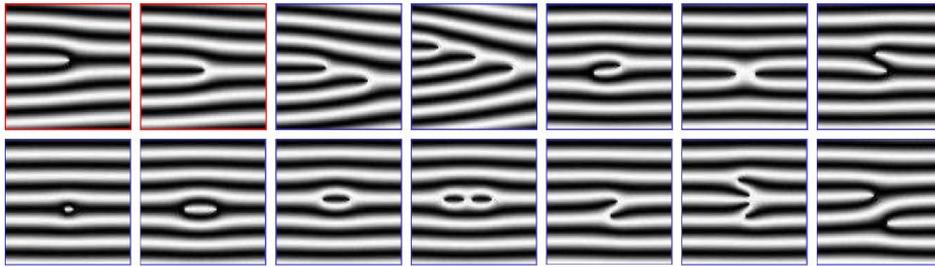


Figure 2.4: Basic types of minutiae (taken and modified from [19]).

2.3 Sensor Technologies for Fingerprint Acquisition

Nowadays, when fingerprint recognition technology is used, regardless of the precise usage (i.e. verification or identification) the first thing to do is to get a fingerprint from the finger to the computer. There are several methods of obtaining a digitalized fingerprint. The traditional dactyloscopic card, where the fingerprint is obtained by moistening the fingertip in ink or a chemical substance (clean fingerprinting), can be scanned. This method leaves fingers dirty and there is no certainty of making a good fingerprint. It is better to have fingers scanned directly into the computer. The principle of these direct methods can be found in the following subchapters. [1] [5] [10]

Fingerprint capturing sensors are divided into three main categories. They are swipe, contactless, and touch (or area) sensors. When using **touch sensors** the finger is placed on the sensor area and left there for a few seconds without moving it. These sensors are very easy to use, even for inexperienced users. The only thing that could go wrong is a bad rotation or position of the finger. A bad rotation often occurs when the thumb is being scanned (20° is usually enough for matching algorithms to stop working). People with longer fingers frequently do not properly estimate the sensor's area, and then the core of the fingerprint is not scanned or appears in the edge of the scan, which is not an optimal position for many matching algorithms. The biggest disadvantage of touch sensors is that latent fingerprints can remain on them. Some technologies can get tricked by the reactivation of the last finger from a latent fingerprint. In this matter, a related problem is that the sensor gets dirty with each scan and must be cleaned, depending on the frequency of scanning. Dirty sensors produce dirty fingerprints, which can result in a higher false rejection rate [10]. A good sensor should also have an area large enough to fit everyone's finger. However, a larger area usually means a higher cost. [1] [10]

Swipe sensors are usually a little bit wider than a finger, but their height is only several millimetres. When using swipe sensors, the finger is swiped vertically over the sensing area. The sensor will then reconstruct the fingerprint from each smaller part captured when the finger was swiped, as can be seen in Figure 2.5. The advantage of this type of sensor is its lower cost, because of the much smaller area. Also, there is no latent fingerprint available (only the last part of it) and finger movement basically cleans the sensor each time it is used. The rotation of the fingerprint, thanks to the vertical movement, is almost non-existent. On the other hand, the sensor is harder to use. There are many things that can go wrong when swiping a finger. The exact speed, position, and steadiness of the movement have to be maintained. In case of the wrong speed or unsteadiness of the finger movement, the final

image is discontinuous or unrealistically long. In addition, when the finger is in the wrong position, the final image is simply only half of a fingerprint. The sensor must be able to scan very quickly to permit a suitable swiping speed. The image reconstruction is time-consuming and it is also a source of inaccuracy and errors in the final image. The first swiping sensor was used with thermal technology, but nowadays the most widely used technology is capacitive or RF capacitive. [1] [10] [23] [24] [25] [26] [27] [28] [29]

The last type of *sensor* is a **contactless** one. These sensors scan ridges even without a finger touching the sensor. Usually they work in a similar way to touch sensors. Because of that, there are no worries of a latent fingerprint, dirt on the sensor, or a bad speed or unsteadiness of the fingerprint movement. On the other hand, the device is usually placed around the whole finger, which implies a higher cost and a lower acceptability. The only thing that is needed is the right position of the finger in the device. That could be tricky because one has to align his/her fingers in three dimensions. [1]

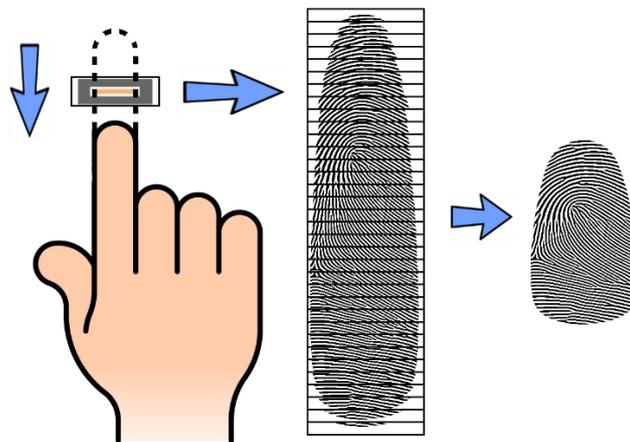


Figure 2.5: Swipe sensor principle.

2.3.1 Optical Technology

Optical fingerprint capture devices are one of the oldest ones; they have existed since the 1970s. They are based on the **Frustrated Total Internal Reflection (FTIR)** [10] principle. Figure 2.6 shows this technology in detail. The finger is placed on the protective glass so that the ridges touch the glass and its valleys are in the distance. The ray from the light source is reflected by the ridges and absorbed (scattered) at the valleys. The reflected rays are channelled through the optics to a charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) camera. The protective glass is illuminated by the light source, as can be seen in Figure 2.6. When the protective glass is replaced by a transparent roller tube and optics, the camera and light source are in it and then a simple swipe optical sensor is designed. It is also possible that such roller functions, like optics or a camera with a light source, are beneath it. Some optical devices utilize contactless technology. These devices work very similarly to primitive photographic devices. The advantages of this technology are that its sensors can withstand temperature fluctuations. They basically operate in three dimensions, so they are more resistant to photograph or fingerprint image attacks. Another type of optical sensor uses **optical coherence tomography (OCT)** [30]. It is very expensive, but it gets the image from a deeper layer of

the skin, which is harder to spoof. It can also obtain an image where the sweat pores are clearly visible. The disadvantages of the optical technology in general is that the sensor is sensitive to dirty fingers and that latent fingerprints are a big problem – with the exception of contactless devices and optical coherence tomography. [1] [5] [9] [10] [15] [24] [29] [30] [31]

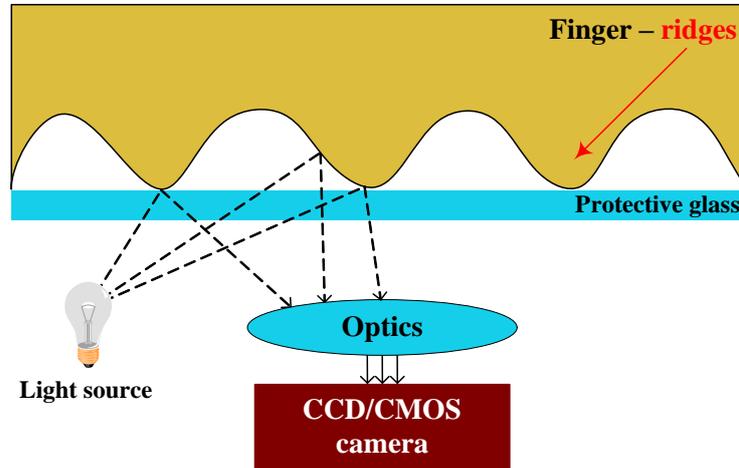


Figure 2.6: Optical technology principle (taken and modified from [19]).

2.3.2 Capacitive Technology

The capacitive sensor is created by a two-dimensional array of a micro-capacitor plate. Ridges and valleys create the second part of these micro-capacitors. In Figure 2.7 the difference between the distances of a ridge and a valley can be seen, because the capacitors have another electrical behaviour, which can be measured. When height of the sensing area is small then it is a swipe sensor. Despite its wide usage, they do have some disadvantages. There is the danger of damaging the whole device when one’s finger is electrostatically charged. There are also chemicals in sweat that can damage the silicon chip. For these purposes there has to be a protective layer, but this layer has to be as thin as possible to have the smallest impact on the measurement of differences between ridges and valleys. [1] [5] [9] [10] [22] [29]

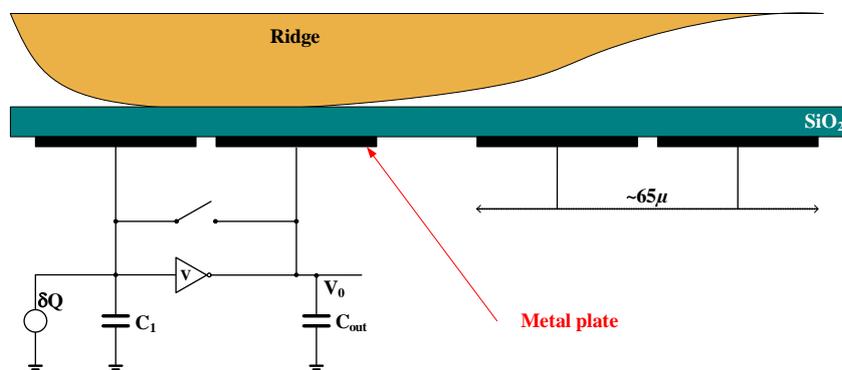


Figure 2.7: Capacitive technology principle [19].

However, there is one modification of the capacitive sensor that is worth mentioning. It is a combination of e-field and capacitive technology. It uses a low radio frequency signal and because of that, it is often known as **radio frequency (RF)** technology. This signal is sent to the skin, and due to that, an electrical RF field is created between the signal reference plane and the live (conductive) layer of the skin. Its equipotential contours mimic the shape of the live layer of the skin, so when it is measured by the antennae array a fingerprint image is acquired. This principle can be seen in Figure 2.8. Another interesting principle is using thin-film transistors on a capacitive touch panel. That way it is possible to have a touch sensitive area (i.e. a smartphone can be controlled this way) and fingerprint sensing at the same time. [9] [24] [29] [31] [32] [33]

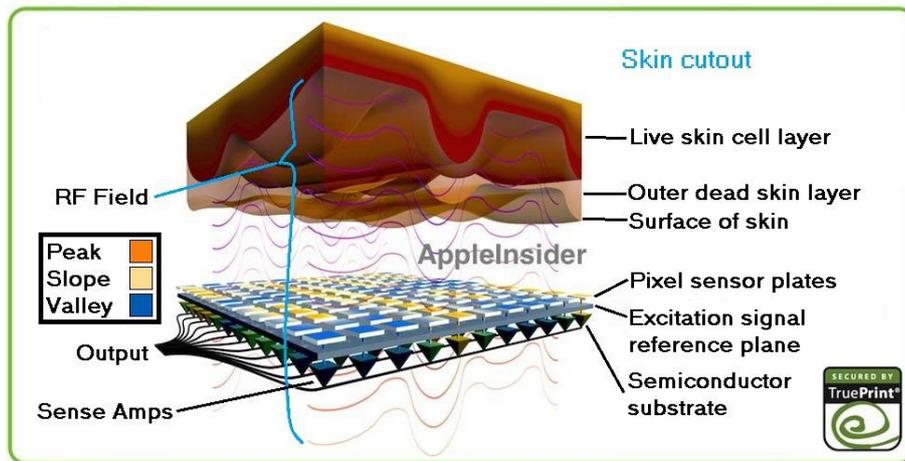


Figure 2.8: An example of the RF capacitive technology principle (taken and modified from [32]).

2.3.3 Thermal Technology

Thermal technology is based on differential thermal radiation. Pyroelectric materials generate the current according to various temperatures. Ridges have higher thermal radiation than valleys, so they have a higher temperature. Since temperatures quickly equalize, it is necessary to use swiping sensors, as can be seen in Figure 2.9. Despite quick equalization, there is a company (NEXT Biometrics) that is using the heat pulse and provides a touch thermal sensor² [34]. The main advantage of thermal technology is that it is very resistant to electrostatic discharge. The protective layer can be very thick as well. [1] [5] [9] [10] [29] [31]

² https://www.nextbiometrics.com/products/fingerprint_sensor_modules/nb-2023-s2-v-and-nb-2023-u2-v/

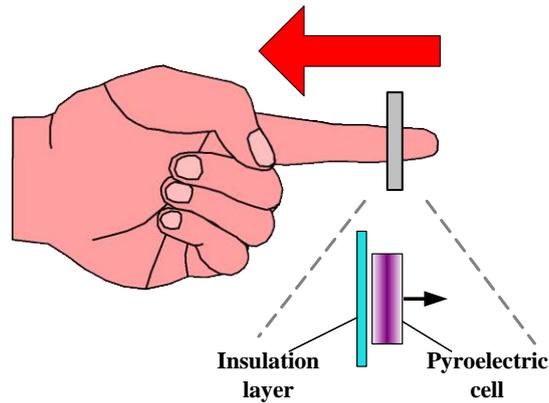


Figure 2.9: Thermal technology principle [19].

2.3.4 Ultrasonic Technology

Ultrasonic capture devices consist of a transmitter and a receiver. The transmitter sends acoustic signals, which are reflected by the ridges (skin) and valleys (air) differently. The transmitter and the receiver move around the finger as it is shown in Figure 2.10. The receiver then receives echo signals, and thanks to the different acoustic impedance, measures the distance and consequently acquires an image of the fingerprint. The frequency used by these sensors is between 20 kHz and several GHz. Higher frequencies help to obtain a higher resolution. Ultrasonic sensors have one of the best image quality and accuracy rates (ten times better than any other technology). The ultrasonic technology penetrates the upper part of the skin, which results in the better detection of spoofed fingers and it is also less influenced by dirt on the fingers, surface damage, and dirt on the sensors. The main disadvantages are the very high cost and the large size of the device. Another problem is also that the ultrasonic technology cannot operate properly at low temperatures. [1] [5] [10] [24]

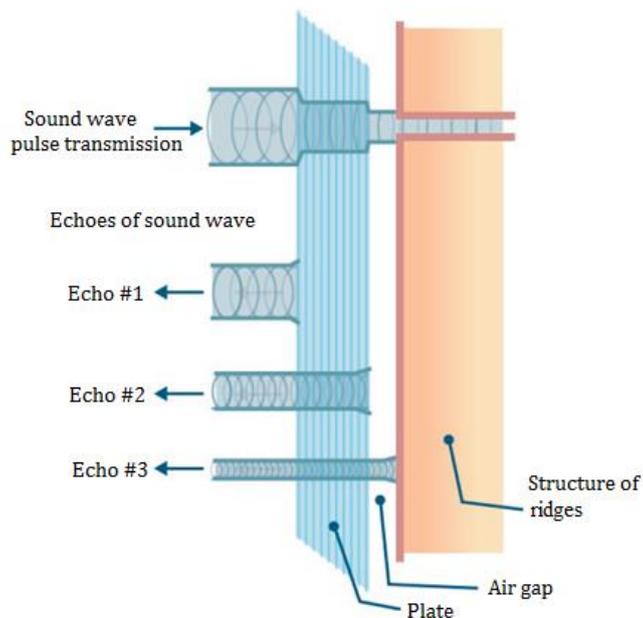


Figure 2.10: Ultrasonic technology sensor movement (taken and modified from [19]).

2.3.5 Pressure Sensitive Technology

The pressure sensitive (or piezoelectric) sensor is composed of three layers. There is a non-conductive gel added between the electro-conductive layers. The whole sensor, with the finger ready to scan, is shown in Figure 2.11. The non-conductive gel is pressed by the finger ridges, which causes the electro-conductive layers to touch. The sensor then measures the current created by the finger and creates an image of the fingerprint from it. The protective layer creates a blur on the whole fingerprint. Also, materials have to be sufficiently sensitive to detect the differences between valleys and ridges. [1] [5] [6] [9] [10]

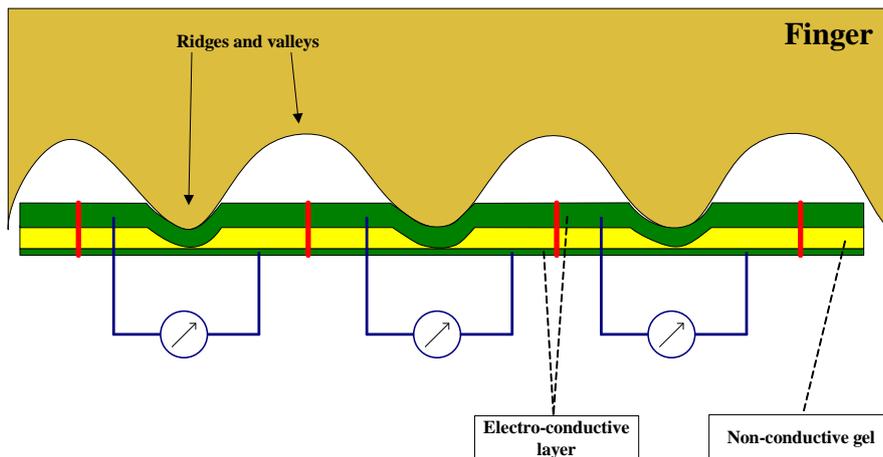


Figure 2.11: Pressure sensitive technology principle [19].

2.3.6 E-field Technology

With this technology, the sensor consists of a drive ring and a matrix of antennae. The drive ring generates a sinusoidal radio frequency signal and the matrix of active antennae receives that signal modulated by the skin's structure or, more precisely, modulated by the dermis structure, because the electric field passes through the upper parts of the skin (the epidermis). Similar to the ultrasonic technology, this technology is also resistant to fingerprint spoofs and ignores the dirt and light injuries on the finger. The image quality here is better than the one from capacitive or electro-optical sensors. The disadvantage is that the sensor is very sensitive to electrostatic charges and there is the possibility of a sensitivity to disturbance in its RF modulation. [1] [6] [9] [10]

2.3.7 Electro-Optical Technology

The electro-optical sensor consists of four layers, which are clearly shown in Figure 2.12. There is an isolation layer, a black coaxial layer, a light-emitting layer, and a basic layer. Underneath there is a CCD/CMOS camera. The light-emitting layer is made from a polymer that emits light when polarized with the proper voltage. When ridges touch the sensor it causes the black coaxial layer to touch the

phosphor layer, which then emits light in the places of ridges. This light passes the basic layer and then a camera captures it. [1] [5] [6] [9] [10]

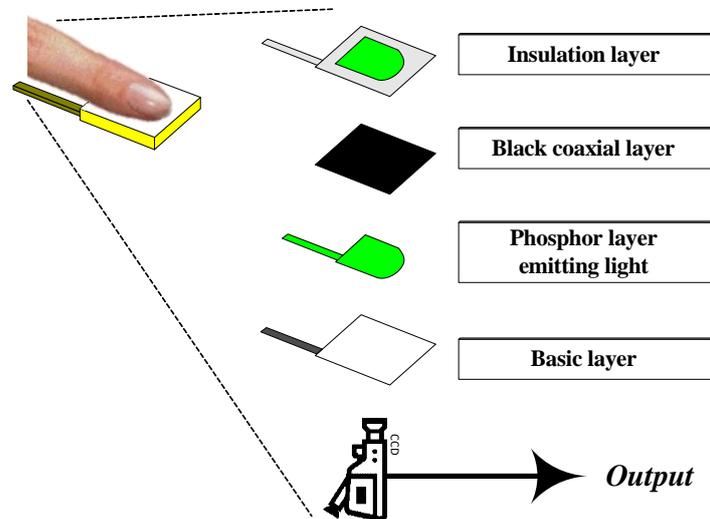


Figure 2.12: Electro-optical technology principle (taken and modified from [19]).

2.3.8 MEMS Technology

The MEMS (Micro-Electro-Mechanical-System) [10] uses micro parts to scan a fingerprint. One of the methods uses piezo-resistive micro beams. The user swipes his/her finger along the sensor, which consists of three rows of piezo-resistive gauges. Their parallel deflection will create a voltage variation which is measured and transformed into the fingerprint. The resulting image is only binary-coloured, which is the big disadvantage of this type of technology. This pressure-based MEMS swipe sensor principle can be seen in Figure 2.13. Another method is to use micro-heaters. This method slightly heats the finger and measures the temperature changes of the heating element. A ridge works as a heat sink so that the heat element, which is connected to the ridge, shows a lower rise in temperature. [1] [6] [9] [10] [29] [31]

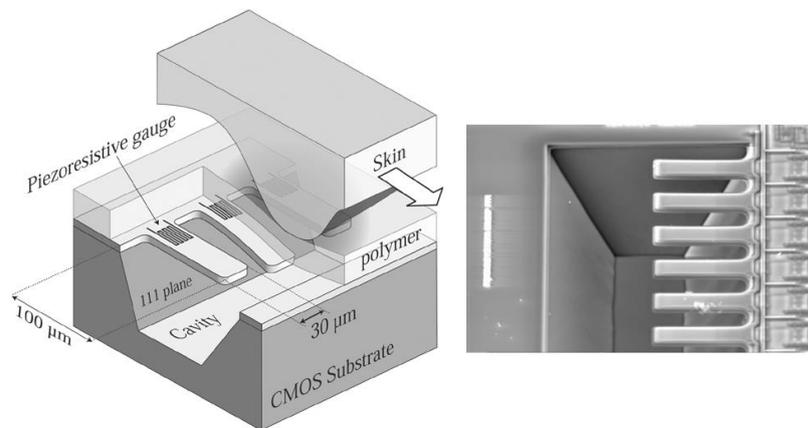


Figure 2.13: Tactile MEMS technology principle (taken and modified from [35]).

2.4 Fingerprint Recognition Process

How to obtain a digitized fingerprint has been explained, but there is still one process yet to be described – the process of recognizing the fingerprint. In Figure 2.14 an overview of this process can be seen. First, a digitalized image of a fingerprint is needed. Nowadays, sensors tend to have liveness detection (anti-spoofing) as a part of the scanning process. The next phase is the enhancement of image quality. In each point of the image, including its surroundings, the direction of the ridges is counted. If this point is on the ridge, it determines (with a high probability) the direction of it. This phase can be divided into smaller ones – the orientation field estimation for each point, the estimation of the block orientation field, and then the final mapping on the original image. Using this information, the image is then enhanced. In this step many various methods can be applied on the image. Usually, the method for adjusting the histogram is used. Image quality enhancements are used, like the Gabor filter, frequency filters (after using FFT) such as the Butterworth filter or the Ikonopoulou filter, etc. (after the application of the filter IFFT is used). [1] [5] [9] [10] [14] [A1]

The next step is binarization. It is usually done by some thresholding method, e.g. by regional average thresholding or by adaptive thresholding. At the end of this step is a binary image, where ridges are black and valleys white. The following process is minutiae detection, and for this purpose only ridges are needed. So in this step the ridges are thinned to be only one pixel wide. The only problem is that the ridges should not decline in any direction – that could cause a problem with the precise minutiae position. The last phase is minutiae detection and extraction. Specialized algorithms are used for this purpose, one of them being the Hong method [5]. In this phase minutiae are detected (in verification systems usually only ridge ending and bifurcation) and its properties (position, type, and gradient) are extracted. After that, different approaches for recognition could be used; for example, global and local minutia alignment, minutia cylinder-code, etc. [36]. [1] [5] [9] [10] [14] [A1]

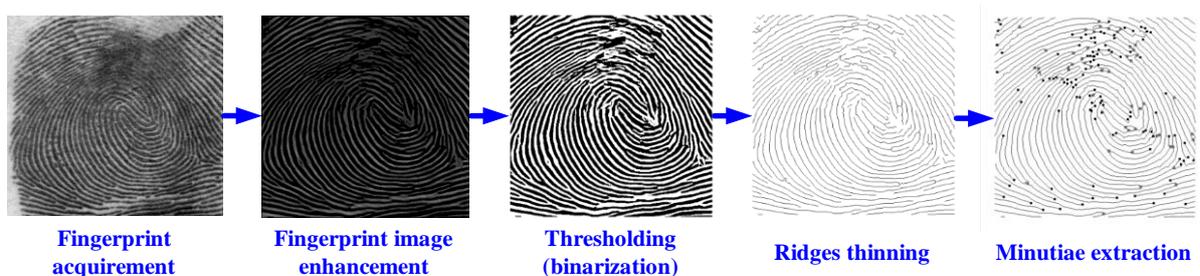


Figure 2.14: An overview of fingerprint recognition process (taken and modified from [19]).

3 Synthetic Fingerprint

Fingerprint recognition technology is being used more and more in this day and age. Along with it, many methods have come into light that make fingerprint recognition more resistant to impostors. The amount of various recognition algorithms is greater, too. These algorithms need testing and are usually tested on small databases. Larger databases (datasets with thousands, or even better, tens of thousands of fingerprints) are very hard to get because making them is very time consuming and expensive. It demands a very well-trusted organization to attempt to collect a database like that, because people tend not to give out their fingerprints to anyone. Collecting such a database is also very tiresome, both for the technician and the users. In this monotonous environment it is easy to make a mistake. Even when such a large database is available, there are usually problems with sharing it because of privacy legislation that protects this type of data. When these databases are not available, algorithms are tested on smaller databases and it is very easy to make them data dependent. So they are very accurate when it comes to a common fingerprint (e.g. loop class), but with an extraordinary fingerprint (like twin loop class) their accuracy collapses. [1] [6] [37] [38]

In these cases it would be great to have some generator (application) that would create a large synthetic fingerprint database. If a synthetic database consists of images very similar to human fingerprints, then that can be used instead of a large database of real fingerprints. It also opens up the possibility for testing a specific type of fingerprint or adapting algorithms to them. This is possible because fingerprints that resemble those from specific workplace environments can be generated. Generating such a database would save a lot of resources (human, money, time) that can be used to create better algorithms. So this is the motivation for creating synthetic fingerprints. [1] [6] [37] [38]

3.1 Methods for Generating Synthetic Fingerprints

Synthetic fingerprint generation is an inverse biometrics problem [39]. According to input variables, it is essentially the fingerprint recognition process (Subchapter 2.4) from the end to the start. Several methods of how to generate a synthetic fingerprint can be found in [6] [11] [21] [40] [41] [42] [43], and when these methods are thoroughly studied, one can find that they are all based on the same principle. The method used by the SFinGe³ seems to be the oldest one and also the most commonly known, so it will be described as a template for others. For example, very similar methods are used by Anguli, which is an Indian Institute of Science fingerprint generator. [1] [16] [44] [A1]

To gain a better understanding, the upper part in Figure 3.1 (figure is located at the end of Subchapter 3.3) shows the generation process. The generating part ends with the so-called master fingerprint (a perfect fingerprint, equivalent to the phase extracted lines from Figure 2.14). First, the fingerprint's shape is determined. The basic shape is oval and each elliptical segment can be changed

³ <http://biolab.csr.unibo.it/research.asp?organize=Activities&select=&selObj=12&pathSubj=111%7C%7C12&>

to create the required shape. The second step is the directional field model. In this step the fingerprint class is chosen together with the position of cores and deltas. This step uses the Sherlock and Monroe ridge [6] flow model to generate a consistent direction field. The third step creates a density map. If the fingerprint is thoroughly examined, it can be seen that the density of ridges is not the same throughout the whole area. After examining several real fingerprints some heuristic criteria could be made. These criteria are based on the position of singularities (cores and deltas) and according to them the density map is generated. The last step is ridge pattern generation. This phase uses all previous steps along with some initial seeds. Iteratively, the image with the initial seeds is refined with the Gabor filter. The filter orientation and frequency are adjusted according to the directional field and density map. Minutiae are automatically generated at random places with random types (dactyloscopic ones, not only ridge ending and bifurcation). After that phase, the master fingerprint is finished. [1] [11] [16] [21] [40] [44] [45] [A1]

As can be seen, the SFinGe generating process is not exactly an inverted recognition process. If this process is strictly followed, so-called fingerprint reconstruction is then performed. These are methods that focus on the creation of a whole fingerprint from only the minutiae saved as a template in fingerprint recognition [46] [47]. Another method lies between these two. It states that fingerprint features are dependent on each other [21]. It follows the same scheme, but with dependencies on the other steps. The orientation field is influenced by singular points. The minutiae density is higher around singularities and their appearance is not random, but rather statistically driven. The minutiae direction is also dependent on their types and on the orientation of the ridges around it. This method firstly determines singular points, after that it is the orientation field, and finally the minutiae. Each step is dependent on the previous one. After all of the steps are completed, the master fingerprint is made with the use of the AM-FM (amplitude modulation, frequency modulation) method. [1] [16] [21] [42] [44] [A1]

The last described method (from SyFDaS generator) uses minutiae as an input. The creation of a whole fingerprint is based on only these minutiae. The biggest difference is that the orientation field is generated from minutiae and not from classes or singular points as it was in the previous methods. It is generated from the minutiae direction and each minutia has a weight based on the distance of it from the point where the orientation field is determined. The disadvantage of this method is that the final fingerprint could have a class that does not exist in the real world. In this method, the density map can be manually changed. The default state is the uniform density map. After that, using a similar method of Gabor filter as in SFinGe, a master fingerprint is generated. Note that instead of the initial seeds, this method uses minutiae as these seeds and the generation starts with them, so precisely defined minutiae do not change in the generation process. [1] [11] [16] [44] [A1]

3.2 Phenomena Influencing a Fingerprint

This chapter tries to sum up all the phenomena that can influence a fingerprint. This information is needed in order to fully revert from the master fingerprint (i.e. the final stage of the extracted lines phase in the fingerprint recognition process) to a realistic looking fingerprint (i.e. the acquired fingerprint phase of the recognition process). There are three main groups of phenomena that can

damage the quality of a fingerprint. They are finger condition, sensor condition, and environment. The influencing factors connected to the user and his/her finger will be described in the following text. [1]

Almost all fingerprint scanners are influenced by **dirt on the finger**, be it a small particle, a few grains of dust, or simply an oily finger. Conductive materials and liquids are usually the most problematic types of dirt. Only ultrasonic, contactless, and e-field technologies are resistant to this type of damage. The **dry** or **moist finger** is one of the most typical cases of damage done to a fingerprint. Whether it is because the users wash their hands, if they are nervous and their fingers are sweating, or if they have very dry hands and lotion was applied, skin resistance can increase or decrease to ten times the normal value. This usually plays a huge role in the recognition of optical, capacitive, and e-field sensors. The **physical damage of a finger**, such as cuts or abrasions, is obviously damaging to a fingerprint. There is a combination of physical damage and non-cooperative behaviour, which is often called **altered fingerprints** [48] [49] [50]. This category includes surgeries that alter or replace ridges, intentional cuts, mutilation by acid, attempts to change fingerprint class, or scorching. If it is not a deep wound that permanently influences the ridges, there are ultrasonic and e-field technologies that scan the finger in the deeper (dermis) layer where the fingerprint is undamaged. There are numerous **skin diseases** [12], but it is hard to tell how many people are affected by these. There are skin diseases that can change ridges. In these cases, only the ultrasonic and the e-field technology can reconstruct the original fingerprint from that user. And if the disease is severe enough to damage the dermis structure of ridges, there is no way of obtaining the original structure. Skin diseases are explained further in Chapter 6. **Pressure** can turn the fingerprint into a big black oval. Only contactless sensors are fully immune to pressure damage. Optical, ultrasonic, and e-field technologies are also resistant to this type of damage. The change of pressure, a very big or a very low pressure, is also considered to be part of the next category: non-cooperative behaviour. All these activities lead to very thick, thin, or blurred images. The **non-cooperative behaviour of the user** is typical when the user dislikes biometric technology or simply tries to find the limits of its functionality. The user usually exerts unexpected pressure, moves when the device is scanning, or places the finger in the wrong place or with a wrong rotation. None of these technologies are fully resistant to this type of behaviour. The **contact region** is a phenomenon which occurs when the user intentionally or unintentionally presents their finger to a sensor in such way that only a part of it can be acquired. [1] [9] [10] [15] [16] [17] [44] [48] [49] [50] [51] [52] [53] [A1] [S1]

Another group of factors affecting the fingerprint images are those connected to the sensor. **Dirt on the surface** has the same effect as dirt on the finger. The problem is that it affects everyone who uses that device. Therefore, in the registration phase, it can create a common error for every user and there is a danger that these users will not be able to be identified after cleaning up the device. Apart from fingers, there are other things that can pollute sensor area: metallic dust, wooden dust, earth dust, fine sand, or excrement (in outdoor use). These could be on fingers as well, but they are easily pictured on the sensor. In addition to ultrasonic and e-field technologies, every swipe sensor is also more resistant to this type of damage. The **latent fingerprint** is closely related to the previous topic. In some way it is a type of dirt on the surface of the sensor. More than damaging a new fingerprint, there is also a security hazard. These latent fingerprints can be copied or reactivated to breach the biometric device. The technologies that are resistant to latent fingerprints are the same as those in the previous topic. **Physical damage** is an extreme but possible influencing factor of the resulting fingerprint. There is no

easy way to prevent the sensor from getting damaged. The damage of the sensor will have different effects on every technology. In optical technology, for example, a glass crack can be seen in the fingerprint. **Sensor technology** itself has a large impact on how the fingerprint looks (there are a lot of things that could go wrong [54]). For instance, some technologies like ultrasonic or optical tomography can access an image from a deeper level of skin and the resulting image is then shown without shallow scars. [1] [9] [10] [15] [16] [17] [44] [51] [52] [53] [A1] [S1]

The last category of influencing factors are those that can be found in the surrounding environment. **Vibration** in some degree is not a problem, but when the vibrations have a high amplitude they can unfasten some internal components, causing the device to break down. Sometimes they can also slightly change the position of a finger. This movement, as it was described in the user influencing factors, can blur the fingerprint. The **temperature** can be different for the sensor, the finger, and the environment. Typically there are no problems, with the exception of thermal technology. Taking into account extreme temperatures, it is possible to have very dry or very moist fingers which can affect the resulting image. It is also known that the ultrasonic technology does not operate properly in extremely low temperatures. **Surrounding light** only affects optical and electro-optical technologies because they have a light-sensing unit. The sensor area is usually small in order to keep the cost of the sensor low. Therefore, the finger covers it and there is no problem with the surrounding light. However, when the sensor area is larger, or if the finger of the user is smaller or a smaller finger like a pinkie is used, or if contactless technology is used then the influence of the surrounding light can be immense. **Electromagnetic radiation** is an influencing factor that affects every technology. The device as a whole can be influenced by electro-magnetic radiation. Wires inside and outside the sensor connecting it to the other parts of the biometric system, as well as all electronic components, can be influenced by electro-magnetic radiation. Some devices will, for example, create a blurred image. [1] [9] [10] [15] [16] [17] [44] [51] [52] [53] [A1] [S1]

3.3 SFinGe

SFinGe (*Synthetic Fingerprint Generator*) [55] is an application for synthetic fingerprint generation implemented at the University of Bologna, Italy. It is currently in its 5.0 version. The fingerprint database generated from different versions of SFinGe was one of the four databases of the FVC (*Fingerprint Verification Contest*) [6]. Each year (2000, 2002, 2004 and 2006), contestants had similar results in synthetic and real fingerprint databases. This implies that SFinGe has the inter-class and intra-class variation of synthetic fingerprints very similar to real ones. [1] [45] [55]

The process of fingerprint generation is shown in Figure 3.1. The upper part, i.e. the part that ends with the generated master fingerprint, is described in Subchapter 3.1. For a more realistic looking fingerprint, certain damage simulation methods are applied. These are in the lower part of Figure 3.1. The first step is the selection of the contact region. To simulate the different placements of the finger on the sensor area a random translation of the ridge pattern is made. This is done without modifying the global fingerprint's shape and position. The next step is the variation in ridge thickness. Ridge thickness is modified to simulate various skin dampness and finger pressure. Wet skin and higher pressure cause ridges to appear thicker, and in that case an erosion operator is used. Dry skin and lower pressure make

these ridges thinner, so in this case a dilatation operator is required. A randomly selected magnitude of dampness and pressure determines which square box will be used and also which morphological operator will be implemented. The next phase is fingerprint distortion. In this phase, skin deformation according to different finger placements over the sensor is simulated. The skin plasticity (compression or stretching) and a different force is applied to each part of the finger to create a non-linear distortion. Lagrange interpolation is used for this distortion. The next step is noising and rendering. In this step many small factors are simulated. Unfortunately, these small factors damage fingerprints the most. These include irregularities in the ridges, non-uniform pressure of the finger, different contact of ridges with the sensor, presence of small pores, and other noise. Noise is generated in four substeps. First, valleys (or white pixels) are saved separately. Second, noise in the form of various types of stains is added. Third, the whole image is smoothed with 3×3 windows. Finally, valleys saved in the first step are returned back to the image (to prevent excessive smoothing in the third step). Another phase is global translation or rotation. This phase simulates an imperfectly placed finger on the sensor, so it slightly translates and/or rotates the whole image. The last step is the generation of a realistic background. There is a set of background images and a mathematical method based on the principle of “KL transform” [40]. Applying this method to the set of images creates new background images. At the end of that step, the fingerprint impression is made. For the generation of databases, several impressions are made from one master fingerprint. [1] [6] [16] [21] [40] [45] [A1]

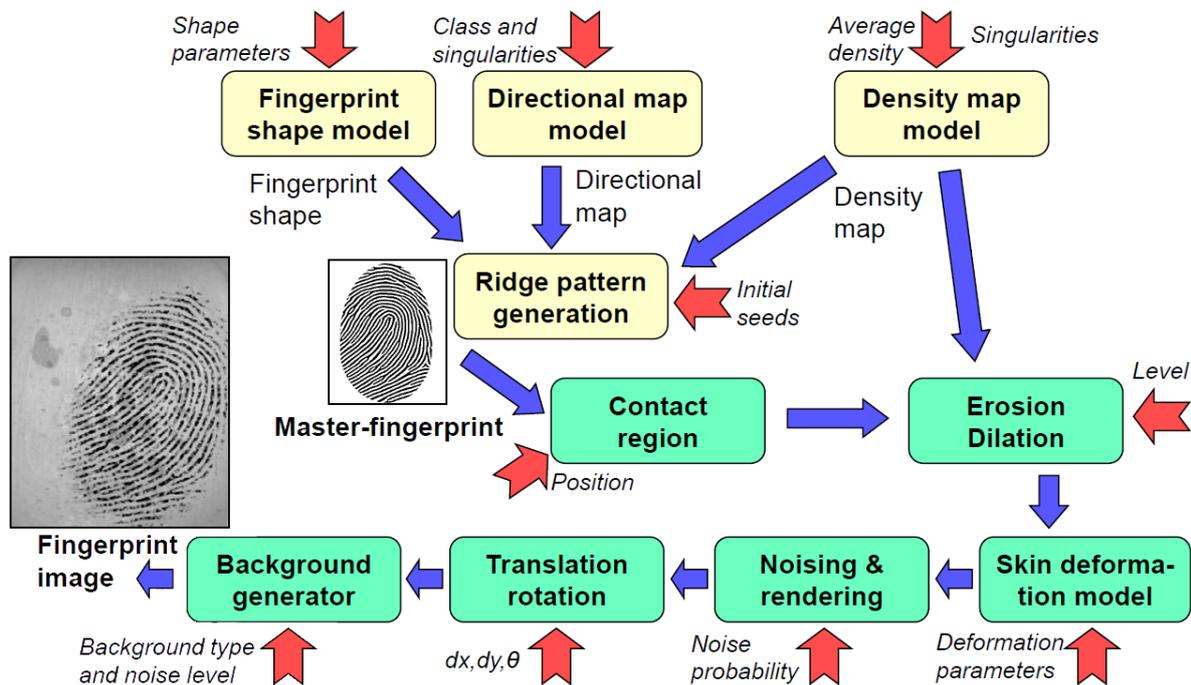


Figure 3.1: SFinGe process of fingerprint generation (taken from [40]).

4 SyFDaS – Synthetic Fingerprint

Damage Simulator

When designing an application for damaging synthetic fingerprints, it was determined to first create a simulation of this application using P/T Petri nets. This simulation is called Fingerprint Generation Petri Net (FGN). After that, the application is described; the primary focus is laid upon the core design and graphical user interface. In this chapter, the enhancement of the generator is described. The biggest portion of this chapter is dedicated to touch-based damages. Lastly, database generation options are discussed.

4.1 Fingerprint Generation Petri Net

Petri nets [56] are specific modelling techniques. They can be defined either by graphs or by a purely mathematical notation. The graphic notation is usually easily understandable, while the mathematical one can be used for various analyses and proofs. As the Petri net is a very old technique, it can be better viewed as a group of various techniques with a similar basis. For the purposes of this article, the P/T Petri net [56] [57] will be used. Petri nets are primarily used in distributed and discrete systems. [44] [A1]

The definition of Petri net is a sextuplet $N = (P, T, F, W, K, M_0)$ where: P, T, F denote places, transitions, and arcs, respectively, $P \cap T = \emptyset, F \subseteq (P \times T) \cup (T \times P)$ is a binary relation, P, T, F are finite, $W: F \rightarrow \mathbb{N} \setminus \{0\}$ is the weight of each arc, $K: P \rightarrow \mathbb{N} \cup \{\omega\}$ is the capacity of each place, and $M_0: P \rightarrow \mathbb{N} \cup \{\omega\}$ is the initial marking so that $\forall p \in P: M_0(p) \leq K(p)$. Note that ω is the supremum of the set \mathbb{N} with these properties: $\forall n \in \mathbb{N}: n < \omega$ and $\forall m \in \mathbb{N} \cup \{\omega\}: m + \omega = \omega + m = \omega - m = \omega$. The graphic representation of the Petri net could look like the one in Figure 4.1. Circles denote places, full rectangles denote transitions, and arrows denote arcs in their direction. $P0, P1$, and $T0$ are labels of each place or transition, numbers above arcs represent weights, and the small dot inside place $P0$ denotes a token. The equivalent notation of the Petri net in Figure 4.1 is $N = (\{P0, P1\}, \{T0\}, \{(P0, T0), (T0, P1)\}, \{W(P0, T0) = 1, W(T0, P1) = 1\}, \{K(P0) = \omega, K(P1) = \omega\}, \{M_0(P0) = 1, M_0(P1) = 0\})$. [44] [56] [57] [58] [59] [A1]

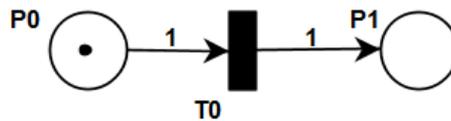


Figure 4.1: An example of the graphical representation of Petri net.

With the structure of the net defined, the only definition missing is that of the transition firing. This definition can be better described using a preset and a postset. The preset $\bullet x$ is defined as $\forall x \in (P \cup T): \bullet x = \{y / yFx\}$ and the postset x^\bullet is defined as $\forall x \in (P \cup T): x^\bullet = \{y / xFy\}$. Similarly, for sets $\bullet X$ is the union of $\bullet x$ for each x in X and X^\bullet is the union of x^\bullet for each x in X . Note that the marking M is defined by the same definition as the initial marking M_0 . The transition $t \in T$ is enabled from the marking of M when these conditions are met: $\forall p \in \bullet t: M(p) \geq W(p, t)$ and $\forall p \in t^\bullet: M(p) \leq K(p) - W(t, p)$. When this transition is fired, one gets the marking of M' , which is defined in this way (Eq. 4.1) [56] [57] [58] [59]: [44] [A1]

$$\forall p \in P: M'(p) = \begin{cases} M(p) - W(p, t) & \text{if } p \in \bullet t \setminus t^\bullet \\ M(p) + W(t, p) & \text{if } p \in t^\bullet \setminus \bullet t \\ M(p) - W(p, t) + W(t, p) & \text{if } p \in \bullet t \cap t^\bullet \\ M(p) & \text{otherwise} \end{cases} \quad (\text{Eq. 4.1})$$

The result of the transition $T0$ firing from the net shown in Figure 4.1 is demonstrated in Figure 4.2.

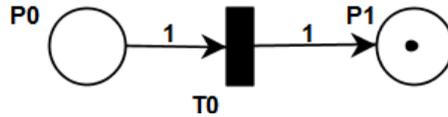


Figure 4.2: An example of the graphical representation of the firing of the transition $T0$ from Figure 4.1

4.1.1 Design of the Fingerprint Generation Petri Net

The goal of this subchapter is to create a Petri net that will simulate the generation of a synthetic fingerprint. The Petri net that will be used is defined in Subchapter 4.1 and the fingerprint generation process is thoroughly described in the previous Chapter 3. Petri nets are often used to simplify complex distributed systems. This is a similar case – it is important to know that the Petri net only simulates the fingerprint generation process. It is advantageous to use them as a clear way of showing all of the possibilities in synthetic fingerprint damage simulation. They can also be used to show specific scenarios. The scenario can then be recreated in software with all the necessary damage simulations implemented. These are the main reasons for choosing Petri nets. [44] [A1]

To make Petri net creation clearer, the generation process will be divided into four distinct parts. The first part is the master fingerprint generation, the second part simulates the state of the environment, the third part simulates the user and finger condition with the respective fingerprint damage, and the last part simulates the sensor conditions that affect the fingerprint. [44] [A1]

The process starts with the place *Start* where the initial token is placed, as can be seen in Figure 4.3 (images are generated from the PIPE⁴ software which exports raster images). After firing the *Input definition* transition, four other transitions are enabled. *Class input* defines (fires the transition) *Class* and then *Cores and Deltas* are defined. After that, the first token is in the *All input data* place. To create the synthetic fingerprint, the *Gabor filter* transition needs to be used. But that is possible only

⁴ <https://github.com/sarahtattersall/PIPE>

when four tokens are in the *All input data* place. Another three tokens get here by defining *Shape*, *Density*, and *Initial Seeds*. Then the *Gabor filter* is fired and the token in *Synthetic Fingerprint* shows that the generation of a master fingerprint is complete. [44] [A1]

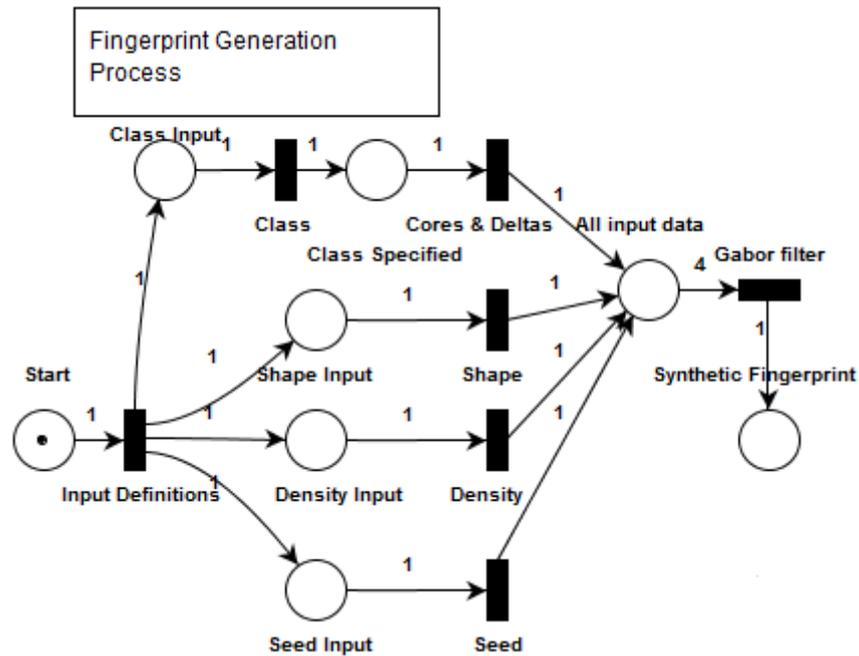


Figure 4.3: The first part of the proposed Petri net – the fingerprint generation process.

The second part is focused on influence of the environment and is shown in Figure 4.4. The initial marking is set to each of the environmental phenomena. After the firing, a value representing each phenomenon should be prepared. After that, all of these values should be saved in the structure representing all phenomena at once, in the place named *All Environmental Effects*. [44] [A1]

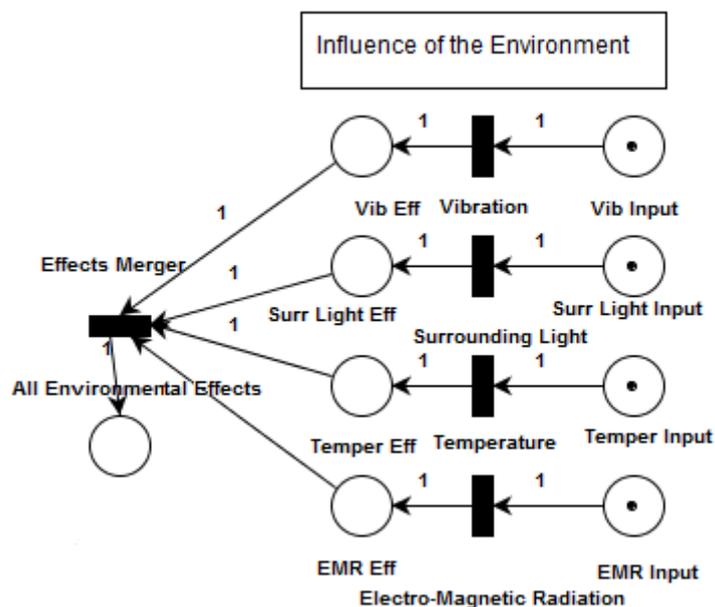


Figure 4.4: The second part of the proposed Petri net – the influence of the environment.

The third part of the net simulates the user and finger phenomena and is represented in Figure 4.5. Transitions in the middle are individual phenomena that are connected to the user or the finger. The places near the transitions have two purposes. First, they enable or disable this type of damage. Second, they ensure that each phenomenon will be used only once. Also, notice that the places *Synthetic Fingerprint* and *All Environmental Effects* are the last places of the previous parts. When these previous parts are done, the process continues after the application of 0 to n damages to the synthetic fingerprint. When the damage is done, transition *User and finger damage done* can be fired and token get to *Synthetic fingerprint with user and finger phenomena and effects of the environment* place. This situation can be seen in Figure 4.6. Note that the place *Return* with the transition *Return of Synthetic Fingerprint and Environmental Effects* are returning the damaged synthetic fingerprint and the information about the environment so that 0 or n damages can be done. [44] [A1]

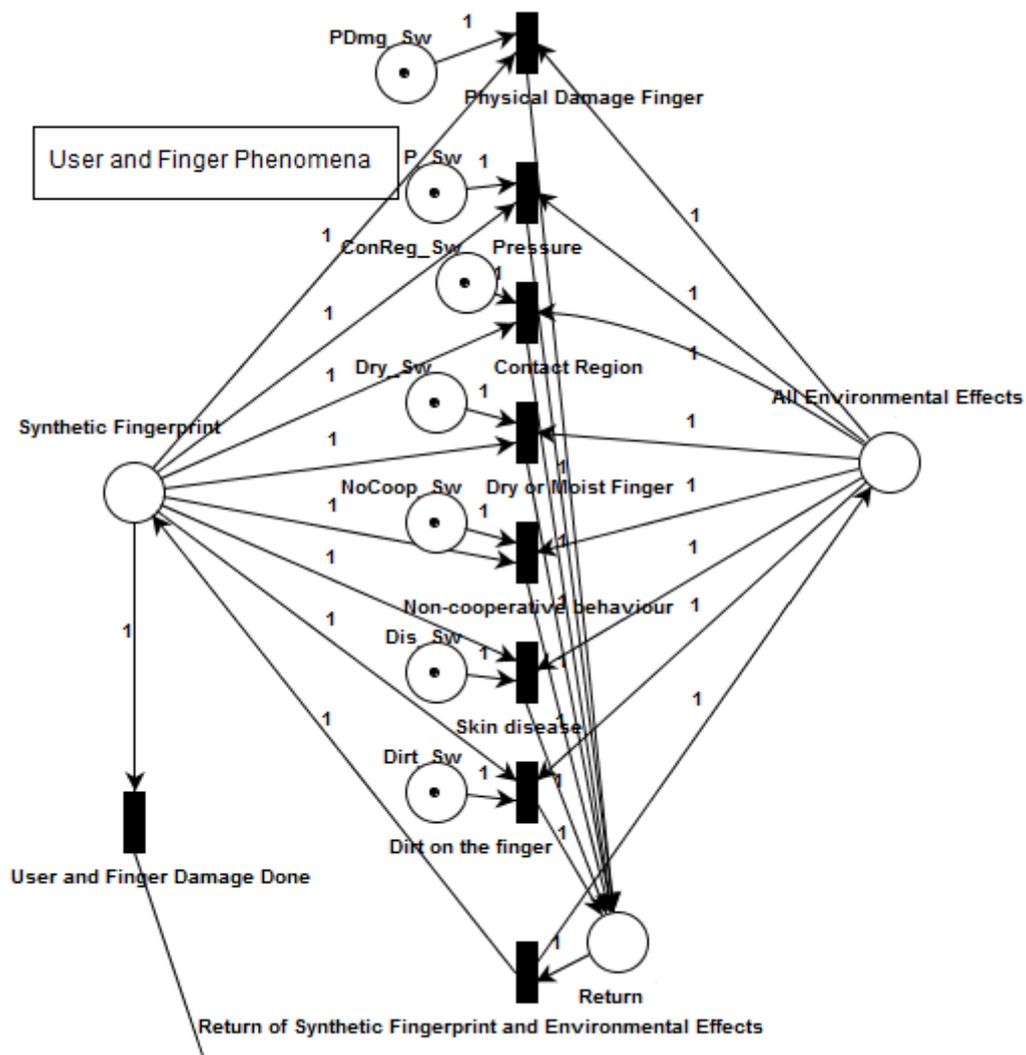


Figure 4.5: The third part of the proposed Petri net – the user and finger phenomena.

The last part starts where the previous one ended and it is focused on the sensor condition. As can be seen in Figure 4.6, there is a similar structure as in the previous part. Using the damaged fingerprint from the previous steps and the same structure of the environment values, it further damages the fingerprint image. Individual conditions of the sensor can be switched on or off and are used only once by the switch places near them. From the *Return 2* place, the token goes by transition back to *All Environmental Effects* and another token representing the generated fingerprint also goes back to its place. Again, the *Sensor Damage Done* transition is fired after 0 or n damages and the synthetic fingerprint generation is complete. One part of this final step is taking the environment into account. In the rare case when the user, finger, and sensor are all in perfect condition then this step is the only way for the environment to influence the final synthetic fingerprint. Generally speaking, the environment influences other damages more often than it has an effect on its own. [44] [A1]

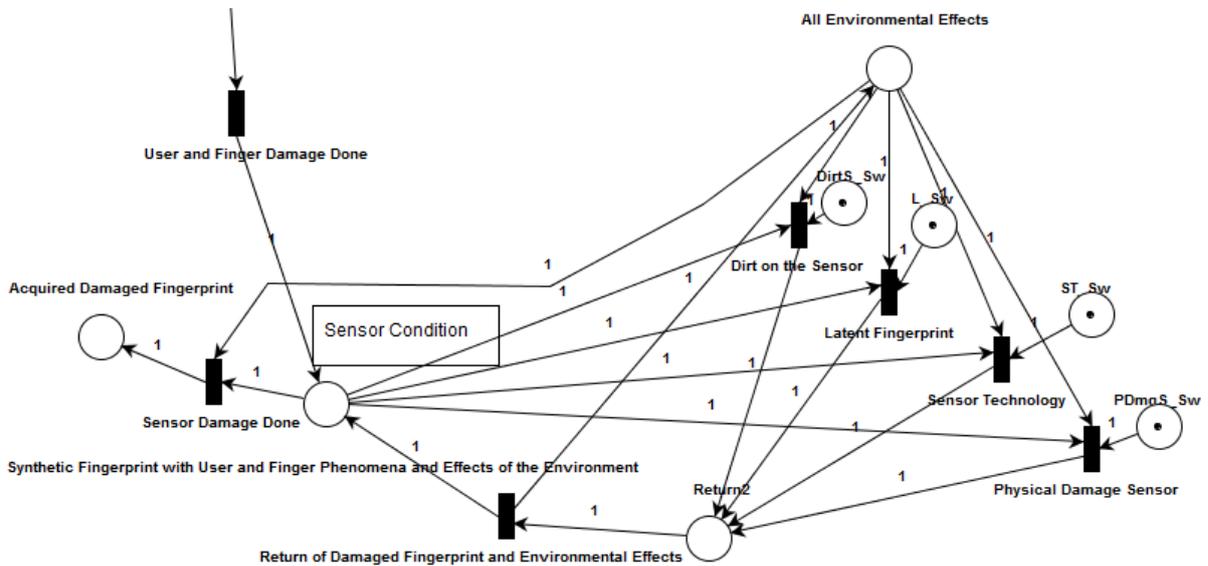


Figure 4.6: The fourth part of the proposed Petri net – the sensor condition.

Figure 4.7 shows the whole proposed Petri net. All four parts are connected together and from a mathematical point of view. The definition of the net is: [44] [A1]

$FGN = (P, T, F, W, K, M_0)$, where

$P = \{\text{Start, Class Input, Class Specified, Shape Input, Density Input, Seed Input, All Input data, Synthetic Fingerprint, PDmg_Sw, P_Sw, ConReg_Sw, Dry_Sw, NoCoop_Sw, Dis_Sw, Dirt_Sw, Return, All Environmental Effects, Vib Eff, Surr Light Eff, Temper Eff, EMR Eff, Vib Input, Surr Light Input, Temper Input, EMR Input, Acquired Damaged Fingerprint, Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Return2, DirtS_Sw, L_Sw, ST_Sw, PdmgS_Sw}\}$.

$T = \{\text{Input Definitions, Class, Cores \& Deltas, Shape, Density, Seed, Gabor filter, User and Finger Damage Done, Physical Damage Finger, Pressure, Contact Region, Dry or Moist Finger, Non-cooperative behaviour, Skin disease, Dirt on the finger, Return of Synthetic Fingerprint and Environmental Effects, Effects Merger, Vibration, Surrounding Light, Temperature, Electro-Magnetic Radiation, Sensor Damage Done, Return of Damaged Fingerprint and Environmental Effects, Dirt on the Sensor, Latent Fingerprint, Sensor Technology, Physical Damage Sensor}\}$.

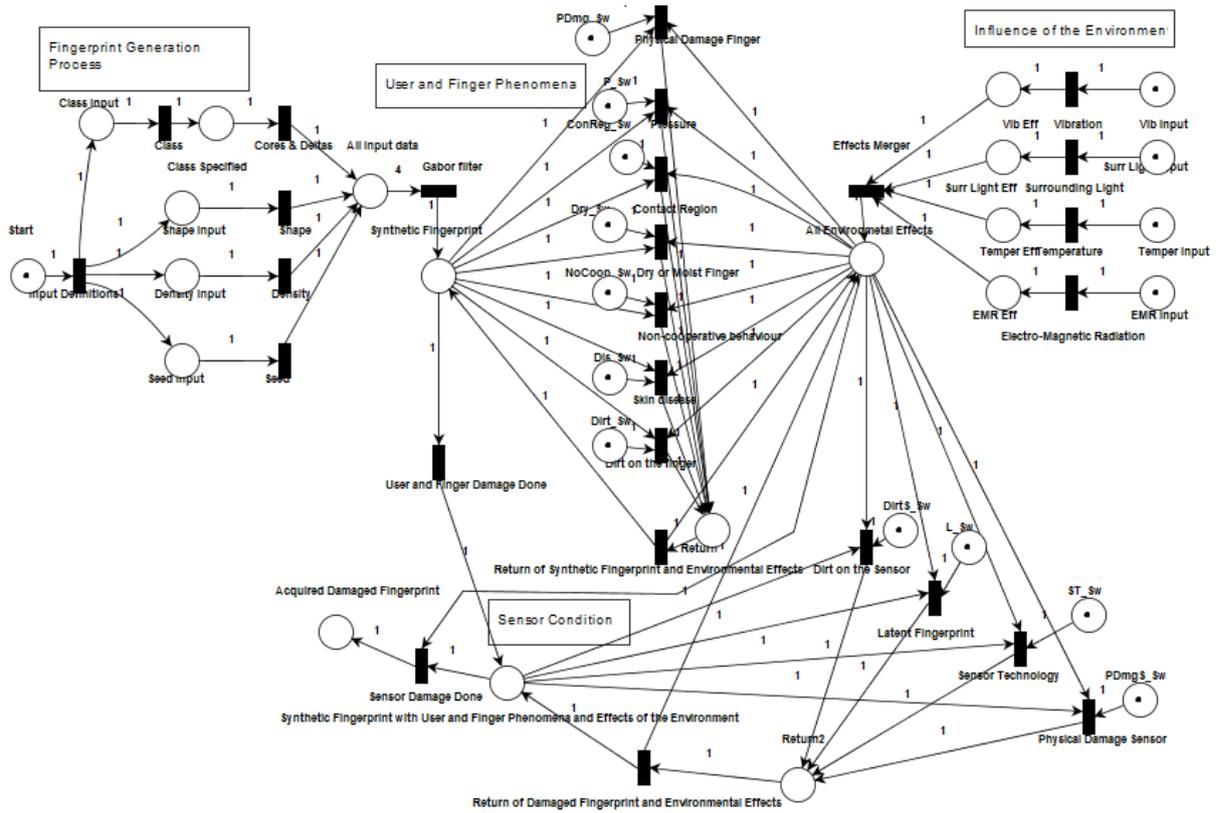


Figure 4.7: The whole proposed Fingerprint generation Petri net.

$F = \{(\text{Start}, \text{Input Definitions}), (\text{Input Definitions}, \text{Class Input}), (\text{Input Definitions}, \text{Shape Input}),$
 $(\text{Input Definitions}, \text{Density Input}), (\text{Input Definitions}, \text{Seed Input}), (\text{Class Input}, \text{Class}), (\text{Shape Input},$
 $\text{Shape}), (\text{Density Input}, \text{Density}), (\text{Seed Input}, \text{Seed}), (\text{Class}, \text{Class Specified}), (\text{Shape}, \text{All Input data}),$
 $(\text{Density}, \text{All Input data}), (\text{Seed}, \text{All Input data}), (\text{Class Specified}, \text{Cores \& Deltas}), (\text{Cores \& Deltas},$
 $\text{All Input data}), (\text{All Input data}, \text{Gabor filter}), (\text{Gabor filter}, \text{Synthetic Fingerprint}), (\text{Synthetic Fingerprint},$
 $\text{Physical Damage Finger}), (\text{PDmg_Sw}, \text{Physical Damage Finger}), (\text{Physical Damage Finger}, \text{Return}),$
 $(\text{Synthetic Fingerprint}, \text{Pressure}), (\text{P_Sw}, \text{Pressure}), (\text{Pressure}, \text{Return}), (\text{Synthetic Fingerprint},$
 $\text{Contact Region}), (\text{ConReg_Sw}, \text{Contact Region}), (\text{Contact Region}, \text{Return}), (\text{Synthetic Fingerprint},$
 $\text{Dry or Moist Finger}), (\text{Dry_Sw}, \text{Dry or Moist Finger}), (\text{Dry or Moist Finger}, \text{Return}),$
 $(\text{Synthetic Fingerprint}, \text{Non-cooperative behaviour}), (\text{NoCoop_Sw}, \text{Non-cooperative behaviour}), (\text{Non-}$
 $\text{cooperative behaviour}, \text{Return}), (\text{Synthetic Fingerprint}, \text{Skin disease}), (\text{Dis_Sw}, \text{Skin disease}), (\text{Skin}$
 $\text{disease}, \text{Return}), (\text{Synthetic Fingerprint}, \text{Dirt on the finger}), (\text{Dirt_Sw}, \text{Dirt on the finger}), (\text{Dirt on the}$
 $\text{finger}, \text{Return}), (\text{Return}, \text{Return of Synthetic Fingerprint and Environmental Effects}), (\text{Return of}$
 $\text{Synthetic Fingerprint and Environmental Effects}, \text{Synthetic Fingerprint}), (\text{Vib Input}, \text{Vibration}),$
 $(\text{SurrLight Input}, \text{Surrounding Light}), (\text{Temper Input}, \text{Temperature}), (\text{EMR Input}, \text{Electro-Magnetic}$
 $\text{Radiation}), (\text{Vibration}, \text{Vib Eff}), (\text{Surrounding Light}, \text{Surr Light Eff}), (\text{Temperature}, \text{Temper Eff}),$
 $(\text{Electro-Magnetic Radiation}, \text{EMR Eff}), (\text{Vib Eff}, \text{Effects Merger}), (\text{Surr Light Eff}, \text{Effects Merger}),$
 $(\text{Temper Eff}, \text{Effects Merger}), (\text{EMR Eff}, \text{Effects Merger}), (\text{Effects Merger}, \text{All Environmental Effects}),$
 $(\text{All Environmental Effects}, \text{Physical Damage Finger}), (\text{All Environmental Effects}, \text{Pressure}), (\text{All}$
 $\text{Environmental Effects}, \text{Contact Region}), (\text{All Environmental Effects}, \text{Dry or Moist Finger}), (\text{All}$

Environmental Effects, Non-cooperative behaviour), (All Environmental Effects, Skin disease), (All Environmental Effects, Dirt on the finger), (All Environmental Effects, Dirt on the Sensor), (All Environmental Effects, Latent Fingerprint), (All Environmental Effects, Sensor Technology), (All Environmental Effects, Physical Damage Sensor), (Return of Synthetic Fingerprint and Environmental Effects, All Environmental Effects), (Return of Damaged Fingerprint and Environmental Effects, All Environmental Effects), (Synthetic Fingerprint, User and Finger Damage Done), (User and Finger Damage Done, Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment), (Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Dirt on the Sensor), (DirtS_Sw, Dirt on the Sensor), (Dirt on the Sensor, Return2), (Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Latent Fingerprint), (L_Sw, Latent Fingerprint), (Latent Fingerprint, Return2), (Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Sensor Technology), (ST_Sw, Sensor Technology), (Sensor Technology, Return2), (Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Physical Damage Sensor), (PDmgS_Sw, Physical Damage Sensor), (Physical Damage Sensor, Return2), (Return2, Return of Damaged Fingerprint and Environmental Effects), (Return of Damaged Fingerprint and Environmental Effect, Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment), (Synthetic Fingerprint with User and Finger Phenomena and Effects of the Environment, Sensor Damage Done), (All Environmental Effects, Sensor Damage Done), (Sensor Damage Done, Acquired Damaged Fingerprint)}.

$$W = \forall t \in T \setminus \{(All\ Input\ data, Gabor\ filter)\}: W(t) = 1, W\{(All\ Input\ data, Gabor\ filter)\} = 4.$$

$$K = \forall p \in P: P(p) = \omega.$$

$$M_0 = \forall p' \in P': M_0(p') = 1, \forall p'' \in P \cap P': M_0(p'') = 0, \text{ where}$$

$P' = \{Start, Vib\ Input, Surr\ Light\ Input, EMR\ Input, PDmg_Sw, P_Sw, ConReg_Sw, Dry_Sw, NoCoop_Sw, Dis_Sw, Dirt_Sw, DirtS_Sw, L_Sw, ST_Sw, PDmgS_Sw\}.$

4.1.2 Examples of the Initial Marking for Various Conditions

The proposed FGN covers fingerprint generation and all possible phenomena that can influence the fingerprint image acquired by a real sensor. The real sensor here is supposed to be understood as a general sensor. In a real case scenario with a specific sensor technology and a specific use, the use of the FGN could be reduced. The used technology could be resistant to some phenomena or the user could be experienced and trained in creating almost perfect fingerprint images. For these simulations of real case scenarios, the net can either be reduced or the initial marking can be changed to reflect the real situation. [44] [A1]

Some of these real case scenarios will now be described with a changed initial marking to cover that situation.

Example 1: The user is healthy, working in the office, and the fingerprint image was acquired with a touchless optical sensor. The skin diseases part is not needed, there is probably no dirt on the finger, or physical fingerprint damage. The use of touchless technology will prevent any latent fingerprints and pressure is impossible to apply. This scenario could be simulated by the net $FGN_{SC1} = (P, T, F, W, K, M_{0SC1})$, where P, T, F, W, K have the same definitions as in the first definition of the FGN and $M_{0SC1} = \forall p_{SC1} \in P_{SC1}: M_{0SC1}(p_{SC1}) = 1, \forall p_{SC1}' \in P \cap P_{SC1}: M_{0SC1}(p_{SC1}') = 0$, where

$P_{SC1} = \{\text{Start, Vib Input, Surr Light Input, EMR Input, ConReg_Sw, Dry_Sw, NoCoop_Sw, DirtS_Sw, ST_Sw, PDmgS_Sw}\}$. [44] [A1]

Example 2: The user is trying a brand new swipe capacitive technology sensor. This capacitive technology is not influenced by surrounding light and the usage of the swipe technology greatly reduces the risk of dirt on the sensor or a possible latent fingerprint. In addition, a brand new sensor is used, therefore it will not likely be physically damaged. The definition of $FGN_{SC2} = (P, T, F, W, K, M_{OSC2})$ is very similar to the previous cases (M_{OSC2} is identical to M_{OSC1} when $p_{SC1}, P_{SC1}, M_{OSC1}, p_{SC1}'$ are renamed respectively, to $p_{SC2}, P_{SC2}, M_{OSC2}, p_{SC2}'$) and $P_{SC2} = \{\text{Start, Vib Input, Surr Light Input, EMR Input, PDmg_Sw, P_Sw, ConReg_Sw, Dry_Sw, NoCoop_Sw, Dis_Sw, Dirt_Sw, ST_Sw}\}$. In this case it is important to note that although the surrounding light has no impact, the token in the *Surr Light Input* place remains there. [44] [A1]

Example 3: A skilled user is using the ultrasonic-capacitive swipe technology. The user's finger is without deep scars or heavy disease. In addition to the swipe technology, which is very resistant to dirt on the sensor and latent fingerprints, the ultrasonic technology ignores any damage or disease on the surface of the finger. A skilled user should achieve an exceptional contact region and some parts of non-cooperative behaviour should not be present. $FGN_{SC3} = (P, T, F, W, K, M_{OSC3})$ with $P_{SC3} = \{\text{Start, Vib Input, Surr Light Input, EMR Input, Dry_Sw, NoCoop_Sw, Dirt_Sw, ST_Sw, PDmgS_Sw}\}$. [44] [A1]

4.2 SyFDaS Core Design and Graphical User Interface

Following the simulation in Subchapter 4.1, the application can be divided into two parts. The first part creates the synthetic fingerprint, and the second part takes care of all damage simulations. For the purpose of fingerprint generation, the base generator (Chaloupka's fingerprint generator) from [11] is used. This generator is using minutiae as an input (as covered in Subchapter 3.1). This generator is further enhanced as is described in Subchapter 4.3. The final graphical user interface (GUI) can be seen in Figure 4.8. By setting the filter, orientation (with fingerprint class), mask, and density in their respective parts of the GUI, the generation is prepared. The most important step is to define some minutiae. After that, generation of the fingerprint can begin. There is, of course, the possibility to save and load minutiae (or the whole fingerprint image).

Consequently, all possible damage simulations must be held in one clean interface. To do that, it is essential to use the modular approach on this interface. The information about a sensor, a type of sensor, damage and all the controls related to it has to be easily accessible. There is also a possibility that more damage simulations or sensors will be added to this application, so the interface has to be prepared for it. Other ancillary operations are part of the solution as well. These are the loading of a prearranged synthetic fingerprint image, saving the current image, the possibility to quickly save and load an image, and to undo/redo. Undo/redo is very useful because there are usually a lot of experiments with fingerprint damages and sometimes experiments can go wrong. [1]

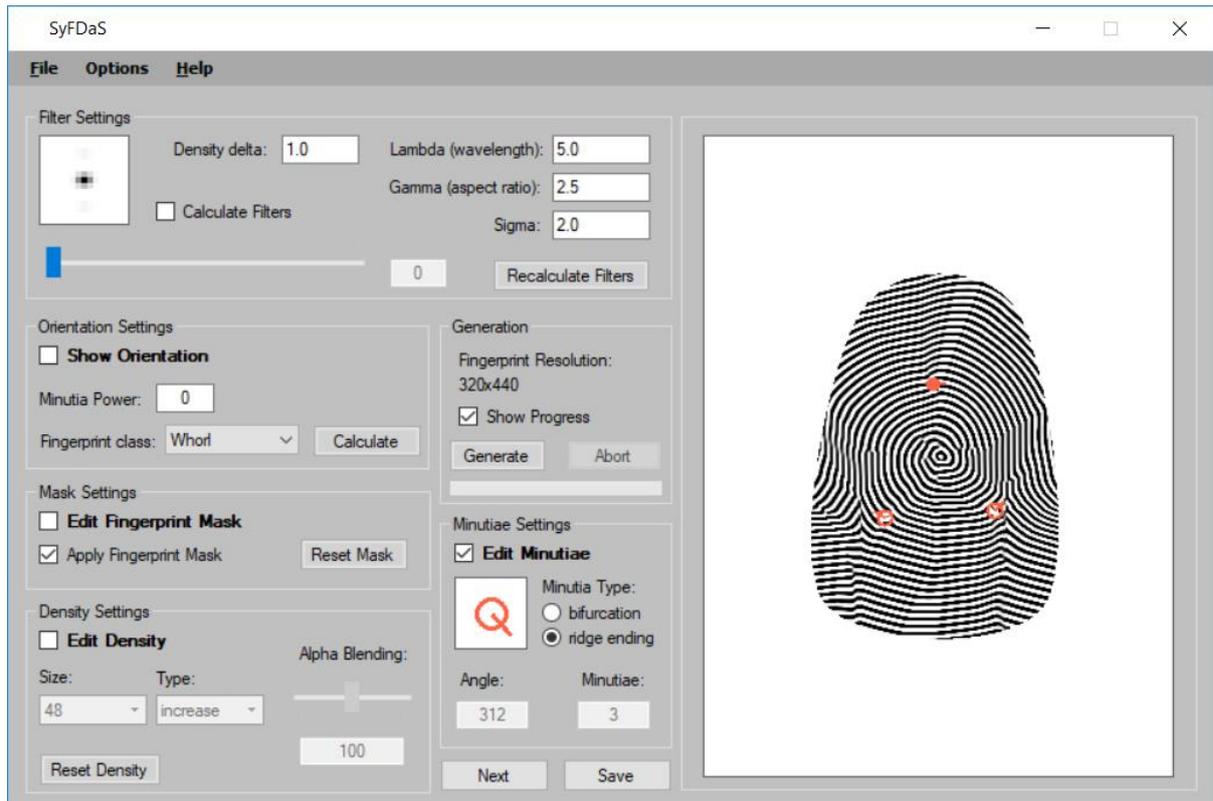


Figure 4.8: Screenshot of SyFDaS – generation part.

The starting point is the generation window. There is not much space for extensions, so the fingerprint damage simulation window is in a new section. This part can theoretically contain all influential factors and it is implemented as a new window as a next step to the generation. To summarize, before the damage simulation can start, information about the sensor technology (Subchapter 2.3), sensor type (touch, swipe, contactless), and damage (which is simulated) are needed. As it was previously said, the number of damages is theoretically overwhelming, so it will be better to divide them as it was in Subchapter 3.2. So the first part of the application deals with the sensor, the second with the damage, and the third part is reserved for the options and input values of individual damage simulations. The next part (on the right) is the fingerprint image that is carried on from the generating window. The last section is between controls and fingerprint imaging, and that is the main control part. There are options for applying the damage, cycle through input values if needed, etc. The final form of the window can be seen in Figure 4.9. [1]

Modularity is achieved by several abstract classes joined with the main GUI. If there is no need to change the GUI, then it is only necessary to create a newly derived class to get new behaviour. For example, the proper settings of the compulsory parts of the abstract class ensures that all available sensors and damage simulations are filled to the GUI. They are also enabled and disabled as needed. Two main abstract classes are *Sensor Type* (that encapsulates information about sensor, mainly name, type and order of damage simulations) and *Damage Type* (that contains all about damages, mainly implementation of the damage and connection to the input part of GUI). As for the other functionalities, the supported formats for saving and loading are bmp, png, tif, gif, and jpg. Basic batch processing is also implemented (more information in Subchapter 4.5).

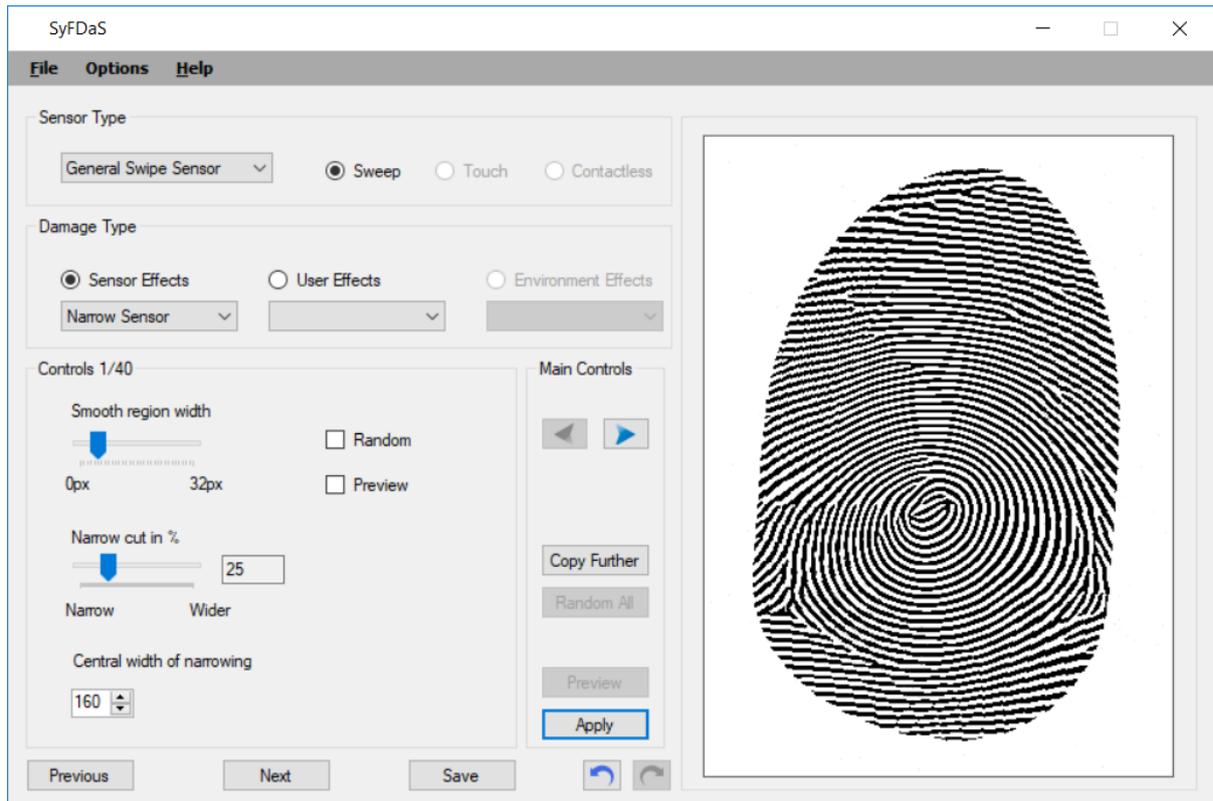


Figure 4.9: Screenshot of SyFDaS – simulation part.

4.3 Enhancement of Fingerprint Generation

The original version of the SyFDaS generator was using only defined minutiae for determining the orientation field. Unfortunately, it was hard to create realistic looking fingerprints because of that. Too often the orientation field was not from one of the fingerprint classes, which had a great impact on the look of the generated image. It was decided to add the desired fingerprint class as one of the inputs for the generator. The methodology of generation had to be preserved – the main inputs should be defined as minutiae. The original algorithm started with an orientation field at 0° in point of the field, after that it was changed based on the given minutiae. The new version changes this part of the algorithm. If a fingerprint class is given an orientation field, it is first processed by the field generation for that particular class. This initialization defines cores and deltas somewhere in the image. That is done by several random generation functions with the presumption that middle of the fingerprint should be in the centre of generated image. The power of minutiae is also taken into account. This power determines how much each minutia should influence this initialized orientation field. If the power is 0, the defined orientation of the minutia is basically irrelevant and it is given new according to the orientation field. Otherwise, following the formula (Eq. 4.2) for the determination of the orientation of the given point (and its surroundings) is used:

$$\text{orientation}(i, j) = \text{orientation}(i, j) - \frac{\text{anglediff}}{\max\left(1, \frac{\text{distance}}{\text{power}}\right)} \quad (\text{Eq. 4.2})$$

Where *power* is user defined as the power of minutiae, *distance* is the distance of the point (*i, j*) from the minutia, and *anglediff* is the difference of current angle from the orientation field and the defined orientation of minutia. With this extension, fingerprints generated by SyFDaS are much more realistic looking. Paradoxically, the user gets the feeling that the process of generation is more influenced by input settings. It is “easier” to make a desired change in the generated fingerprints.

4.4 Touch-based Sensor Damage Simulation

The application has some of the damage simulations implemented. All of them are implemented for a hypothetical general sensor. In reality, they are close to a capacitive or optical touch sensor. In the subchapters below, each of the completed damage simulations will be discussed. Some of these are taken into account in Chapter 5 when designing complicated damage simulations.

4.4.1 Damaged Sensor

There are databases (specifically with optical sensors) where this type of damage is clearly shown. It is a thin black line usually connected to the edge of the acquired fingerprint. This line corresponds with the crack on the protective glass. In extreme cases there could be a web of broken glass instead of one crack. Some types of dirt on the sensor look like this crack. For example, an eyelash of straight hair leaves the same trace on the acquired fingerprint. This phenomenon was also listed in Subchapter 3.2. [1]

It is simulated by simply drawing a line in the desired area on the fingerprint. It is necessary to find the right thickness of the line to properly simulate the crack or hair on the sensor. This method is required to be fully determined by a starting point (x_{start}, y_{start}), a direction, a length, and a thickness. The end point of the line is determined by Eq. 4.3. [1]

$$\begin{aligned} x_{end} &= \sin(\text{angle}) \cdot \text{length} \\ y_{end} &= \cos(\text{angle}) \cdot \text{length} \end{aligned} \quad (\text{Eq. 4.3})$$

Length is defined relatively to the image width by Eq. 4.4.

$$\text{length}_{absolute} = \frac{\text{ImageWidth}}{10} \cdot \left(1 + \frac{\text{length}_{relative} \cdot 2}{10}\right) \quad (\text{Eq. 4.4})$$

4.4.2 Pressure and Moisture

When it comes to applying intentional damage to the fingerprint, too much pressure is the first thing that comes to mind. Similarly, as in the simulation of a damaged sensor, moisture influences the final

image in the same way as pressure. Both dampness and pressure increase the thickness and the contrast of the ridges. The more pressure the user applies or the damper their finger is, the thicker the lines are. In extreme cases almost no lines are visible on the fingerprint, because the fingerprint is either entirely black or white. This factor was also mentioned in Subchapter 3.2. [1]

Morphological operations of erosion and dilation [60] will be used to simulate these effects. These operators are commonly used in image processing, for example to increase readability of the text or thinning the lines as in the same part of the fingerprint recognition process (Subchapter 2.4). They are defined and used only on binary-coloured or greyscale images. Applying pressure does the same thing as morphological operators, which enlarge or shrink ridges. Morphological operators only need a structure element to determine their magnitude. The structural elements used can be seen on Figure 4.10. When creating dilatation, structural elements are used to thin the ridges. On the other hand, when creating erosion the same structural elements are used to thicken up the ridges. The exact formulas (presuming that image is defined by Eq. 4.5) are Eq. 4.6 for erosion and Eq. 4.7 for dilatation. [1]

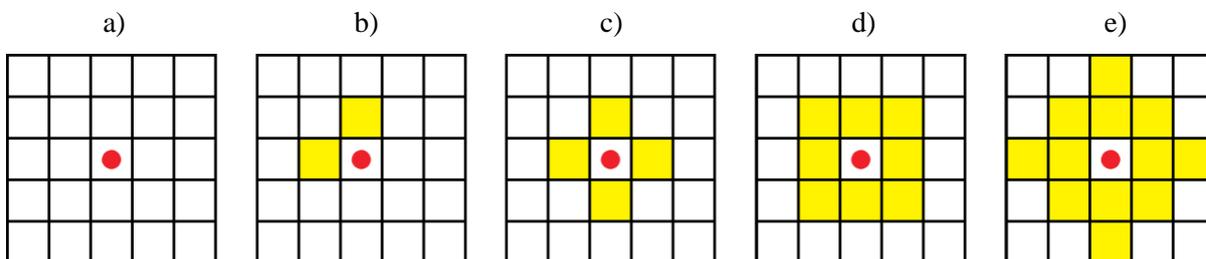


Figure 4.10: A demonstration of all structure elements used [1].

$$f(x, y) = \text{intensity} \quad (\text{Eq. 4.5})$$

The image in point x, y is defined by its colour intensity. In this case, only greyscale images are used so only one intensity is defined. S in following formulas is an input – structure element.

$$\text{erosion}(x, y) = \min_{\forall(u,v) \in S} (x + u, y + v) \quad (\text{Eq. 4.6})$$

$$\text{dilatation}(x, y) = \max_{\forall(u,v) \in S} (x + u, y + v) \quad (\text{Eq. 4.7})$$

The structure element consists of pixels in the neighbourhood of the investigated pixel x, y . In this application it should be evenly distributed around the one being investigated. After defining the smallest structure element that is evenly distributed (which can be seen in Figure 4.10c) it turns out that it has too great of an impact on the fingerprint image. Because of that it was necessary to include the structure element in Figure 4.10b despite the fact that its damage to the fingerprint is inaccurate. [1]

4.4.3 Fingerprint Distortion

Fingerprint distortion is typically done unintentionally. This type of damage is so common that it is almost impossible to make a fingerprint image without it. It is created due to skin deformation and the non-orthogonal finger pressure to the sensor. In fact, every little finger movement when touching the sensor glass creates this distortion. The skin is very elastic, and with the exception of extreme cases,

users do not even feel it. To make a non-distorted image the major focus would have to be on not moving the finger and on applying the pressure exactly orthogonally. Even this might not be enough, because two-dimensional images are created out of a three-dimensional finger, so the skin is stretching and compressing and thus creating distortion. Fingerprint distortion is one of the few damages that can change the position of the minutia and even change the distance among the minutiae themselves. This is a problem for fingerprint recognition algorithms that use the minutiae position as one of their main recognition elements. Despite that fact, distortion is almost invisible to an untrained eye. Specialized images with marked minutiae or using a square grid are necessary. In Subchapter 3.2 the non-cooperative behaviour of the user is described. If the user is forced to give his/her fingerprints or wants to inconspicuously damage the fingerprint by small movements and changes of pressure, that can lead to distortion. [1]

The same distortion model as in SFinGe will be used to simulate this distortion. In [61] a model was designed and also verified. The model divides the fingerprint into three areas (as can be seen in Figure 4.11). The internal area (shown by the red colour) where the finger is pushed so hard that the skin cannot be deformed. The second is the external area (shown by the yellow colour) where the pressure is so low that the skin is maximally distorted. And the third area is the transition area (shown by the orange colour), which combines the two previous areas. The greater the intensity of the orange colour is shown, the lower distortion is applied. Each image in Figure 4.11 shows that a different level of skin plasticity has been set. After this, the angle and the translation in each axis are needed to fully determine and apply the distortion. Because it can be made worse by intentionally trying to achieve this type of damage, the range of input values was increased to cover these cases. For input, models need 3 values: a skin plasticity coefficient, translation in x and y , and a rotation angle. [1]



Figure 4.11: Various fingerprint distortion areas [1].

The mathematical background for this damage is provided. For easy recognition, the rotation angle will be labelled as θ and the translation value as dx and dy . For the definition of all areas, skin plasticity (labelled as sp) and ellipse defined by the centre of ellipse ($ellip_x$ and $ellip_y$) and semi-axes ($semi_x$ and $semi_y$) are used. More information can be found in the following paragraphs. The method, or more precisely the distortion model, is defined by these basic formulas [61]: [1]

$$\begin{aligned} \text{distortion: } \mathfrak{R}^2 \rightarrow \mathfrak{R}^2, \text{ coef: } \mathfrak{R} \times \mathfrak{R} \rightarrow \mathfrak{R}, \Delta: \mathfrak{R}^2 \rightarrow \mathfrak{R}^2, \text{ dist: } \mathfrak{R}^2 \rightarrow \mathfrak{R} \\ \text{distortion}(\mathbf{v}) = \mathbf{v} + \Delta(\mathbf{v}) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) \end{aligned} \quad (\text{Eq. 4.8})$$

$$\text{coef}(\text{dist}(\mathbf{v}), sp) = \frac{1}{2} \left[1 - \cos \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \right] \quad (\text{Eq. 4.9})$$

$$\begin{aligned} \Delta(\mathbf{v}) = [R_\theta(\mathbf{v} - \mathbf{ellip}) + \mathbf{ellip} + \mathbf{d}] - \mathbf{v}, \\ \mathbf{d} = \begin{bmatrix} dx \\ dy \end{bmatrix}, \quad R_\theta = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \quad \mathbf{ellip} = \begin{bmatrix} \text{ellip}_x \\ \text{ellip}_y \end{bmatrix} \end{aligned} \quad (\text{Eq. 4.10})$$

$$\text{dist}(\mathbf{v}) = \sqrt{(\mathbf{v} - \mathbf{ellip})^T \mathbf{A}^{-1} (\mathbf{v} - \mathbf{ellip})} - 1, \quad \mathbf{A} = \begin{bmatrix} \text{semi}_x^2 & 0 \\ 0 & \text{semi}_y^2 \end{bmatrix} \quad (\text{Eq. 4.11})$$

Special cases: Area A: $\text{coef} = 0$, $\text{dist} = 0$ Area C: $\text{coef} = 1$

As indicated, the model is not changing the intensity of the pixels, but it is changing their position.

So each pixel $\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$ is transformed to new coordinates $\mathbf{v}' = \begin{bmatrix} v_x' \\ v_y' \end{bmatrix}$ following the formula Eq. 4.8.

At first the ellipse usage is explained. The model is calculating the distortion of a finger when creating two-dimensional images. As can be seen, the elevated part of the fingertips is more or less elliptical. That part is determining the individual areas which the model uses to create distortion. The shape was generalized to ellipse with the centre of the image and semi-axes values $\text{semi}_x = 60$ and $\text{semi}_y = 100$. These values were determined experimentally because there is no information about fingerprint images and there is no better way of determining them in this state. All other input values are taken from the user. [1]

In the next paragraphs, all formulas (Eq. 4.8, Eq. 4.9, Eq. 4.10, and Eq. 4.11) are explained. Although the mathematical formulas listed above are very comprehensible for humans, they are not comprehensible for a computer. Matrix and vector operations are usually very slow and have limited power. For these reasons most of the formulas were modified and their modified versions are shown below. **Area A** corresponds with the internal area (red area in Figure 4.11) and it is represented by a defined ellipse (including boundaries). **Area B** corresponds with the transition area (orange area in Figure 4.11) and it is represented by points around the ellipse that are in a certain distance, i.e. a distance which must be lower or equal to skin plasticity. Finally, **area C** is the external area (yellow area in Figure 4.11) and it is represented by other pixels. [1]

Formula Eq. 4.11 determines the distance of the current pixel from the nearest point of the ellipse (area A). The Mahalanobis distance decreased by one is used (as it was in [61]). In the method dist the Eq. 4.11 is adjusted to (Eq. 4.12): [1]

$$\text{dist}(\mathbf{v}) = \sqrt{(\mathbf{v} - \mathbf{ellip})^T \frac{1}{\text{semi}_x^2 \cdot \text{semi}_y^2} \begin{bmatrix} \text{semi}_x^2 & 0 \\ 0 & \text{semi}_y^2 \end{bmatrix} (\mathbf{v} - \mathbf{ellip})} - 1$$

$$dist(\mathbf{v}) = \sqrt{(\mathbf{v} - \mathbf{ellip})^T \begin{bmatrix} semi_x^{-2} & 0 \\ 0 & semi_y^{-2} \end{bmatrix} (\mathbf{v} - \mathbf{ellip})} - 1$$

$$dist(\mathbf{v}) = \sqrt{\begin{bmatrix} (v_x - ellip_x) \cdot semi_x^{-2} \\ (v_y - ellip_y) \cdot semi_y^{-2} \end{bmatrix} (\mathbf{v} - \mathbf{ellip})} - 1$$

$$dist(\mathbf{v}) = \sqrt{[(v_x - ellip_x)^2 \cdot semi_x^{-2} + (v_y - ellip_y)^2 \cdot semi_y^{-2}] - 1} \quad (\text{Eq. 4.12})$$

Formula Eq. 4.9 basically specifies where the point is (in area B) between area A and C. The coefficient effect can be seen in Figure 4.11 where it is used to define the intensity of the orange colour representing area B. The formula itself remains as is. [1]

Formula Eq. 4.10 is representing the effect of rotation and translation. It shifts the image so that it has the centre of rotation (centre of ellipse) in the coordinates $(0, 0)^T$. Then it uses the R_θ matrix to rotate and shift the image back to the original coordinates. After that it does translation by adding the respective value and subtracts the original value of the pixel to create a differential value. The adjusted formula in its final form is below (Eq. 4.13). [1]

$$\begin{aligned} \Delta(v_x) &= [(v_x - ellip_x) \cdot \cos(\theta) + (v_y - ellip_y) \cdot \sin(\theta)] + ellip_x + dx - v_x \\ \Delta(v_y) &= [(v_x - ellip_x) \cdot (-\sin(\theta)) + (v_y - ellip_y) \cdot \cos(\theta)] + ellip_y + dy - v_y \end{aligned} \quad (\text{Eq. 4.13})$$

Formula Eq. 4.8 puts all things together. The difference is modified by a coefficient and added to the original value. If \mathbf{v} is in area A, it stays as it was. If \mathbf{v} is in area C, it is maximally translated and rotated because the *coef* is 1. If \mathbf{v} is in area B, it is translated and rotated to some degree depending on the accurate location. [1]

When real fingers stretch or compress, the sensor acquires a distorted image, but even when a perfect non-distorted image is acquired, there is a limited number of points. In spots where the finger is stretching, it is not certain that the model will have points to distort. To fill in these places, interpolation is required. In order to calculate the interpolation, the original (undistorted) coordinates of the points are also required. This means that the calculation of the inversion model must be done. This is complicated because this model cannot be analytically inverted. To determine its inverted value a numerical method is required. For this model the Newton-Raphson method [62] [63], which can numerically compute the inversion of multiple variable functions, was used. Its variant for two variables can be written as [62] [63]: [1]

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \begin{bmatrix} X_i \\ Y_i \end{bmatrix} - [J]^{-1} \cdot \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} \quad \text{where } \mathbf{x} = \begin{bmatrix} X_i \\ Y_i \end{bmatrix}, \quad J = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x} & \frac{\partial f_1(\mathbf{x})}{\partial y} \\ \frac{\partial f_2(\mathbf{x})}{\partial x} & \frac{\partial f_2(\mathbf{x})}{\partial y} \end{bmatrix} \quad (\text{Eq. 4.14})$$

Now it is necessary to prepare the formula Eq. 4.8 which represents the model to use in the formula Eq. 4.14. As can be seen, two functions are needed. They are done by adjusting the formula Eq. 4.8 to not use vectors: [1]

$$\begin{aligned} \text{distort}_x(\mathbf{v}) &= v_x + \Delta(v_x) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) \\ \text{distort}_y(\mathbf{v}) &= v_y + \Delta(v_y) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) \end{aligned} \quad (\text{Eq. 4.15})$$

After that the formulas from Eq. 4.15 are inserted into the formula Eq. 4.14:

$$\begin{aligned} \begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} &= \begin{bmatrix} v_{x(i)} \\ v_{y(i)} \end{bmatrix} - [\mathbf{J}]^{-1} \cdot \begin{bmatrix} \text{distort}_x(\mathbf{v}) \\ \text{distort}_y(\mathbf{v}) \end{bmatrix} \\ \text{where } \mathbf{v} &= \begin{bmatrix} v_{x(i)} \\ v_{y(i)} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \frac{\partial \text{distort}_x(\mathbf{v})}{\partial v_x} & \frac{\partial \text{distort}_x(\mathbf{v})}{\partial v_y} \\ \frac{\partial \text{distort}_y(\mathbf{v})}{\partial v_x} & \frac{\partial \text{distort}_y(\mathbf{v})}{\partial v_y} \end{bmatrix} \end{aligned} \quad (\text{Eq. 4.16})$$

A further modification will be divided by area where current x is located. When x is in area A there is no distortion, so points are the same as in the original picture and no interpolation nor distortion is needed to be calculated. If x is in area C, it means that the coefficient is equal to one and the Jacobian matrix J_C for area C will be (Eq. 4.17): [1]

$$J_C = \begin{bmatrix} \frac{\partial v_x}{\partial v_x} + \frac{\partial \Delta(v_x)}{\partial v_x} & \frac{\partial v_x}{\partial v_y} + \frac{\partial \Delta(v_x)}{\partial v_y} \\ \frac{\partial v_y}{\partial v_x} + \frac{\partial \Delta(v_y)}{\partial v_x} & \frac{\partial v_y}{\partial v_y} + \frac{\partial \Delta(v_y)}{\partial v_y} \end{bmatrix} \quad (\text{Eq. 4.17})$$

$$\begin{aligned} \frac{\partial \Delta(v_x)}{\partial v_x} &= [((v_x - \text{ellip}_x) \cdot \cos(\theta))' + (0 - 0) \cdot 0] + 0 + 0 - 1 = \cos(\theta) - 1 \\ \frac{\partial \Delta(v_x)}{\partial v_y} &= [(0 - 0) \cdot 0 + ((v_y - \text{ellip}_y) \cdot \sin(\theta))'] + 0 + 0 - 0 = \sin(\theta) \\ \frac{\partial \Delta(v_y)}{\partial v_x} &= [((v_x - \text{ellip}_x) \cdot (-\sin(\theta)))' + (0 - 0) \cdot 0] + 0 + 0 - 0 = -\sin(\theta) \\ \frac{\partial \Delta(v_y)}{\partial v_y} &= [(0 - 0) \cdot (-0) + ((v_y - \text{ellip}_y) \cdot \cos(\theta))'] + 0 + 0 - 1 = \cos(\theta) - 1 \end{aligned} \quad (\text{Eq. 4.18})$$

$$J_C = \begin{bmatrix} 1 + \cos(\theta) - 1 & 0 + \sin(\theta) \\ 0 - \sin(\theta) & 1 + \cos(\theta) - 1 \end{bmatrix}$$

$$J_C = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{Eq. 4.19})$$

Now the formula Eq. 4.19 can be inserted into Eq. 4.16 and deduce the final recurrent formula for the area C.

$$\begin{aligned} \begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} &= \begin{bmatrix} v_{x(i)} \\ v_{y(i)} \end{bmatrix} - \left(\frac{1}{\cos^2(\theta) + \sin^2(\theta)} \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \right) \cdot \begin{bmatrix} \text{distort}_x(\mathbf{v}) \\ \text{distort}_y(\mathbf{v}) \end{bmatrix} \\ \begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} &= \begin{bmatrix} v_{x(i)} \\ v_{y(i)} \end{bmatrix} - \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \text{distort}_x(\mathbf{v}) \\ \text{distort}_y(\mathbf{v}) \end{bmatrix} \\ \begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} &= \begin{bmatrix} v_{x(i)} - (\cos(\theta) \cdot \text{distort}_x(\mathbf{v}) - \sin(\theta) \cdot \text{distort}_y(\mathbf{v})) \\ v_{y(i)} - (\sin(\theta) \cdot \text{distort}_x(\mathbf{v}) + \cos(\theta) \cdot \text{distort}_y(\mathbf{v})) \end{bmatrix} \end{aligned} \quad (\text{Eq. 4.20})$$

The formula Eq. 4.20 can, after some iterations, come up with the inverted value for the pixel $(0, 0)^T$ in area C for any input values. The next step that is needed is to generalize that formula to invert any pixel in area C. Fortunately, that can be easily done by shifting the Eq. 4.20 by the value of the pixel, which is inverted and marked as $(v_{x0}, v_{y0})^T$. The final formula for the numerical inverting of any pixel of any input value in area C is: [1]

$$\begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} = \begin{bmatrix} v_{x(i)} - (\cos(\theta) \cdot (\text{distort}_x(\mathbf{v}) - v_{x0}) - \sin(\theta) \cdot (\text{distort}_y(\mathbf{v}) - v_{y0})) \\ v_{y(i)} - (\sin(\theta) \cdot (\text{distort}_x(\mathbf{v}) - v_{x0}) + \cos(\theta) \cdot (\text{distort}_y(\mathbf{v}) - v_{y0})) \end{bmatrix} \quad (\text{Eq. 4.21})$$

If x is in area B, then the evaluation of the Jacobian matrix gets rather complicated. Thus, for the inversion of all values in image it is needed to determine the J_B – Jacobian matrix for area B. [1]

$$J_B = \begin{bmatrix} \frac{\partial v_x}{\partial v_x} + \frac{\partial[\Delta(v_x) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp)]}{\partial v_x} & \frac{\partial v_x}{\partial v_y} + \frac{\partial[\Delta(v_x) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp)]}{\partial v_x} \\ \frac{\partial v_y}{\partial v_x} + \frac{\partial[\Delta(v_y) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp)]}{\partial v_x} & \frac{\partial v_y}{\partial v_y} + \frac{\partial[\Delta(v_y) \cdot \text{coef}(\text{dist}(\mathbf{v}), sp)]}{\partial v_y} \end{bmatrix} \quad (\text{Eq. 4.22})$$

To ensure a higher readability, each element of the matrix J_B (Eq. 4.22) is discussed separately. Their labels are:

$$\begin{aligned} J_B &= \begin{bmatrix} j_{b1} & j_{b2} \\ j_{b3} & j_{b4} \end{bmatrix} \\ j_{b1} &= 1 + \frac{\partial \Delta(v_x)}{\partial v_x} \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) + \Delta(v_x) \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} \\ j_{b2} &= 0 + \frac{\partial \Delta(v_x)}{\partial v_y} \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) + \Delta(v_x) \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_y} \\ j_{b3} &= 0 + \frac{\partial \Delta(v_y)}{\partial v_x} \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) + \Delta(v_y) \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} \\ j_{b4} &= 1 + \frac{\partial \Delta(v_y)}{\partial v_y} \cdot \text{coef}(\text{dist}(\mathbf{v}), sp) + \Delta(v_y) \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_y} \end{aligned} \quad (\text{Eq. 4.23})$$

As can be seen, to evaluate these elements it is needed to find out partial derivatives of the function Δ from the formula Eq. 4.18 and partial derivatives of the function coef . [1]

$$\begin{aligned} \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} &= \frac{1}{2} \left[1 - \cos \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \right]' = \frac{1}{2} \left[-\cos' \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \cdot \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right)' \right] = \\ &= \frac{1}{2} \left[\sin \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \cdot \left(\frac{(\text{dist}(\mathbf{v}) \cdot \pi)' \cdot sp}{sp^2} \right)' \right] = \frac{1}{2} \left[\sin \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \cdot \left(\frac{\text{dist}(\mathbf{v})' \cdot \pi \cdot sp}{sp^2} \right) \right] \quad (\text{Eq. 4.24}) \end{aligned}$$

From the last result (Eq. 4.24) it can be seen that the derivation of coefficient is the same for ∂x and ∂y and it is:

$$\frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} = \frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_y} = \frac{1}{2} \left[\sin \left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp} \right) \cdot \left(\frac{\text{dist}(\mathbf{v})' \cdot \pi}{sp} \right) \right] \quad (\text{Eq. 4.25})$$

But it is still needed to compute the derivation of the function *dist*.

$$\begin{aligned} \frac{\partial \text{dist}(\mathbf{v})}{\partial v_x} &= \left[\left(\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2} \right)^{\frac{1}{2}} \right]' = \\ &= \frac{1}{2} \left(\frac{1}{\sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \right) \left(\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2} \right)' = \\ &= \frac{1}{2} \frac{1}{\sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \frac{[(v_x - \text{ellip}_x)^2]' \cdot \text{semi}_x^2}{\text{semi}_x^4} = \quad (\text{Eq. 4.26}) \end{aligned}$$

$$= \frac{\frac{1}{2} \cdot \frac{2(v_x - \text{ellip}_x)}{\text{semi}_x^2}}{\sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} = \frac{(v_x - \text{ellip}_x)}{\text{semi}_x^2 \sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \quad (\text{Eq. 4.27})$$

The only difference in computing partial derivatives of function *dist* with respect to v_y is in the last step in the formula, Eq. 4.26. Without inferring again, here is the result:

$$\frac{\partial \text{dist}(\mathbf{v})}{\partial v_y} = \frac{(v_y - \text{ellip}_y)}{\text{semi}_y^2 \sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \quad (\text{Eq. 4.28})$$

Now it is possible to complete the formula Eq. 4.25 for partial derivatives by the substitution of the results in the formula Eq. 4.27 and Eq. 4.28. [1]

$$\frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} = \frac{1}{2} \left[\sin\left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp}\right) \cdot \frac{\frac{(v_x - \text{ellip}_x)}{\sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}}}{sp} \right]$$

$$\frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_x} = \frac{\sin\left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp}\right) \cdot (v_x - \text{ellip}_x) \cdot \pi}{2 \cdot sp \cdot \text{semi}_x^2 \cdot \sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \quad (\text{Eq. 4.29})$$

$$\frac{\partial \text{coef}(\text{dist}(\mathbf{v}), sp)}{\partial v_y} = \frac{\sin\left(\frac{\text{dist}(\mathbf{v}) \cdot \pi}{sp}\right) \cdot (v_y - \text{ellip}_y) \cdot \pi}{2 \cdot sp \cdot \text{semi}_y^2 \cdot \sqrt{\frac{(v_x - \text{ellip}_x)^2}{\text{semi}_x^2} + \frac{(v_y - \text{ellip}_y)^2}{\text{semi}_y^2}}} \quad (\text{Eq. 4.30})$$

Partial derivatives of the function *coef* and Δ in formulas Eq. 4.29, Eq. 4.30, Eq. 4.18 can now be substituted to the formula Eq. 4.23 of elements of the Jacobian matrix for area B J_B . In the result there are no ways to reduce the complexity, so for the sake of readability the final formula stays in the format without substitution and uses labelling from the Eq. 4.23. The Jacobian matrix J_B (in the formula Eq. 4.16) is used to get the recurrent formula (Eq. 4.31): [1]

$$\begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} = \begin{bmatrix} v_{x(i)} \\ v_{y(i)} \end{bmatrix} - \left(\frac{1}{b_{j1}b_{j4} - b_{j2}b_{j3}} \cdot \begin{bmatrix} b_{j4} & -b_{j2} \\ -b_{j3} & b_{j1} \end{bmatrix} \right) \cdot \begin{bmatrix} \text{distort}_x(\mathbf{v}) \\ \text{distort}_y(\mathbf{v}) \end{bmatrix} \quad (\text{Eq. 4.31})$$

As with the formula Eq. 4.21, for this formula to work properly on any input pixels adding shifting parameters is required, $(v_{x0}, v_{y0})^T$ which denotes coordinates from which the calculation of the inverse formula starts. [1]

$$\begin{bmatrix} v_{x(i+1)} \\ v_{y(i+1)} \end{bmatrix} = \begin{bmatrix} v_{x(i)} - \left(\frac{b_{j4}(\text{distort}_x(\mathbf{v}) - v_{x0})}{b_{j1}b_{j4} - b_{j2}b_{j3}} - \frac{b_{j2}(\text{distort}_y(\mathbf{v}) - v_{y0})}{b_{j1}b_{j4} - b_{j2}b_{j3}} \right) \\ v_{y(i)} - \left(\frac{-b_{j3}(\text{distort}_x(\mathbf{v}) - v_{x0})}{b_{j1}b_{j4} - b_{j2}b_{j3}} + \frac{b_{j1}(\text{distort}_y(\mathbf{v}) - v_{y0})}{b_{j1}b_{j4} - b_{j2}b_{j3}} \right) \end{bmatrix} \quad (\text{Eq. 4.32})$$

The formula Eq. 4.32 with the substituted function Δ from the formula Eq. 4.23 corresponds with the method used in the implementation. Each numerical method must have an ending condition. In this case if the following condition (Eq. 4.33) is satisfied, it ends the iterations. [1]

$$|v_{x(i+1)} - v_{xi}| < 10^{-4} \wedge |v_{y(i+1)} - v_{yi}| < 10^{-4} \quad (\text{Eq. 4.33})$$

As it experimentally shows up, although that method should be by [63] convergent, it sometimes cycles between some solutions. To prevent this behaviour, after 100 iterations it does 10 more and uses the best solution thus far (the best is the one with the smallest cumulative error when computing the inverted value back using the original model). All these fixed values can be changed. [1]

After this, it is finally possible to use the interpolation to approximate the value that should be in this point. For this purpose, bilinear interpolation is used. The basic formula for this interpolation of point (v_x, v_y) (v_x, v_y can be in this case real numbers) is (Eq. 4.34) [64]: [1]

$$\begin{aligned} \text{intensity}(v_x, v_y) &= (1-t)(1-u)i_1 + t(1-u)i_2 + tui_3 + (1-t)ui_4 \\ i_1 &= \text{intensity}(x_{low}, y_{low}) \\ i_2 &= \text{intensity}(x_{low} + 1, y_{low}) \\ i_3 &= \text{intensity}(x_{low} + 1, y_{low} + 1) \\ i_4 &= \text{intensity}(x_{low}, y_{low} + 1) \end{aligned} \quad (\text{Eq. 4.34})$$

Where x_{low} and y_{low} are integer parts of numbers v_x and v_y , respectively. Because pixels of the original image are creating a uniform square grid it can be used as a simplified definition of values t and u . Values t and u are fractional parts of the numbers v_x and v_y . After the interpolation of all marked points, the distortion is done. As it is known from the previous paragraphs, the pixel coordinates are integers, but the model returns real numbers. To achieve a more precise distortion it is better to use the interpolation to all points of the image (with the exception of that in area A). However, the computing of inverted values and interpolation takes time, so this fully interpolated variant is slower than the applied model, which interpolates only the required points. [1]

4.5 Database Generation

In addition to a single image processing it is often required to have a possibility to create a database (or a batch) from several images. Massive processing is one of the biggest advantages of synthetic fingerprints. The preparation of damage could be time-consuming (especially for swipe sensors) because there has to be a way to save the damage. For saving, the .xml format was chosen. All input data (sensor information, damage information, etc.) is saved, the only exception is the image itself. After all, that path to the folder with images has to be defined that way it would be possible to damage each image once. Often it is interesting to combine damages. For this, there is the **full combination** setting. This means that all damages will be combined to all damages. If one image is taken then all one damage combinations are done to it, all two damages combinations, etc. Let *num* denote the number of damaged images generated from one source image, *d* the number of damages, and *FC* the full combination setting.

$$num_{FC} = \sum_{i=1}^d \binom{d}{i} \quad (\text{Eq. 4.35})$$

That is correct, but a rather complicated formula. Because it is a combination of all damages presented by all numbers of them, it is similar to a powerset. Here is a much clearer formula:

$$num_{FC} = 2^d \quad (\text{Eq. 4.36})$$

The only small difference between Eq. 4.35 and Eq. 4.36 is that the latter also counts no damage at all (original image) as one of the variants. Now there is a difficult decision. If the full combinations of given damages are done then it could be problematic if the same type of damage cannot be applied to the same image, i.e., it makes no sense that the image would be distorted in two different ways at the same time. That is the reason for the **restricted combination** setting. This setting allows only one of each damage type in each combination. That way all damage combinations can happen without the cases when there would be, for example, two distortions in one image. In this case the number (*num*) of images generated by the number of damages from a particular damage type (*d*) from one image is a bit lower. Let *t* denote number of damage types and *RC* as the restricted combination setting.

$$num_{RC} = \prod_{i=0}^{i < t} (d_i + 1) \quad (\text{Eq. 4.37})$$

The +1 in the formula is to include no damage as an additional damage in all damage types. Similarly to Eq. 4.36, this also counts with no damage at all as one of the variants. When discussing possible damage combinations, fingerprint sensors could be damaged in two places at once. In that case the image should be damaged by two damages of the same type. Without explicit semantics of the combinations of all damage types, there is no general solution to this problem. Because of that, the last level of combinations was introduced – it is **no combination**. Using this, all damages that are loaded will be made in that order to all images. This is a possibility for some really specific combination to be done to an image database. In this setting all images only have one damage impression. In Figure 4.12, the batch input GUI could be seen.

The first damage in the list of all damages to use is very important, because it sets the order of damages. This order is defined for each sensor and it could be a little different for each sensor. It could also happen that some damages are not even listed in some sensors. If damage is not listed, then it is used as a special damage type rest (no matter what its real damage type is). Structures for batch generation are prepared in the defined order. After that, all combinations are generated by recursion.

Batch

#	Damage Name	Damage Type	Path
1	PM_slowhigh_to_low	Pressure & Moisture	C:\PM_slowhigh_to_low.xml

< >

Order of damages is taken from the sensor order of first loaded damage.

Combinatorial level	Each image will produce
<input checked="" type="radio"/> Full - every damage combination	1 image
<input type="radio"/> Restricted - every damage type combination	1 image
<input type="radio"/> None - only listed combination used	1 image

Folder with Images

Figure 4.12: Examples of real fingerprints from swipe sensor.

5 Swipe Sensor Damage Simulation

This chapter is focused on altering existing synthetic fingerprints in the master fingerprint phase to a synthetic fingerprint that will look like a real fingerprint created by a swipe fingerprint sensor. At the moment, there is no conclusive research on sensor specific variations of synthetic fingerprints. The only efforts made in this area were some projects that tried to create a sensor specific background. A description of swipe sensors could be found in Subchapter 2.3. The primary point is that the image acquired by swipe sensors is a combination of several smaller images. These smaller images are from different parts of the finger which is doing a swipe motion over the sensor unit. The reconstruction algorithm [23] [24] [25] [26] [27] [28] [65] then merges these images into one final image.

5.1 Damage Analysis

The complexity of fingerprint acquisition using a swipe sensor is reflected in determining the new influential factors for this type of sensor, and a skilled and trained user is capable of creating high-quality results with it. That is because a reconstruction algorithm, which is merging individual acquired images, can repair some damages done by the user. On the other hand, unskilled users can in the same way create more damages because of bad cooperation between them and the algorithm. When observing skilled and basic users using a swipe sensor, it was found that it is possible to simulate a swipe sensor similar to the touch sensors. The main difference is that each factor can appear in each small image, which are later merged to the final acquired image. That means that *pressure* can be very high in the first few images and very low in the last one. The reason behind this type of behaviour is that the user was applying high *pressure* at the start of a swipe and in the end, he stops applying *pressure* because he thought that the sensor is not collecting data. This type of behaviour can be generalized to other factors as well. Various damages from swipe sensors can be seen in Figure 5.1.



Figure 5.1: Examples of real fingerprints from swipe sensor.

Bearing that in mind, it is also very easy to extend the effects of *non-cooperative behaviour*. Swiping a finger askew across the sensor, a very quick swipe, trying to swipe in the opposite direction, or trying to swipe two fingers combined are only some ways of clearly exhibiting intentional, non-cooperative behaviour, in spite of the fact that it could also be the *contact region* damage (see Subchapter 3.2). Listed factors are so intentional that they should be included as *non-cooperative behaviour*. On the contrary, there are factors that belong to the *contact region* even when they could be seen as *non-cooperative behaviour* because they are really common among unskilled users. Users often try to create a good fingerprint image but swipe incorrectly. Their motion often neglects important parts of the fingerprint (like deltas), which are either missing or they are at the edge of the fingerprint (which is common in the case of the thumb). Another example is when users try really hard to create a good fingerprint image and they acquire another phalange of the finger. The *sensor condition* group of influential factors can also be expanded. Swipe sensors are often used because they are small and cheap. A cheap sensor can be made even cheaper by using even smaller sensing unit. However, some sensors are then too small to fit even index fingers, which are the most commonly used. [29]

From the description above, important influential factors can be defined. These factors can be divided into two groups: the first one with the factors that are altering damages common with the touch sensors, and the second group with factors that are completely new in swipe sensor usage. The first group contains already known damages like *pressure*, *contact region*, and *skin distortion* used in swipe mode (see Subchapter 5.2). In the second group, this work focuses only on *narrowing* the fingerprint image. Successful simulation of these factors leads to creating a fingerprint which will be close to a fingerprint taken from the swipe sensor.

5.2 Swipe Mode

This group focuses on the already created touch fingerprint damages, however, those are used in a swipe mode. Swipe mode is basically the recreation of a reconstruction algorithm. The algorithm works in this way: the sensor usually acquires a short and wide part of fingerprint (based on the size of sensing unit). This small image is called a *slice*. By swipe motions, the sensor gets a lot of slices and it stores them in the order of their acquirement. Its responsibility is to reconstruct the fingerprint image based on correlation, the swipe speed, and the known parameters of sensing unit (as can be seen on Figure 5.2). Synthetic fingerprints are generated without slices. It is necessary to take steps in the inverse order. Synthetic images will be divided into slices. Each of them will be treated as an independent image. After that, all slices will be merged together.

An important issue is how the input data will be transferred to these slices. There are several solutions. **Fully-automatic settings** will be defined by the first (seed) settings as well as some optional information about the trend of the values. **Semi-automatic settings** will be defined by the input data spread out on a set of slices. **Manual settings** will be defined by providing input data for each slice. Each option has its own pros and cons. Fully-automatic settings are the easiest to use, but very difficult

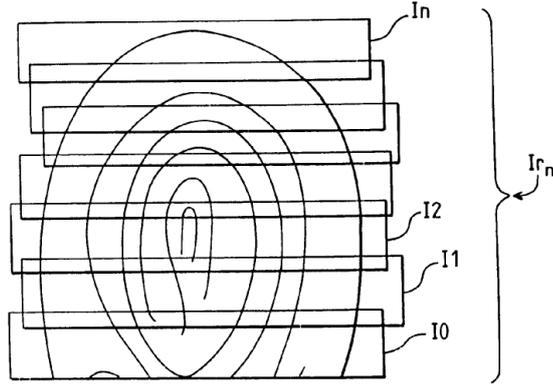


Figure 5.2: Example of assembled slices to create a fingerprint (taken from [65])

to simulate. It is certain that there has to be a limit on how much the input can change in the two following slices. It is possible to make a large, sudden change. However, to make these changes in one swipe motion is rather difficult on the fine motor skills of the user. Also, maximum change in input values between the two following slices is influenced by an overlap of the slices (closer the slices, closer the values), the type of input value (skin elasticity would probably be constant, on the other hand the rotation of the finger can change), the type of damage (generally, it is easier to make a change in pressure than in skin distortion, for example), etc. Although it could be done, it was not chosen as an option for this work. The semi-automatic solution is the middle ground among these choices. It is still simpler to input all the values, but the big question is how to approximate values among them. If something other than the default approximation is used, then it will probably be necessary to input all the values, which leads to a manual setting solution. It provides the biggest flexibility, and a disadvantage of this solution is that all input values must be filled, so the number of slices should stay at some reasonable number. Nevertheless, manual settings will be used in this work.

5.2.1 Swipe Sensor Definition in Swipe Mode

Generally, there are only a few boundaries regarding how many slices there are in one image. It is dependent on user swipe speed and sensing unit height. This general swipe mode would be difficult to use, and the user would have to pay attention if another slice is a few pixels away or shifted to the right or really far down. For that reason, swipe mode is simplified. Slices are uniformly distributed in the image, and they are not moved in horizontal axis. After this simplification, the variables needed for this swipe mode can be defined. Primarily, it is the *height* and *overlap* of the slices. With the given synthetic *image height* it is possible to determine how many slices there will be in the image (Eq. 5.1).

$$\begin{aligned}
 Slices_{number} &= \frac{Image_{height} - Slices_{height}}{Slices_{height} - Overlap} + 1 \\
 &\text{where } Slices_{number} \in \mathbb{N}, \\
 &0 < Slices_{height} \leq Image_{height} \text{ and} \\
 &0 < Overlap < Slices_{height}
 \end{aligned}
 \tag{Eq. 5.1}$$

In some cases, for example using a stored damage to another image, it is beneficial to have also percent representation of *slices height* and *overlap*. That way the *number of slices* even with different *image height* will stay the same ± 1 . Note that images usually have height in hundreds of pixels, in case of images with extremely low height, e.g., 1 or 2 px the difference between *number of slices* could be higher. If needed *number of slices* is rounded down. When that happen there is a part of original image which is not covered by slices. If it was an image from real sensor that part would be non-existent. In this case it was decided to fill this remainder with *background colour* of sensor. Height of remainder (*rem*) can be computed as (Eq. 5.2):

$$Rem = Image_{height} - [(Slices_{number} - 1)(Slices_{height} - Overlap) + Slices_{height}] \quad (\text{Eq. 5.2})$$

After knowing all this information, the image can be easily divided into slices. All of them are damaged independently. When damage is defined in swipe mode, each slice has its own settings for damage. If the damage is applied, all slices are according to these settings damage separately. Now all there is left to do is to merge all of the damaged slices together to one image. Unlike the real reconstruction function, there is no need to calculate correlation between slices. The position of slices in the image is already known.

Lastly, the merging function has to be defined. It reconstructs images line by line (a line is a slice with a height of 1 px). Each line of the image has 1 to n contributing line in slices. Presume that $line_{(x, pos)}$ defines the pixel in position pos in x -th line that contributes to the image, and $point_{(pos)}$ is the pixel on position pos in the current line of the image. For each colour channel (red, green, blue, alpha) the value is defined as (Eq. 5.3):

$$Point_{(pos)} = \frac{\sum_{i=1}^n Line_{(i, pos)}}{n} \quad (\text{Eq. 5.3})$$

After passing all lines of images and all positions in them, the merged image is done. It is important to mention that this process is not semantically right for all damages (e.g. the damage sensor will be damaged in all slices, not just one).

For testing slices, the settings were set as can be seen on Figure 5.3, with a 44 px slice height and 34 px overlap of slices. In real applications the redundancy (overlap) is relatively high. That is to cover fast swipes and to have data for reconstruction if some slices are low quality or damaged. High overlap

Height in pixels	Overlap in pixels	Image height
44	34	440
Height in percent	Overlap in percent	Image height covered
10.00	77.27	434
Background colour	<input type="checkbox"/>	
Number of slices	40	OK

Figure 5.3: GUI for slices settings when using swipe sensor.

values lead to an enormous number of slices, which is not reasonable with the manual settings of values. Using of larger slices height allows bigger overlap. Designed values meet all these requirements, overlap is rather high and number of slices is reasonable. The next subchapters will discuss the possible usage of touch-based damages that were mentioned in Subchapter 4.4. It also shows the specific damage settings used for testing in Subchapter 5.5.

5.3 Damages Exclusive to Swipe Sensors

This category describes the damages that are typical for swipe sensors. Of course, with some extra effort and *non-cooperative behaviour* some of these damages can be replicated on touch sensors. Further restriction is that damages in this category are not altered touch damages – these were described in Subchapter 5.2.

The most important damage is the **narrow sensor**. The greatest advantage of this technology is its price. To further leverage this advantage, the sensing unit is not only short, but also narrow. Only a portion of the width of the finger is scanned. If the finger is swiped askew then the behaviour is dependent on the reconstruction algorithms. Some of them are able to detect this anomaly and move the next slice down and to the side (so translation in x and y axis). This results in images that have higher widths than the sensing unit. However, most of the fingerprint images have a width the same as the sensing unit. The exception is the part where there is information from more slices, and the reconstruction algorithm is able to merge several of them (the width of the image would be a little bit higher than the sensor unit width).

Closely connected to the movement and unskilled user is the acquirement of **another phalange**. Acquired images often start in the middle of the fingerprint and continue over the joint as the user is trying to make a long swipe. This is a very special type of “damage”. To simulate this, it is needed to add a ridge line print from the inner side of the joint and the second phalanx. When prolonging a synthetic fingerprint image, one can rely on the fact that the next phalange of the finger usually has a very flat arch class. The best possible solution would be to additionally generate this fingerprint class below the current fingerprint image with a gap representing the transition between phalanges. This transition is often accompanied with a few deep wrinkles. Additional generation without the original orientation field, cores, and deltas could be problematic. The contact region of the master fingerprint can be set higher, and then the important part (like delta) should be generated. After that comes the additional generation, and because of that, this type of damage is not simulated.

Faults or **exploitation** of reconstruction algorithm is heavily dependent on the specific implementation and properties of the algorithm used inside of sensor. When someone has agile fingers, it is possible to swipe two fingers at once. Some reconstruction algorithms work continuously, so with some dexterity it is possible to make a very long fingerprint image with several fingers. Part of the algorithm needs to approximate swipe speed. Whenever the motion of capture is done slow at the start and then swiftly accelerates, it is possible to also create unrealistically long images or ridges. Because of the similar reason as in previous point, this damage is also not simulated.

The last damage to be discussed is **motion blur**. Damage can be often seen on the edges of the fingerprint, and the categorization is rather unclear. Motion blur can be caused by changing skin

distortion or movement (translation, rotation) of the finger during the swipe, causing problems with the reconstruction algorithm. On the other hand, it could be caused by a fall of precision when a finger is getting out of range on the edge of the sensing area. It was determined to deal with the motion blur as a movement during the scan in Subchapter 5.4.4.

5.3.1 Narrow Sensor

This subchapter describes the implementation of damage done by the narrow sensor. The most important part is to determine how much the image should be narrowed (*narrow cut*). When the results are observed it can be seen that it does not look natural. Whenever the image is taken from a narrow sensor, the edges are not sharply cut. There is a small overlap between the clear image and the blank space where the sensor cannot take an image. Width of the overlap is the next input to this damage simulation (*smooth width*). The last piece of information is the position of the centre of the sensor (*central width*). After that, the *left* and *right* variable is determined (Eq. 5.4).

$$\begin{aligned} \text{left} &= \text{Width}_{\text{central}} - \text{Width}_{\text{image}} \frac{\text{Narrow cut}}{100} \\ \text{right} &= \text{Width}_{\text{central}} + \text{Width}_{\text{image}} \frac{\text{Narrow cut}}{100} \end{aligned} \quad (\text{Eq. 5.4})$$

If any of them is lower or higher than the width of the image, it is set to this minimum (or maximum). Pixels in the area from *left* to *right* have the same colour as they had. Pixels in the area from the *image start* to (*left - smooth width*) value are white. Pixels within the area from the (*right + smooth width*) value to the *image end* are also white. The remaining pixels (i.e. inside the *smooth width* range) have an additional alpha channel added to them. This makes transparency which softens the edges. It is done gradually by this formula (Eq. 5.5):

$$\text{Alpha} = \frac{\text{maximal intensity}}{\text{smooth width}} k \quad (\text{Eq. 5.5})$$

where *k* is the number of pixels from the start of the smooth area (i.e. *left* or *right*). Figure 5.4 shows how the fingerprint will be narrowed along with the resulting image. This damage simulation is now looking very similar to a real fingerprint image. It can be added as touch-based damage to the application.

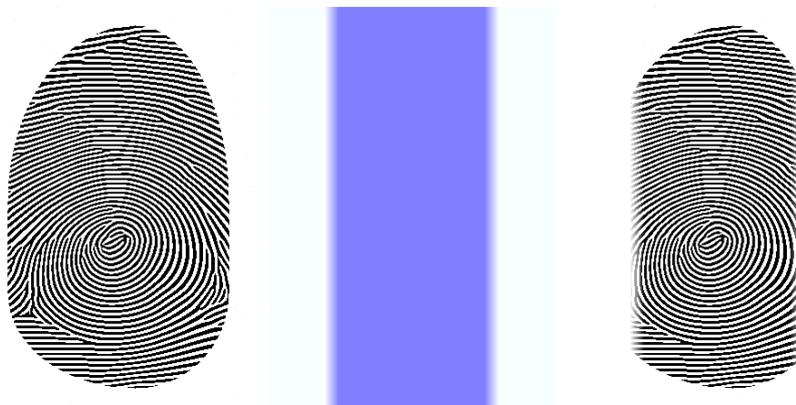


Figure 5.4: Narrow sensor simulation (before damage, preview and after applying the damage).

5.4 Examples of Damages

To create examples of swipe sensor damages, the suitable database required for this task was acquired. Focus was on acquiring fingerprints from all fingers and unskilled users. Using thumb, ring, or little fingers often causes faults because users are not used to swiping with these fingers. Contrary to usual habits, no images were deleted because of low quality. Users were told how to use the sensor, but they were not instructed to do some intentional damage. The database contains over 1,000 fingerprints from approximately 100 users. Fingerprints are from the Eikon II swipe capacitive sensor. Images from this database are used as real examples later on in this subchapter.

As stated in Subchapter 5.3, swipe sensors are usually narrow. On the contrary, synthetic fingerprints are usually generated as undamaged live prints (oval shaped) [42]. To create realistic-looking fingerprints it is better to have them narrowed. For that, a *narrow sensor* touch-based damage is used. Settings of narrow damage: smooth region width 5 px, narrow cut 25 %, central width 160 px (half of the image). For the sake of comparing all damages in this work, the wide (normal width) and narrowed synthetic fingerprint are tested. It is clear that narrow fingerprints are looking more realistic and the effects of individual damages are easier to see. Also, some damages use absolute values for their settings. In these cases, two variants of settings have to be defined for both narrow and wide databases. Find out more about databases used for testing in Subchapter 5.5.1.

5.4.1 Pressure and Moisture in Swipe Mode

Pressure is one of the easiest place to see damage. In addition, a change of pressure is very frequent in a swipe motion. There are a lot of combinations of greater or lower pressures in the database. In the end, these seven were chosen. The list contains the damage name, the testing shortcut, and a short description with settings. Settings is described as slice numbers (1–40) where 1 is the first slice on top of the image, and pressure settings is (-4 to +4) where -4 is extremely low pressure (for more information see Subchapter 4.4.2).

- **All low (pm0):** Caused by dry skin or low pressure in a whole swipe. Real and synthetic images can be seen in Figure 5.5bc. Settings: slices 1-40 pressure -1.
- **Extreme (pm1):** Extreme cases could be created by steady pressure changes when doing a swipe motion. Real and synthetic images can be seen in Figure 5.5de. Settings: slices 1-3 pressure -4, sl. 4-6 pressure -3, sl. 7-9 pressure -2, sl. 10-12 pressure -1, sl. 13-15 pressure 0, sl. 16-18 pressure +1, sl. 19-23 pressure +2, sl. 24-26 pressure +1, sl. 27-29 pressure 0, sl. 30-32 pressure -1, sl. 33-35 pressure -2, sl. 36-38 pressure -3, sl. 39-40 pressure -4.
- **High to normal (pm2):** Damage caused by keen users who started the motion with a lot of pressure and finished it with normal pressure. Real and synthetic images can be seen in Figure 5.5fg. Settings: slices 1-14 pressure +3, sl. 15-17 pressure +2, sl. 18-20 pressure +1, sl. 21-35 pressure 0, sl. 36-40 pressure -1.
- **Low to high to low (pm3):** Very common damage where the user has low pressure until the user's finger gets to the sensor. After that, the pressure is raised and when the finger is at the

edge of the sensor the pressure is lowered again. Real and synthetic images can be seen in Figure 5.6ab. Settings: slices 1-10 pressure -1, sl. 11-29 pressure +1, sl. 30-40 pressure -1.

- **Normal to low (pm4):** Similar to the high to normal variant. Instead of a keen user, there is usually a very cautious one whose pressure levels are lower. Real and synthetic images can be seen in Figure 5.6cd. Settings: slices 1-18 pressure 0, sl. 19-21 pressure -1, sl. 22-40 pressure -2.
- **Recurrent normal to low (pm5):** Cause by sudden drops of pressure or some slices that were unable to reconstruct correctly. Real and synthetic images can be seen in Figure 5.6ef. Settings: slices 1-12 pressure 0, sl. 13 pressure -1, sl. 14-18 pressure -2, sl. 19 pressure -1, sl. 20-24 pressure 0, sl. 25 pressure -1, sl. 26-31 pressure -2, sl. 32 pressure -1, sl. 33-40 pressure 0.
- **Slowly high to low (pm6):** This is a combination of two factors. The user either applies greater pressure or has moist fingers. After that, when the user feels a change in structure (start of the sensing unit) an even greater pressure is applied. Real and synthetic images can be seen in Figure 5.6gh. Settings: slices 1-8 pressure +1, sl. 9-14 pressure +2, sl. 15-17 pressure +1, sl. 18-33 pressure 0, sl. 34-40 pressure -1.

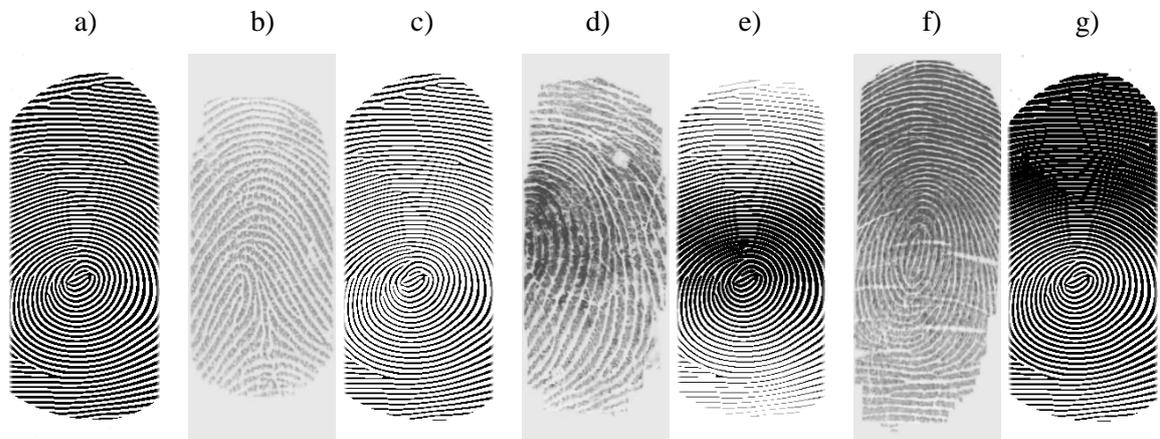


Figure 5.5: Examples of pressure damage (a – original synthetic image, b, d, f – real images, c, e, g – damaged impressions).

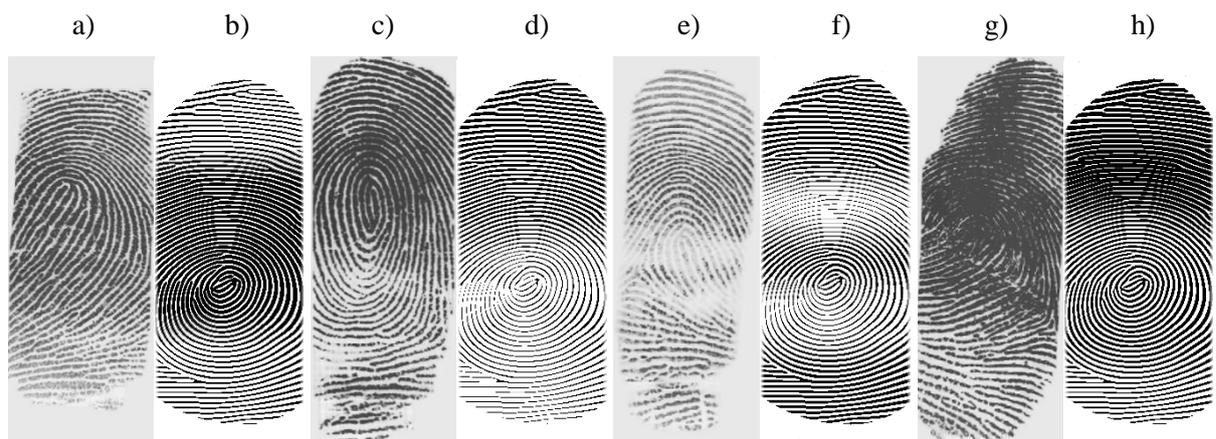


Figure 5.6: Examples of pressure damage 2 (a, c, e, g – real images, b, d, f, h – damaged impressions).

5.4.2 Narrow Sensor in Swipe Mode

Usage of the narrow sensor in swipe mode can simulate different contact regions. The exact results of contact region damage in swipe sensors are dependent on the used reconstruction algorithm. In this case (same as in the acquired database), fixed sensing region is assumed. Wrong (not complete) contact regions are in a high percentage of images. The use of non-traditional fingers in databases leads to a lot of these damages. These are often made by the little finger or thumb.

In the end, 11 damages were chosen. The list contains the damage name, the testing shortcut, and a short description with settings. Settings are described as slice numbers (1–40) where 1 is the first slice on the top of the image, and the narrow settings, which contains the smooth region width (SRW) in px, *narrow cut* (NC) in percentage, and *central width* (CW). For more information see Subchapter 5.3.1. For simplicity, some slices are marked as no damage (that means SRW 0, NC 100 % and CW in half of image width). This definition is little bit hard to use when inspecting real images, especially when images are cut from both sides. The following formulas (Eq. 5.6 and Eq. 5.7) transform percentage damages (cut from left and right) to define settings.

$$\text{Narrow cut} = \frac{(100 - \text{Cut}_{\text{right}}) - \text{Cut}_{\text{left}}}{2} \quad (\text{Eq. 5.6})$$

$$\begin{aligned} \text{Central width} = & \frac{\left[(100 - \text{Cut}_{\text{right}}) \frac{\text{Image}_{\text{width}}}{100} \right] - \text{Cut}_{\text{left}} \frac{\text{Image}_{\text{width}}}{100}}{2} \\ & + \text{Cut}_{\text{left}} \frac{\text{Image}_{\text{width}}}{100} \end{aligned} \quad (\text{Eq. 5.7})$$

If the results have a decimal point, it is rounded because the final value must be an integer. In some examples, settings are defined by slices, left, and right percentages cut instead of the previously stated settings. For further simplification, a shortcut for image width (IW) will be used. Images in the narrow database have IW 170, images in standard database IW 320. In all damages, with exception to no damage, the SRW is 3 px.

- **All sideways sharp (narr0):** Both sideway damages were created predominantly when using a thumb. Users hold the sensor in hand and swipe the thumb over it, which is what created the sideway swipe. Real and synthetic images can be seen in Figure 5.7bc. Settings: slices 1-15 CW IW, sl. 16-24 no damage, sl. 25-40 CW 0. Slices 1-15 NC 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96; slices 25-40 NC 99, 98, 97, 95, 93, 90, 87, 83, 79, 75, 70, 65, 59, 53, 46, 39.
- **All sideways steady (narr1):** This steady variant is a little more frequent in the database as opposed to the sharp variant. Real and synthetic images can be seen in Figure 5.7de. Settings: slices 1-20 CW 0, sl. 32-40 CW IW. Slices 1-20 NC 65, 66, 67, 68, 70, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 82, 84, 85, 86, 87; slices 32-40 NC 77, 75, 73, 71, 69, 67, 65, 63, 61. Slices 21-31 left% 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21; right% 12, 11, 9, 8, 7, 6, 5, 4, 3, 2, 1.
- **Cutdown (narr2):** Caused by a sudden lift of the finger when the user thought that the swipe was done. Real and synthetic images can be seen in Figure 5.7fg. Settings: slices 1-35 no damage, 36-40 SRW 0, NC 0, CW 0.

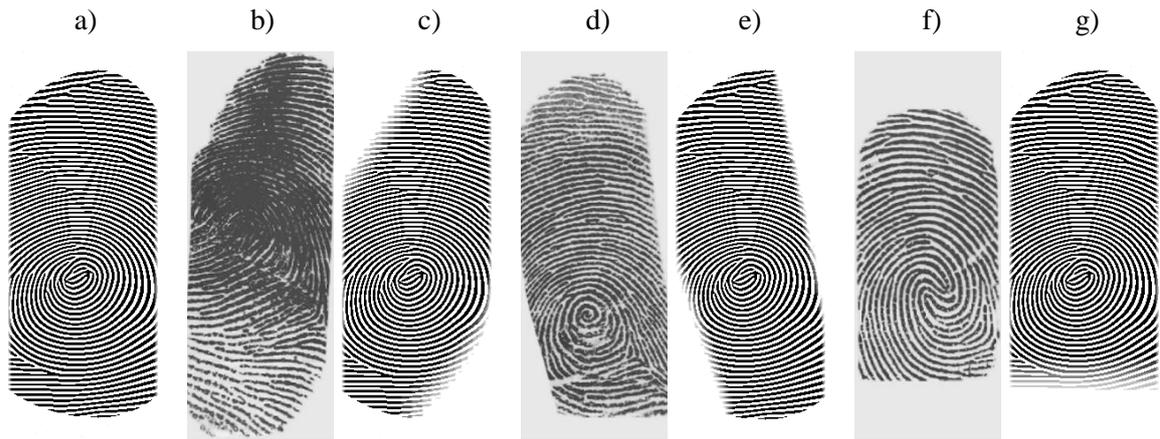


Figure 5.7: Examples of narrow damage (a – original synthetic image, b, d, f – real images, c, e, g – damaged impressions).

- **One side (narr3):** Real and synthetic images can be seen in Figure 5.8ab. Settings: slices 1-14 CW IW, sl. 15-21 no damage, 22-40 CW IW. Slices 1-14 NC 52, 55, 59, 62, 66, 69, 73, 76, 80, 83, 87, 90, 94, 97; slices 22-40 NC 98, 96, 94, 92, 90, 88, 86, 84, 82, 80, 78, 76, 74, 72, 70, 68, 66, 64, 62.
- **Side zigzags (narr4):** Real and synthetic images can be seen in Figure 5.8cd. Settings: slices 1-8 no damage, 23-40 no damage, 9-22 left% 15, 28, 39, 48, 55, 48, 39, 28, 15, 0, 0, 0, 0, 0, right% 0, 0, 0, 0, 0, 7, 14, 28, 42, 56, 42, 28, 14, 7.
- **Tip bottom jumpy (narr5):** Real and synthetic images can be seen in Figure 5.8ef. Settings: slices 1-20 CW 0, sl. 32-40 CW IW. Slices 1-20 NC 65, 66, 67, 68, 70, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 82, 84, 85, 86, 87; slices 32-40 NC 77, 75, 73, 71, 69, 67, 65, 63, 61. Slices 21-31 left% 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21; right% 12, 11, 9, 8, 7, 6, 5, 4, 3, 2, 1.
- **Tip bottom one side (narr6):** Real and synthetic images can be seen in Figure 5.8gh. Settings: slices 1-20 CW 0, sl. 32-40 CW IW. Slices 1-20 NC 65, 66, 67, 68, 70, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 82, 84, 85, 86, 87; slices 32-40 NC 77, 75, 73, 71, 69, 67, 65, 63, 61. Slices 21-31 left% 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21; right% 12, 11, 9, 8, 7, 6, 5, 4, 3, 2, 1.

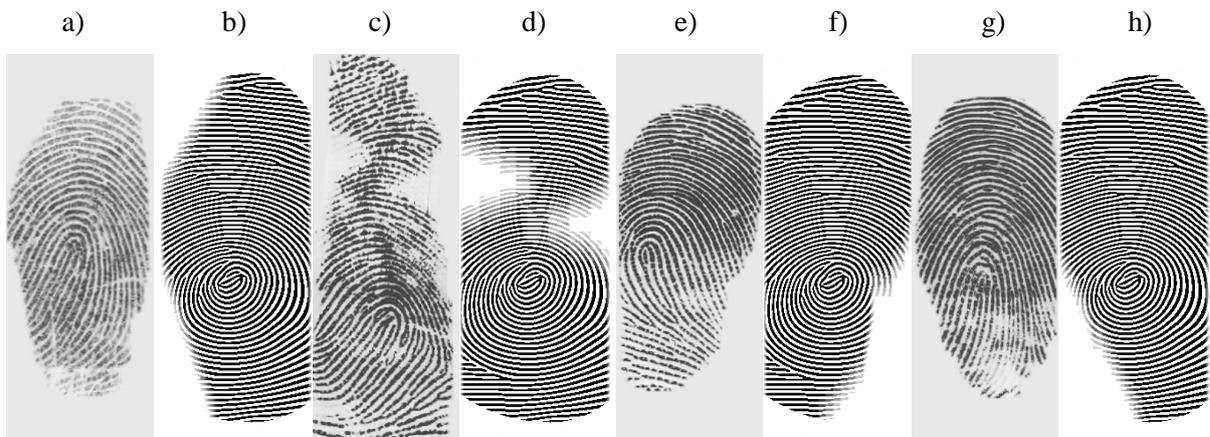


Figure 5.8: Examples of narrow damage 2 (a, c, e, g – real images, b, d, f, h – damaged impressions).

- **Tip bottom standard (narr7):** Real and synthetic images can be seen in Figure 5.9ab. Settings: slices 1-17 no damage, slices 18-40 left% 0, 0, 0, 1, 3, 4, 6, 7, 9, 10, 12, 13, 9, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, right% 3, 5, 7, 8, 9, 10, 14, 17, 17, 18, 18, 19, 19, 20, 21, 23, 25, 27, 29, 31, 33, 35, 36.
- **Tip both sides (narr8):** Real and synthetic images can be seen in Figure 5.9cd. Settings: slice 1-15 left% 40, 35, 30, 26, 22, 19, 17, 16, 15, 12, 9, 6, 3, 0, 0, right% 30, 26, 22, 19, 16, 13, 11, 9, 7, 5, 4, 3, 2, 1, 1, sl. 16-19 no damage, sl. 20-40 left% 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 7, 9, 11, 13, 15, 17, 19, right% 4, 5, 6, 7, 8, 13, 14, 15, 16, 18, 21, 25, 30, 36, 44, 45, 46, 47, 48, 49, 50.
- **Tip top round (narr9):** Real and synthetic images can be seen in Figure 5.9ef. Settings: slices 19-40 no damage, slices 1-19 left% 52, 43, 35, 28, 21, 15, 10, 6, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, right% 35, 31, 28, 25, 22, 19, 16, 14, 12, 10, 8, 6, 5, 4, 3, 2, 1, 1.
- **Tip top sharp (narr10):** Real and synthetic images can be seen in Figure 5.9gh. Settings: slices 12-40 no damage, slice 1-12 left% 40, 38, 32, 24, 16, 8, 0, 0, 0, 0, 0, 0, 0, right% 40, 38, 36, 32, 28, 24, 20, 16, 12, 8, 4.

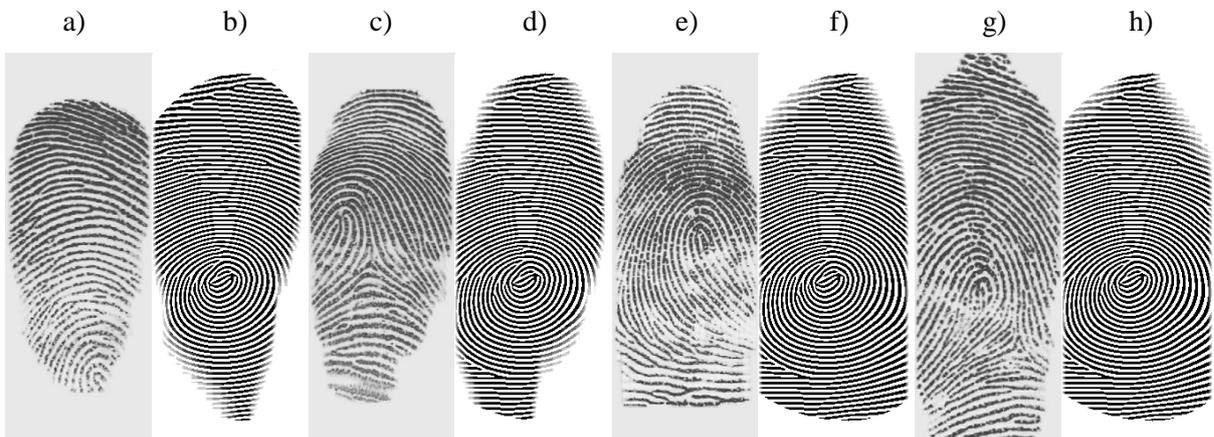


Figure 5.9: Examples of narrow damage 3 (a, c, e, g – real images, b, d, f, h – damaged impressions).

5.4.3 Damaged Sensor in Swipe Mode

Swipe sensors, mainly because of their size, are more resistant to damage. Based on the previous damages and the simplification of fixed sensing, region expected behaviour can be simulated. Two types of damage were chosen for this simulation. This damage is the one which has the same settings for all slices. If the sensor is damaged it will be in all slices and in the same place. As in previous parts, the list contains the damage name, the testing shortcuts, and a short description with settings. Settings is described as damaged settings which contains line length (1-15), direction (0-359°), and start position x , y in pixels. For more information see Subchapter 4.4.1. Because the start point coordinates variable is defined in absolute values, separate settings are required for standard and narrow images in the database. Experience from touch sensors reveals that damage is usually close to the edge of the sensor (respectively, protective glass). Designed damages show both short and long cracks.

- **Long, narrow (dmg0):** The synthetic image can be seen in Figure 5.10b. Settings: length 15, direction 98° , start point x, y 0, 22.
- **Long, wide (dmg0):** The synthetic image can be seen in Figure 5.10c. Settings: length 9, direction 98° , start point x, y 0, 22.
- **Short, narrow (dmg1):** The synthetic image can be seen in Figure 5.10d. Settings: length 6, direction 300° , start point x, y 170, 3.
- **Short, wide (dmg1):** The synthetic image can be seen in Figure 5.10e. Settings: length 1, direction 300° , start point x, y 320, 3.

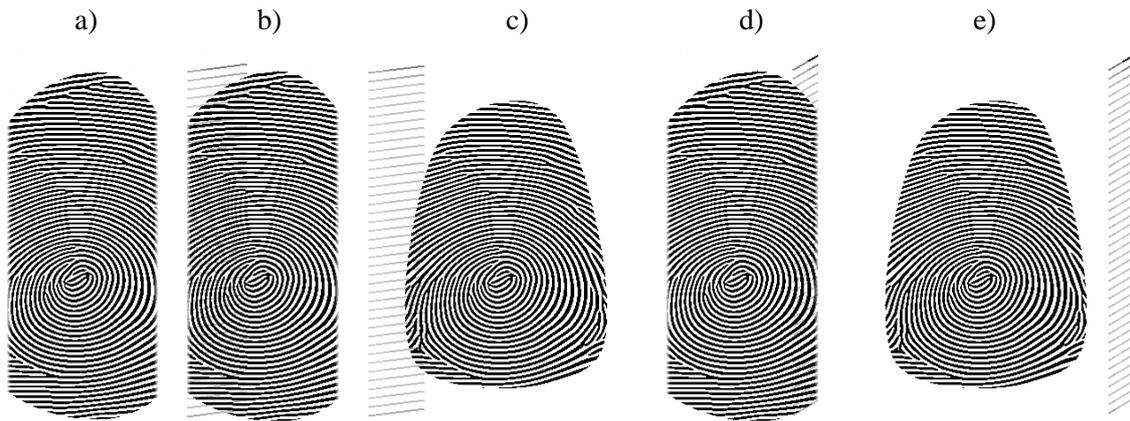


Figure 5.10: Examples of damaged sensor (a – original synthetic image, b, c, d, e – damaged impressions).

5.4.4 Distortion in Swipe Mode

The final example is using skin distortion. This damage is probably the worst to correctly simulate. It is certain that skin has to be distorted when doing the swipe motion. Distortions are also hard to see in real images. On the other hand, if the finger is moving sideways or too fast, for example, there has to be distortion and it can be seen as motion blur. That is because the correlation part of the reconstruction algorithm finds non-distorted parts which are consistent with previous slices as well as the distorted parts which are not. Some mistakes or inaccuracies will be surely made when this happens. Distortion could also mean that part of the skin is not touching the sensing area, thereby creating contact region damage. The chosen examples are a little extreme. On the other hand, they can be easily seen and the motion blur (with translation or rotation) is obvious in each example. Real images with motion blur and other damages can be seen in Figure 5.11efg.

In the end, three damages were chosen to be simulated. The list contains the damage name, the testing shortcut, and the short description with settings. Settings are described as slice numbers (1–40) where 1 is first slice on top of the image and distortion settings that contains skin elasticity coefficient (1–30), rotation (-30 – 30°), translation d_x, d_y in pixels (could be negative), and usage of full interpolation. For more information see Subchapter 4.4.3. For simplicity, the skin elasticity coefficient will be 10 for every slice as well as usage of full interpolation.

- **Extreme (dis0):** In this example, an extreme case of rotation is used. It would mean that the user made a 40° turn during the swipe. Some translation is also applied. The synthetic image can be seen in Figure 5.11b. Settings: slices 1-40 translation d_x, d_y 15, 2; rotation -20, -18, -17, -16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20.
- **Move X (dis1):** This example shows the steady translation on the x axis and small rotation. The synthetic image can be seen in Figure 5.11c. Settings: slices 1-3 rotation 0, translation d_x, d_y 0, 0; sl. 4-6 rotation 1, translation d_x, d_y 1, 0; sl. 7-9 rotation 2, translation d_x, d_y 2, 0; sl. 10-12 rotation 3, translation d_x, d_y 3, 0; sl. 13-15 rotation 4, translation d_x, d_y 4, 0; sl. 16-18 rotation 5, translation d_x, d_y 5, 0; sl. 19-21 rotation 6, translation d_x, d_y 6, 0; sl. 22-24 rotation 7, translation d_x, d_y 7, 0; sl. 25-27 rotation 8, translation d_x, d_y 8, 0; sl. 28-30 rotation 9, translation d_x, d_y 9, 0; sl. 31-33 rotation 10, translation d_x, d_y 10, 0; sl. 34-36 rotation 11, translation d_x, d_y 11, 0; sl. 37-40 rotation 12, translation d_x, d_y 12, 0;
- **Move Y (dis2):** Gradually raising translation in y axis shows uneven swipe speed. The synthetic image can be seen in Figure 5.11d. Settings: slices 1-4 rotation 0, translation d_x, d_y 0, 0; sl. 5-8 rotation -1, translation d_x, d_y 0, (0, 1, 1, 1); sl. 9-12 rotation -2, translation d_x, d_y 0, (1, 1, 2, 2); sl. 13-16 rotation -3, translation d_x, d_y 0, (2, 2, 3, 3); sl. 17-19 rotation -4, translation d_x, d_y 0, (3, 3, 4); sl. 20-23 rotation -5, translation d_x, d_y 0, (4, 4, 5, 5); sl. 24-27 rotation -6, translation d_x, d_y 0, (5, 6, 6, 7); sl. 28-31 rotation -7, translation d_x, d_y 0, (7, 8, 9, 10); sl. 29-35 rotation -8, translation d_x, d_y 0, (8, 9, 10, 11, 13, 15, 18); sl. 36-39 rotation -9, translation d_x, d_y 0, (21, 24, 28, 32); sl. 40 rotation -10, translation d_x, d_y 0, 36;

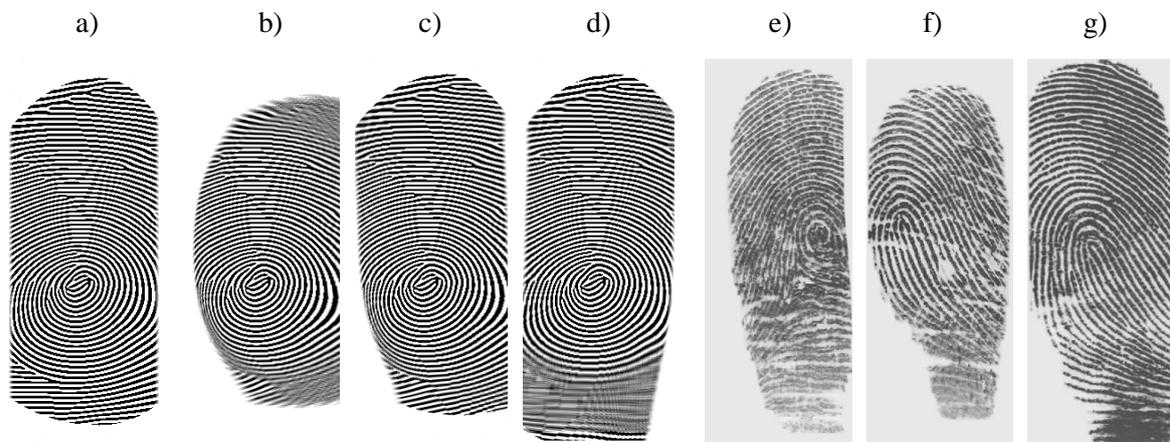


Figure 5.11: Examples of skin distortion damage (a – original synthetic image, b, c, d – damaged impressions, e, f, g – real images).

5.5 Evaluation

The first part of the evaluation can be found in the previous subchapters. It was done by describing the solution and its similarities with real applications and also by comparing damaged images with real ones. However, in order to quantify how much damage was done to the synthetic fingerprints, it is

necessary to find out the quality of them. Several methods for quality determination are used and described in the following subchapters. Even before that, however, it is necessary to present the testing database.

5.5.1 Synthetic Database Used for Evaluation

All available synthetic generators were used to generate testing databases. This database has 150 synthetic images: 60 from SFinGe [40], 60 from Anguli [41], and 30 from SyFDaS. Fingerprints were used in the state of master fingerprint. That means images without any damages, with exception to contact region (i.e. images can have an oval shape). The database was generated with natural fingerprint class distribution (arch 3.7 %, left loop 33.8 %, right loop 31.7 %, tented arch 2.9 %, whorl 27.9 % [66]). The precise distribution of image classes is in Table 5.1.

Table 5.1: Distribution of classes in the testing database.

	Anguli	SFinGe	SyFDaS	Sum
Arch	3	2	1	6
Left Loop	20	20	10	50
Right Loop	19	19	10	48
Tented Arch	1	2	1	4
Whorl	17	17	8	42
Sum	60	60	30	150

Unfortunately, each generator creates images of different sizes. To unify that, all images were scaled or cropped. The unified resolution was set to 320 px width and 440 px height. Images from Anguli were expanded to the final resolution and scaled to 90 %. Images from SFinGe were generated with an acquisition area 14.6 mm to 19.5 mm with 500 dots per cm. Generated images were set to full size and the grey areas were deleted. This image was then scaled to 88 % and cropped to the final resolution. Images from SyFDaS were generated with resolution of 320 px to 471 px. After that they were scaled to 140 % and cropped to the final resolution.

5.5.2 Methods Used for Quality Measurement

Quality measurement is the second step of verifying the results. The first step was a visual check against real fingerprint images. The premise is simple – damaged fingerprints should be of lower quality than original images. Three different measurement methods are used here. The first one is the *NEUROtechnology VeriFinger*⁵ – it is commercial software used primarily for fingerprint recognition, however, a part of the algorithm quality is also determined. This quality measurement and comparison score are used. The second method is from the National Institute of Standards and Technology (NIST) called *NFIQ*⁶ (NIST Fingerprint Image Quality). This algorithm is the only standard used for quality measurement. It has its flaws, which is probably reason why NFIQ 2.0 is in development. Nevertheless, the development of the new version is not yet complete and the draft version shows some flaws when

⁵ <https://www.neurotechnology.com/verifinger.html>

⁶ <https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis#NFIQ>

dealing with the synthetic images as well (for more information see [67]). The last method used is the algorithm of quality measurement designed by Mr. *Oravec* [68]. His method solves some of the problems of the NFIQ method.

5.5.2.1 NEUROtechnology VeriFinger

There is not a lot of information about the algorithms used. Nevertheless, that is understandable because it is a commercial product. Even the manual for VeriFinger's API [69] has almost no information. The only available source is the description enclosed to the results of the MM_FV 5.5 algorithm (it is probable that the old version of VeriFinger used this algorithm) in FVC-onGoing (Fingerprint Verification Competition) [70]. This competition evaluates algorithms and sorts them based on their performance. Competitors can hide their result, despite that NEUROtechnology algorithm is now ranked the second best in both EER (Equal Error Rate) and FMR (False Match Rate) [70]. The description given for MM_FV 5.5 is that it uses minutiae, ridge count, and local ridge frequency for matching. It is prepared to align displacement, rotation, and non-linear distortion [70]. The verification process (by its definition) needs two images to get a comparison score. In this case the inspected image and image without any damage was used to get this score. For example, image number 3 with pressure and moisture damage was compared to image number 3 without any damage. The same rules were applied to get the results from image without damage – these images were just verified against themselves. Quality estimation is done as an optional part of the VeriFinger comparison process and it is not described anywhere.

The results of the quality estimation are in the range of 0 to 100. The comparison score has a much higher range; from 0 to 2,250. To be specific, the maximum is not defined precisely – it is likely that some ideally created image could have an even higher score. VeriFinger version 10.0 from the MegaMatcher package directly from NUEROtechnology was used.

5.5.2.2 NIST Fingerprint Image Quality

Quality by NFIQ is a “predictor of a matcher's performance” [71]. This decision has a great impact on quality estimation. Because in its core it is not focused on quality, but on a matcher performance. That leads to some flaws, causing inaccuracies. However, fingerprint quality is estimated by measuring the feature vector and projecting this vector to quality classes via a neural network. The feature vector has 11 dimensions (foreground, total number of minutiae, minutiae that have quality over 0.5, 0.6, 0.75, 0.8, 0.9, and quality zone with the quality 1, 2, 3, 4) [72]. Two essential pieces of information are needed – quality map and minutiae. The minutia point detection algorithm is rather benevolent, so there are a lot of false detections. The quality map is a combination of the direction map (direction of ridges), the low contrast map (focuses on the part of the image with low contrast, which is useful for the differentiation of the fingerprint and the background), a low flow map (parts of image where direction map must be estimated from surroundings), and a high curve map (parts with a high curvature of ridges). This quality map is assigning values from 0 to 4 to a fingerprint image, where 0 is the background and the higher number is a higher quality part of image. [68] [71] [72]

It uses five classes for quality. This was set as a middle value between 3 and 10, which is (based on NIST research) the useable number of classes for fingerprint matchers [72]. NFIQ is used as a standard and has its implementation available online. It is not surprising that the NFIQ algorithm is an

optional part of VeriFinger's software. This same logic can be applied to Oravec's quality estimate. Surprisingly enough, values from each implementation are not the same, which is strange because they should be based on the same standardized algorithm. Because of that, both results (from VeriFinger and Oravec) are used in the evaluation process. As it turns out, statistical results (represented here by graphs) are for both algorithms the same with only one exception. In this case both graphs are shown, otherwise only one representing both cases is displayed.

5.5.2.3 Oravec's Quality Measurement

This method is based on several factors which should contribute to the quality of a fingerprint. The emphasis is on the clarity of ridges, their continuity, consistency of the fingerprint, and the size of it. The source image is divided into blocks. When inspecting a block a wider area is considered. As a pre-processing step, the orientation of the ridges is determined. After that, blocks are rotated so that ridges are aligned vertically. This is based on the sum of the discrete Fourier transformation. Each of the blocks that are not in the background is measured for six values. Orientation precision (the ratio between the chosen rotation of blocks and other rotations), continuity of ridge structure within a block, the continuity of orientation in scope of surrounding blocks, linear regression (used for determining overlap of the colours of valleys and ridges), ridge and valley ratio (using some information from linear regression method), and range of contrast. Each of these values can detect different damage to the fingerprint. Weight averaging of these values then determines block quality. The generalization of these blocks is a fingerprint quality. [68]

The quality score is in the form of a percentage, so the range is from 0 to 100. The original implementation of this method by Mr. Oravec is used.

5.5.3 Evaluation of Generated Narrow Images

First, the evaluation of narrow images is presented. That is because narrow images for swipe sensors are more common. Evaluation is structured to all basic options (as described in Subchapter 5.4.) and then to extreme damages (the most damaging combinations of basic options). For each measurement of evaluation there are graphs that show the minimal value for all fingerprint images, the maximal value (red dots), and the median (shown as black dash). It is important to note that damage which has the worst score (closer to minimal quality) has done a higher damage to an image, thus is treated as the best damage. All graphs in the following subchapters have the same value range. Sometimes the graphs can be unnecessarily large, but on the other hand all graphs are easily comparable.

5.5.3.1 Pressure and Moisture Damage Evaluation

Example images and more information about damages can be found in Subchapter 5.4.1. The Oravec quality score is the first metric, which is shown in Figure 5.12. As can be seen, the worst result (hence most damage done to images) is achieved by *pm2* (high to normal pressure). That is interesting because damages *pm6* (slowly high to low) and *pm1* (extreme) have a similar nature to *pm2*. Apparently, high pressure in some cases can be worse than the low-contrast parts of the fingerprint.

NFIQ scores (shown in Figure 5.13) determine the best damage $pm0$ (all low). It is best only because its maximal score (the NFIQ score is the only one where a higher score means lower quality) is at 4. It is worth noting that there are damages which have in some parts lower pressure than $pm0$ and their quality is not that low.

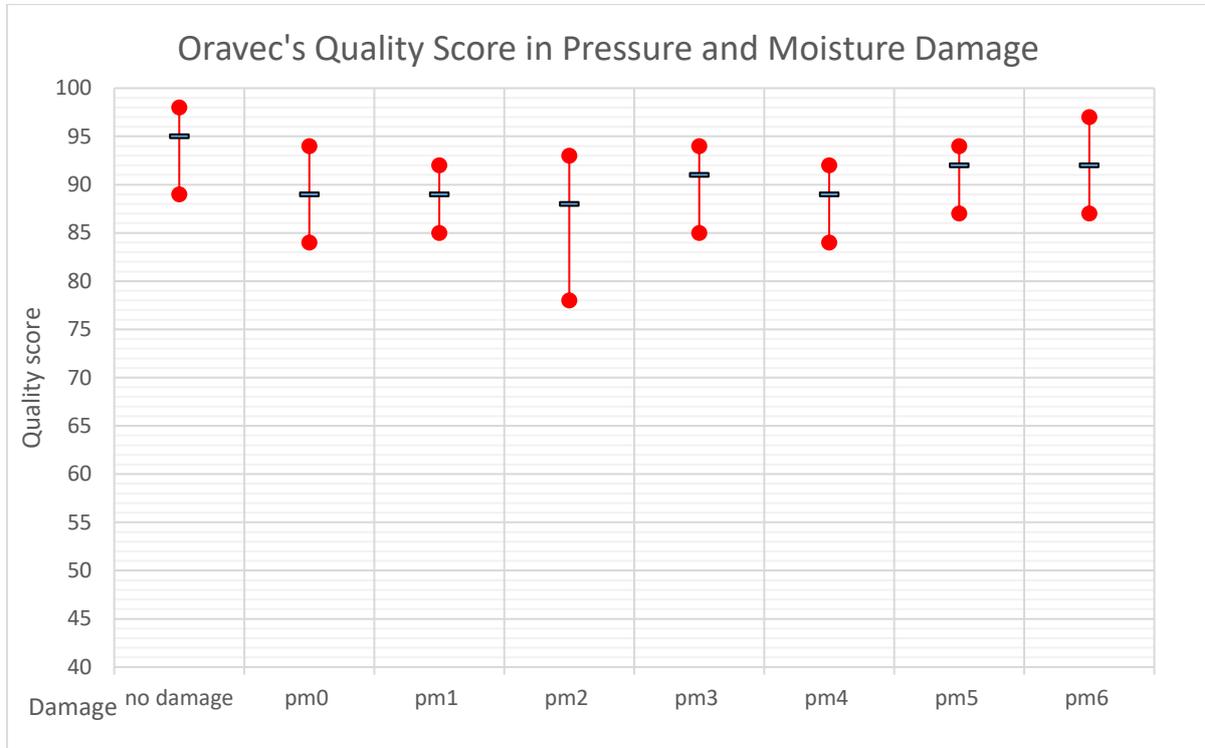


Figure 5.12: Graph of Oravec's quality score in pressure and moisture damage (narrow).

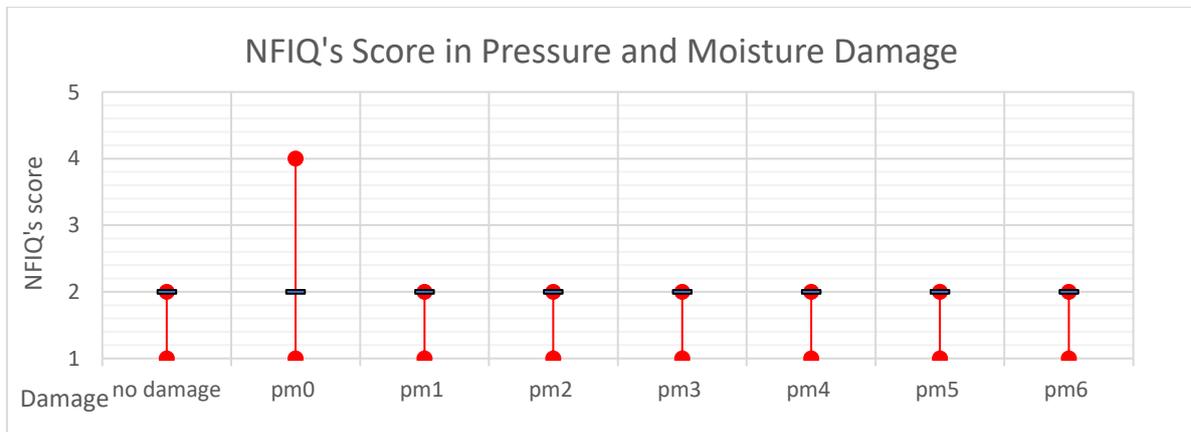


Figure 5.13: Graph of NFIQ's score in pressure and moisture damage (narrow).

VeriFinger's quality score (Figure 5.14) has a wide range of values. The absolute minimal score is achieved by $pm1$ (extreme), but in the median values the best is $pm2$ (high to normal). $Pm1$ as an extreme damage was expected to have low quality. It is the median result of the $pm2$, which hints that this damage would probably be one of the best altogether from the pressure and moisture group. Worth noting is also $pm0$, (all low) which has a better quality than $no\ damage$ image.

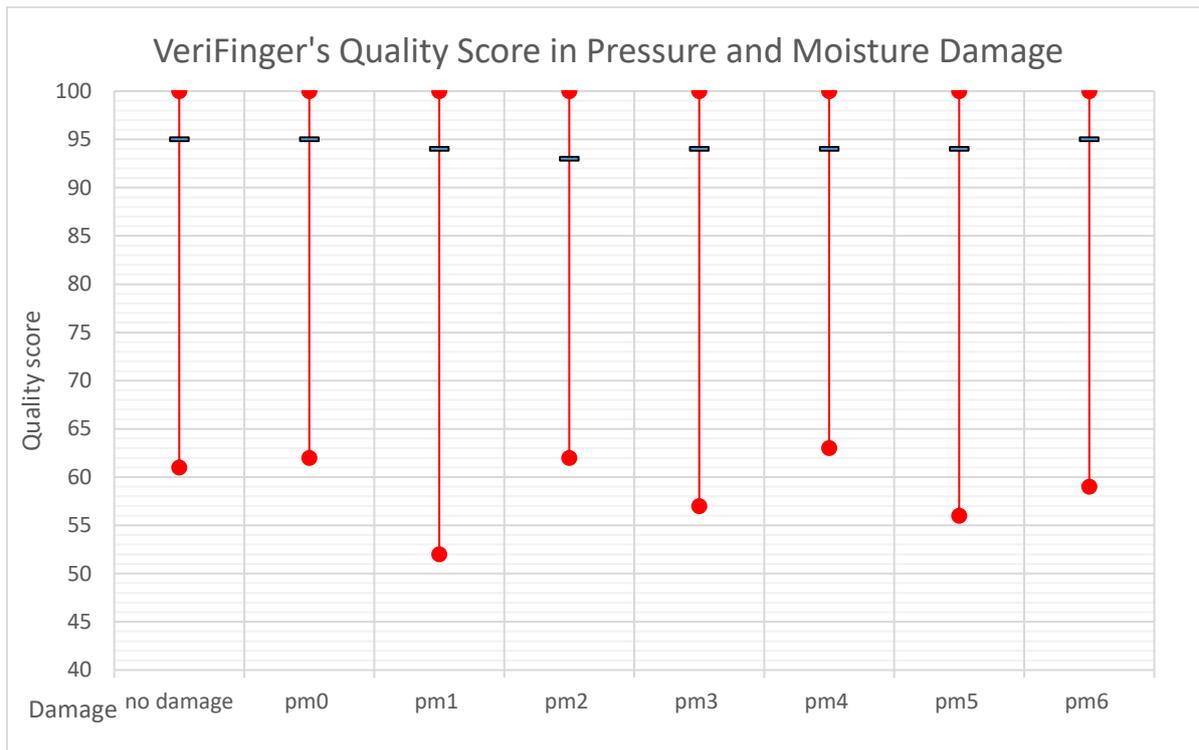


Figure 5.14: Graph of VeriFinger's quality score in pressure and moisture damage (narrow).

The last graph (Figure 5.15) shows results in the comparison score. An enormous gap between *no damage* and damaged images can be seen. The lowest values for the median are practically the same for *pm0* (all low), *pm1* (extreme), and *pm3* (low to high to low) – the exact numbers being 1005.5, 1009, and 1007.5, respectively. By the minimal score values it can be declared that *pm1* is the best in this

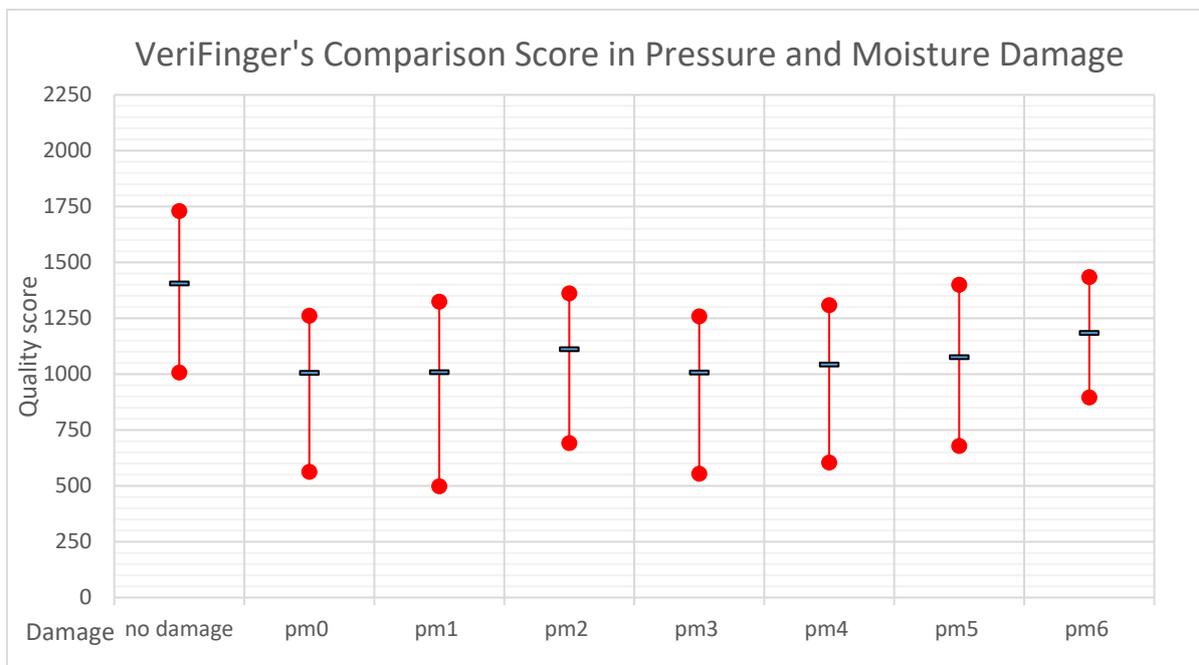


Figure 5.15: Graph of VeriFinger's comparison score in pressure and moisture damage (narrow).

metric. Both damages with high pressure (*pm2* – high to low and *pm6* – slowly high to low) have shown bad results in the comparison score.

Generally, the best results are achieved by *pm0* (all low), *pm1* (extreme), and *pm2* (high to low). For most of the metrics, there is a substantial difference between *no damage* and damaged images, which is another proof of verification.

5.5.3.2 Distortion and Damaged Sensor Damage Evaluation

Example images and more information about sensor damage can be found in Subchapter 5.4.3 and information about distortion is in Subchapter 5.4.4. Similar to previous subchapters, the focus is on the best damage in each metric.

The Oravec metric (Figure 5.16) clearly shows that the biggest damage is done by *dis0* (extreme). *Dmg0* (long) and *dmg1* (short) show very similar lines as the *no damage*. The second in damage done is *dis2* (move Y), which shows minimal values the same as the *dis0*, but the median is quite higher.

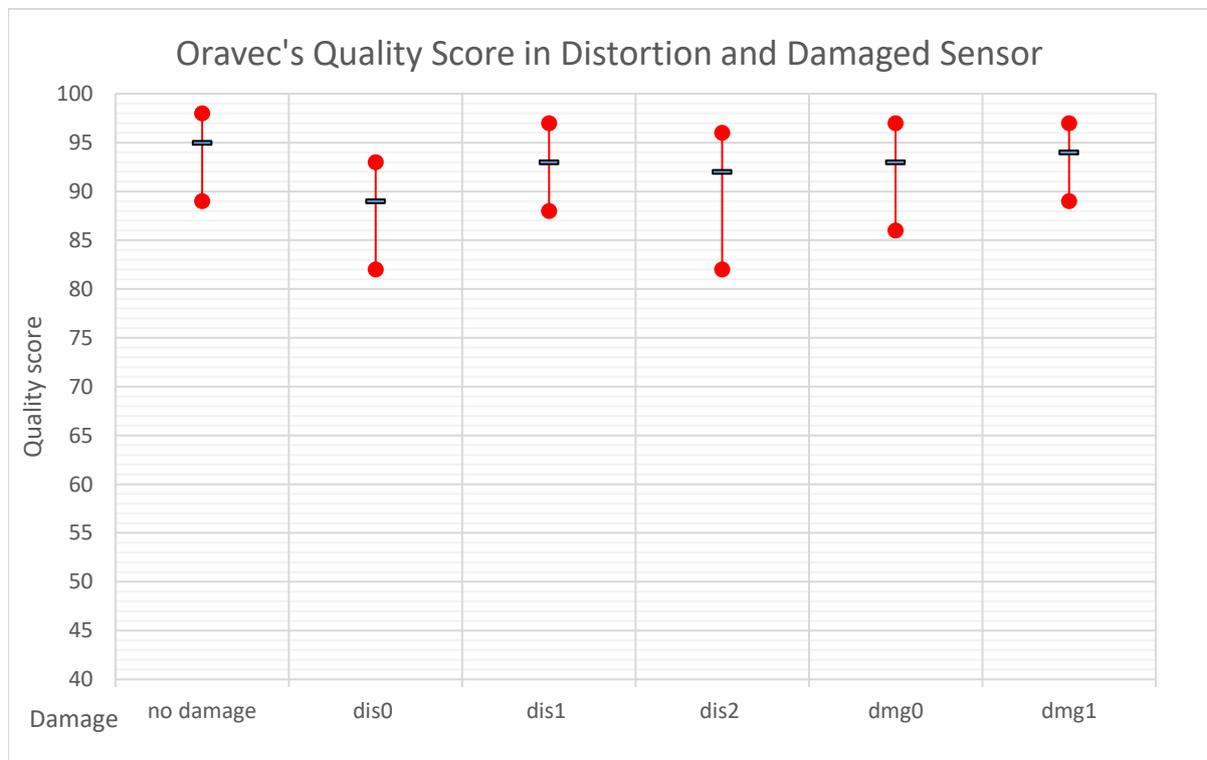


Figure 5.16: Graph of Oravec's quality score in distortion and damaged sensor (narrow).

The same results can be seen by NFIQ (see in Figure 5.17). The only damages which have a higher maximal damage are *dis0* and *dis2*. The increase is one class lower than in the pressure and moisture damage.

Figure 5.18 shows an interesting behaviour. Once again, the ranges are quite high. *Dis0* (extreme) shows the lowest median score. As far as the absolutely minimal score is concerned, the lowest values have *dmg0* (long) and *no damage* with *dmg1* (short). It can be concluded that *dis0* must constantly have low scores to achieve that median value.

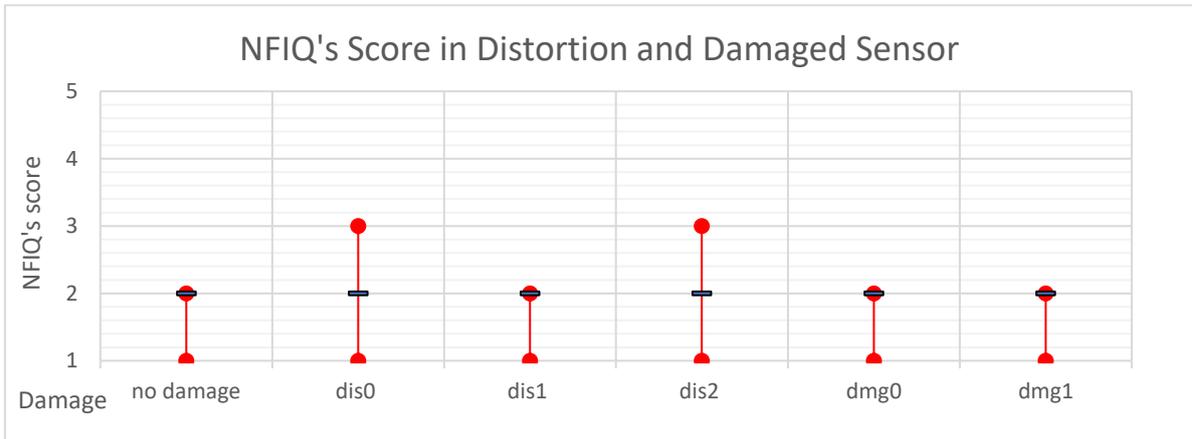


Figure 5.17: Graph of NFIQ's score in distortion and damaged sensor (narrow).

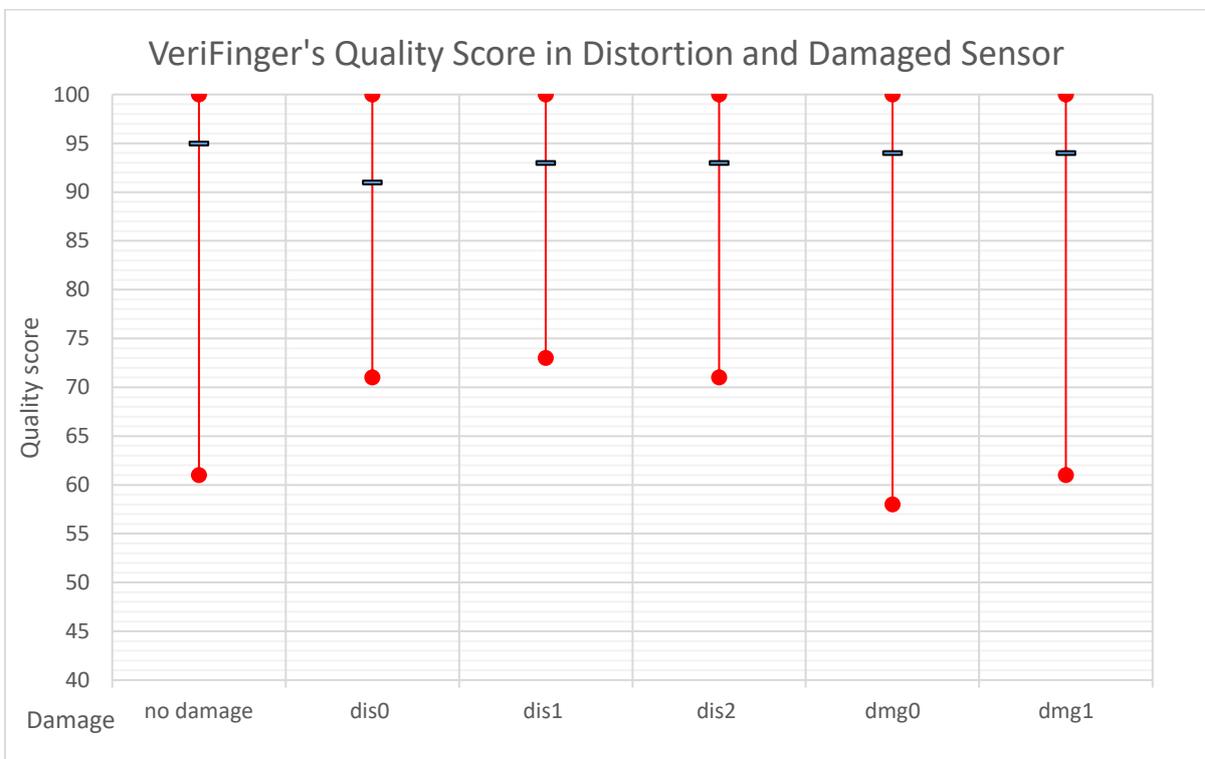


Figure 5.18: Graph of VeriFinger's quality score in distortion and damaged sensor (narrow).

In Figure 5.19 VeriFinger's comparison score (with the exception of *no damage*) sorted damages from the best to the worst (from left to right). *Dis0* (extreme) shows all basic damages with the worst results.

The clear winner of the best damage in this category is the *dis0* (extreme). Since this distortion was prepared as extreme, this result is not a big surprise. On the other hand, pressure damage and the edge parts of distortion look similar. This suggests that it is the rotation of minutiae points, which made this damage so much worse. From the damaged sensor category, the better one would probably be *dmg0* (long), but only narrowly. To sum up, all damages have lower results than the images without damage. It is close in some metrics, but the difference is there.

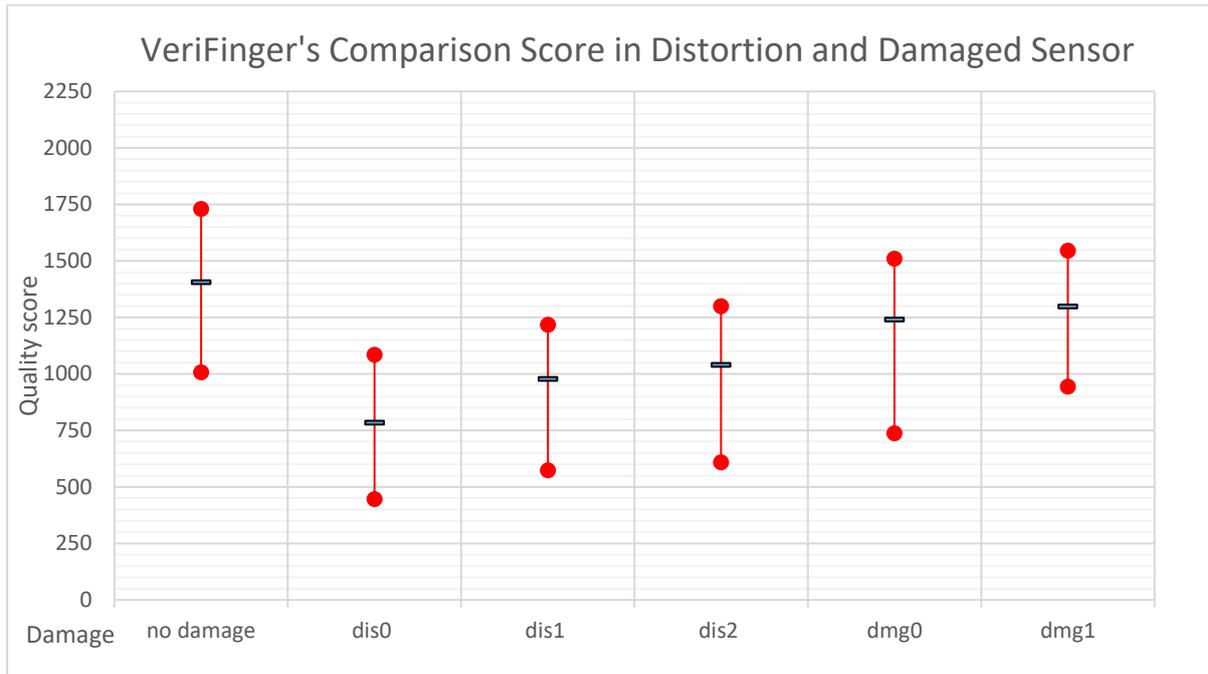


Figure 5.19: Graph of VeriFinger's comparison score in distortion and damaged sensor (narrow).

5.5.3.3 Narrow Damage Evaluation

This kind of damage is very interesting to examine. Deleting part of the image means deleting some of the minutiae points, which should immediately result in a worse quality score. However, this score is obtained from a single image (with the exception of the comparison score), so the information about less minutiae points is not available for the quality measurement metrics. Also, algorithms do not have to use the area of the fingerprint as an important part. In that case, there is a possibility to get a better quality image than that of the original (if metrics like minutiae to area ratio are used). There are 11 basic damages in this category, which means that there are two graphs (part 1 and 2) instead of only one for each metric. Nevertheless, graphs are shown one after another and the results are discussed together. Images and more information about specific damages are in Subchapter 5.4.2.

There are two damages (in the Figure 5.20 and Figure 5.21) with the same lowest median values and one of them has a slightly better minimal (and maximal) score. They are *narr4* (side zigzags) as the slightly one and *narr1* (all sideways steady). It is worth noting that *narr2* (cutdown), *narr9* (tip top round), and *narr10* (tip top sharp) have the same values as the *no damage* one. Basically, the algorithm is treating these images as the same quality. It is true that damage done by these narrow cuts in the top or bottom of the fingerprint image are very dependent on the exact size and location of the original image (a bigger image will have a larger area cut by the damage), which should result in a higher quality reduction.

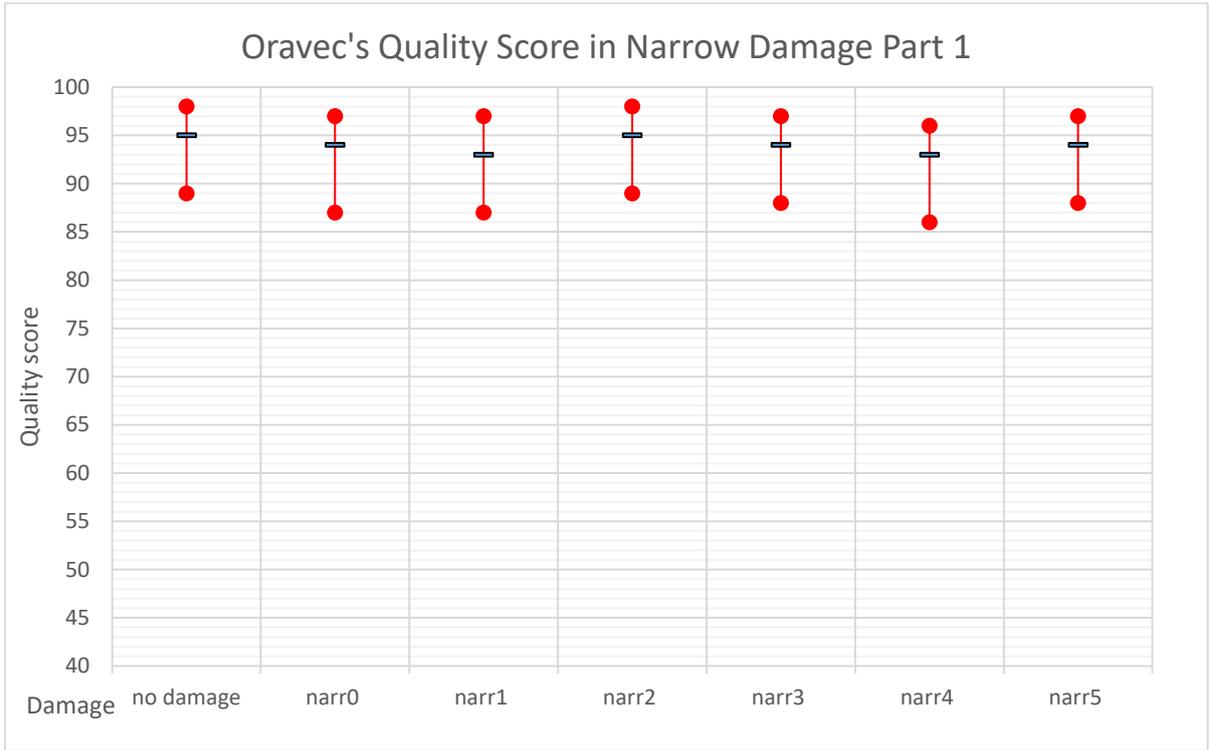


Figure 5.20: Graph of Oravec's quality score in narrow damage – part 1 (narrow).

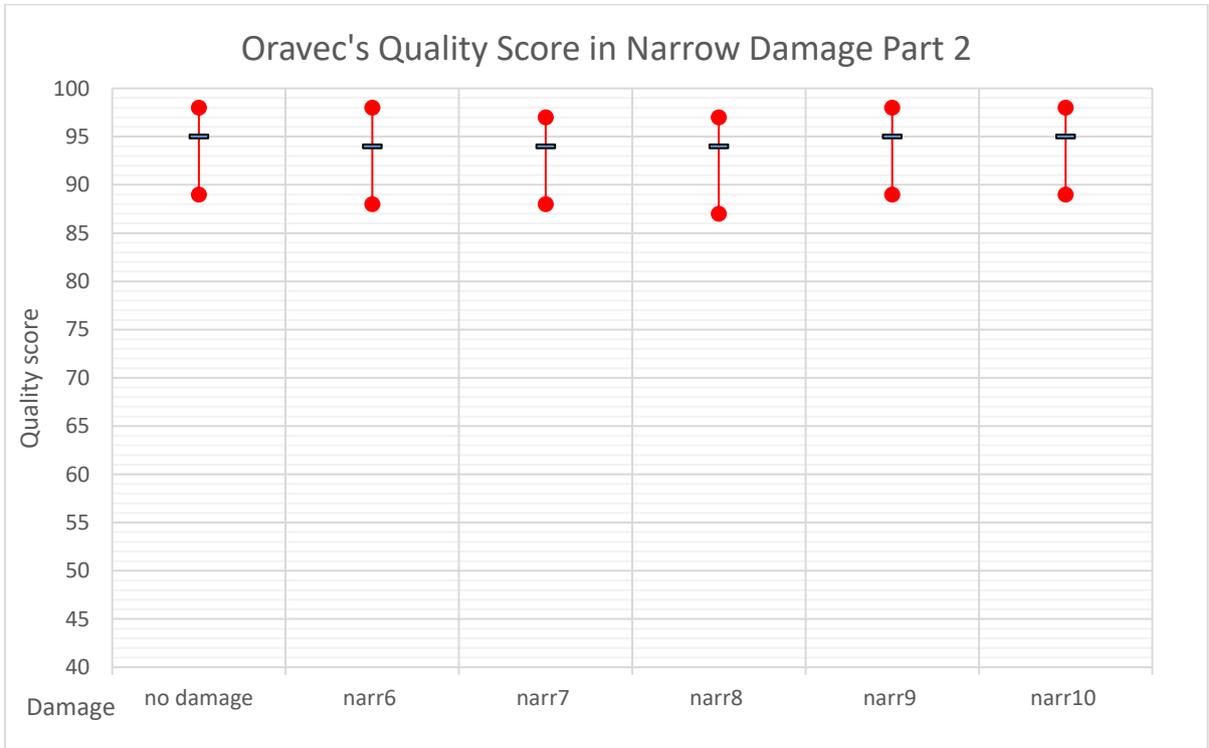


Figure 5.21: Graph of Oravec's quality score in narrow damage – part 2 (narrow).

NFIQ scores (Figure 5.22 and Figure 5.23) are not showing a lot of interesting results. Generally, it can be said that *narr1* (all sideways steady) and *narr5* (tip bottom jumpy) are a little bit better than other damages, but it is only by the maximal value and only in one class.

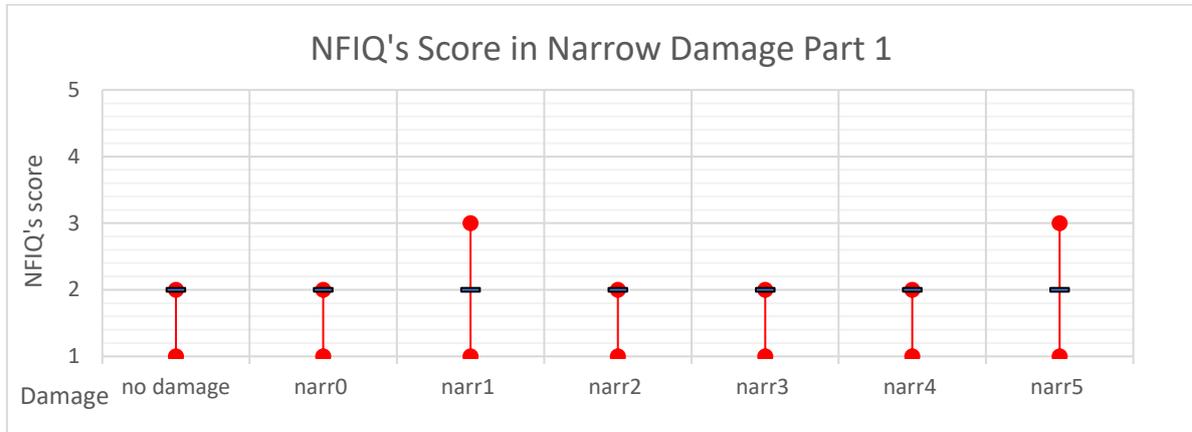


Figure 5.22: Graph of NFIQ's score in narrow damage – part 1 (narrow).

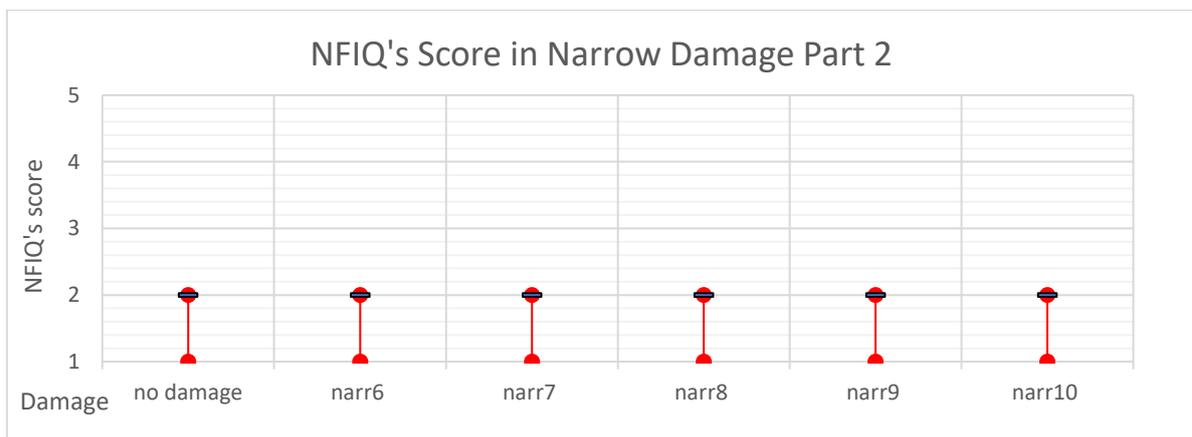


Figure 5.23: Graph of NFIQ's score in narrow damage – part 2 (narrow).

The best damage in the VeriFinger quality metric (seen in Figure 5.24 and Figure 5.25) is, without a doubt, *narr4* (side zigzags). On the other hand, *narr7* (tip bottom standard) exhibits better results than *no damage*, and *narr10* (tip top sharp) and *narr2* (cutdown) also shows the same results as the *no damage*.

The best damages by the comparison score (Figure 5.26 and Figure 5.27) are *narr4* (side zigzags), *narr7* (tip bottom standard), and *narr1* (all sideways steady) in this order. *Narr4* has the lowest median score (barely), *narr7* has the lowest minimal score, and *narr1* is very close to these values. There are not any damages that would be better than the *no damage*. This is probably because in the comparison score information about the missing minutiae points in the edges of the damaged images is available. Thus, this small area reduction is influencing the score.

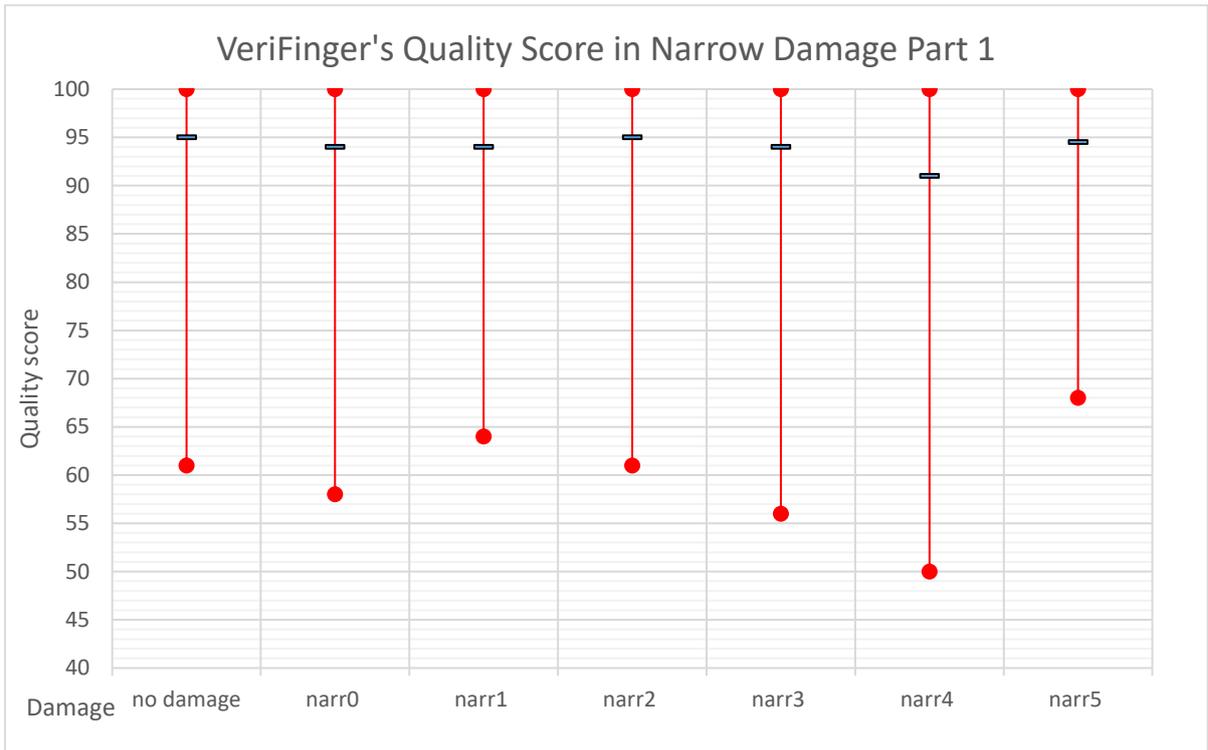


Figure 5.24: Graph of VeriFinger's quality score in narrow damage – part 1 (narrow).

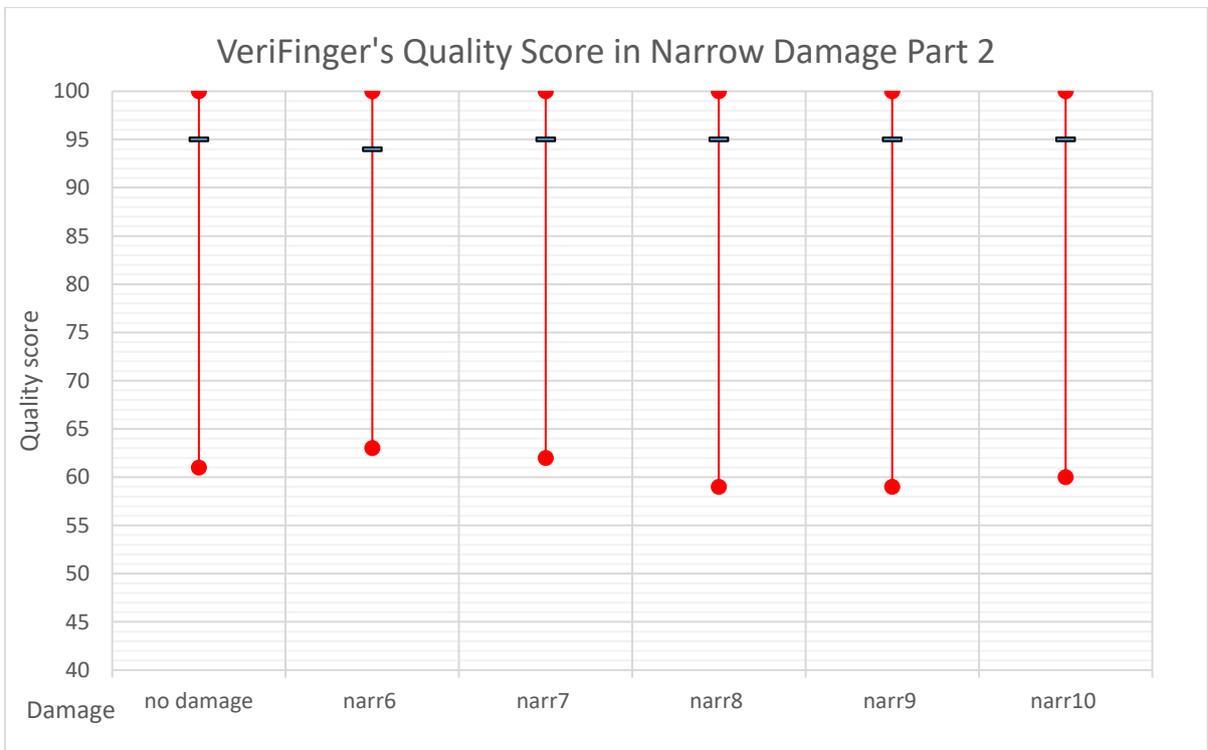


Figure 5.25: Graph of VeriFinger's quality score in narrow damage – part 2 (narrow).

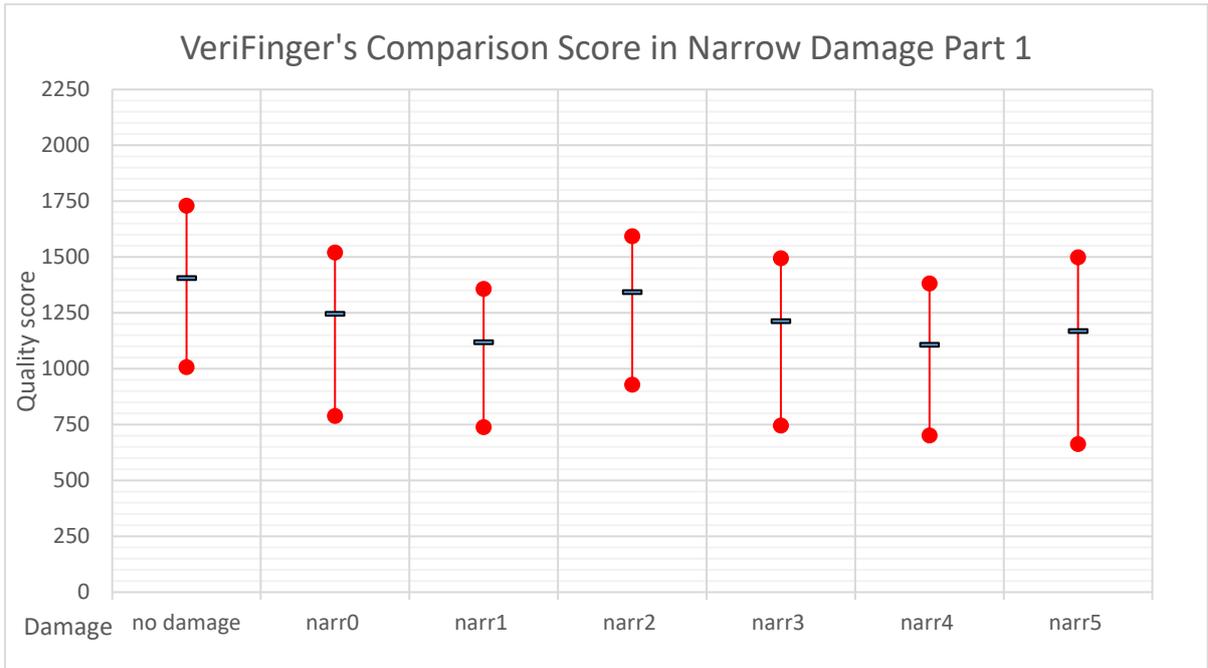


Figure 5.26: Graph of VeriFinger's comparison score in narrow damage – part 1 (narrow).

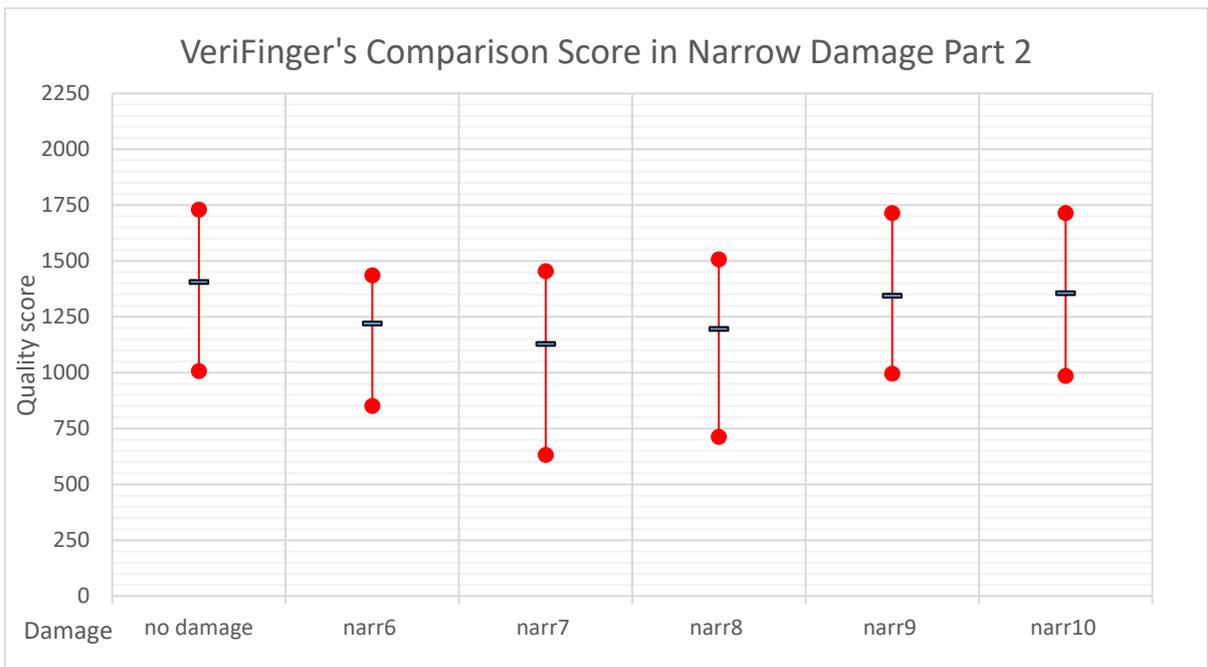


Figure 5.27: Graph of VeriFinger's comparison score in narrow damage – part 2 (narrow).

When examining all results, the best score in the narrow category is the *narr4*. In the second place would be probably *narr1*. *Narr4* is the only damage that could directly damage the fingerprint core. That is a crucial point not only because of its importance for fingerprint classification, but also because there is usually the highest density of minutiae points. *Narr1* presumably has the biggest area cut, but that is heavily dependent on the exact location of the fingerprint.

5.5.3.4 Extreme Damages

The database generated for the evaluation was composed of basic damages (evaluated in previous subchapters) and also from the combinations of these damages. Restricted combination settings were used (for more information see Subchapter 4.5). This resulted in 1,152 combinations (8, 4, 12, 3 – see Eq. 4.37). From Subchapter 5.5.1 it is known that there were 150 source images in the database. 150 images – each has 1,152 impressions, which gives 172,800 images in total. This subchapter picks the seven best damage combinations to evaluate.

An important factor is how to pick the best damages. In the end the following methodology was applied. For each quality metric (the Oravec quality score, the NFIQ, etc.) all damages were sorted by median value and minimal (or maximal) value (so the result would be the best damage). From these sorted damages, at least 10 results were taken. If there were the same results when all of them were taken, sometimes more than 10 results were chosen. These chosen lists of damages from all metrics were then joined together and the frequency and order on the original list were calculated. The most frequent damages were marked as the best damages. If the situation occurred where more damages have same frequency, then damage with a lower cumulative order number on the original list was taken. In this case, the chosen damages (sorted from the most damaging) were:

- **Pm2 dis0 narr4 dmg1:** This damage is composed from high to normal pressure, extreme distortion, side zigzag narrowing, and short sensor damage. Can be seen in Figure 5.28b.
- **Pm2 dis0 narr4 dmg0:** This damage is composed from high to normal pressure, extreme distortion, side zigzags narrowing, and long sensor damage. Can be seen in Figure 5.28c.
- **Pm2 dis0 narr1 dmg1:** This damage is composed from high to normal pressure, extreme distortion, all sideways steady narrowing, and short sensor damage. Can be seen in Figure 5.28d.
- **Pm1 dis0 narr4 dmg1:** This damage is composed from extreme pressure, extreme distortion, side zigzags narrowing, and short sensor damage. Can be seen in Figure 5.28e.
- **Pm2 dis0 narr5 dmg0:** This damage is composed from high to normal pressure, extreme distortion, tip bottom jumpy narrowing, and long sensor damage. Can be seen in Figure 5.28f.

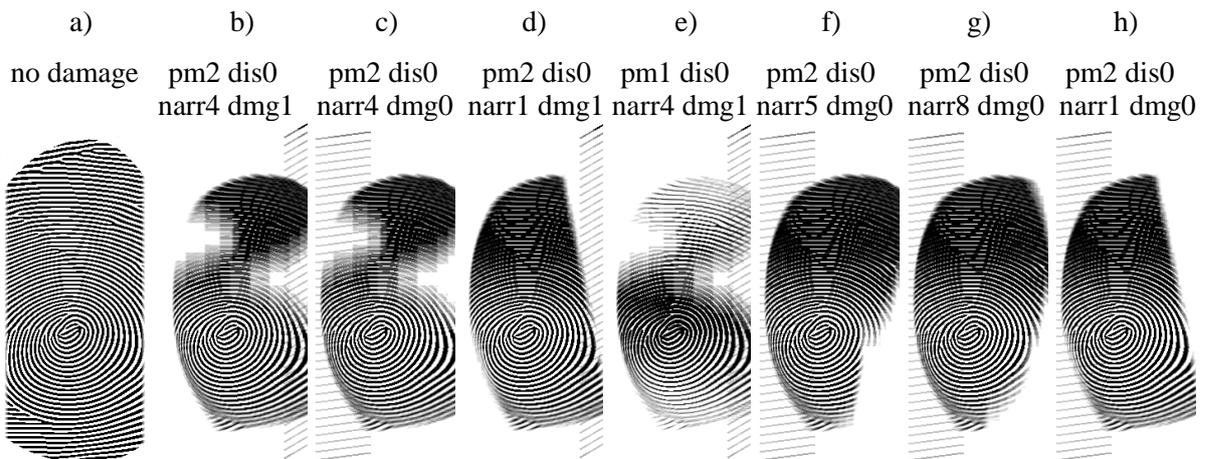


Figure 5.28: Examples of extreme damages in narrow width images.

- **Pm2 dis0 narr8 dmg0:** This damage is composed from high to normal pressure, extreme distortion, tip both sides narrowing, and long sensor damage. Can be seen in Figure 5.28g.
- **Pm2 dis0 narr1 dmg0:** This damage is composed from high to normal pressure, extreme distortion, all sideways steady narrowing, and long sensor damage. Can be seen in Figure 5.28h.

Differences between the scores of basic damages and these extreme ones are significant. On the other hand, the difference between extreme damages in Oravec's quality (can be seen in Figure 5.29) is small. The only exceptions are "pm2 dis0 narr4 dmg0", whose median value is lower and "pm1 dis0 narr4 dmg1", whose values are higher.

Results in the NFIQ score (Figure 5.30) are different as well. However, for the sake of damage comparison it is not much. The only score that stands out are the minimal values for "pm2 dis0 narr4 dmg1" and "pm2 dis0 narr4 dmg0".

The first metric where it would be possible to sort damages is the VeriFinger quality score (Figure 5.31). The first damage is "pm2 dis0 narr4 dmg1", the second "pm2 dis0 narr1 dmg1", and the third is "pm1 dis0 narr5 dmg0".

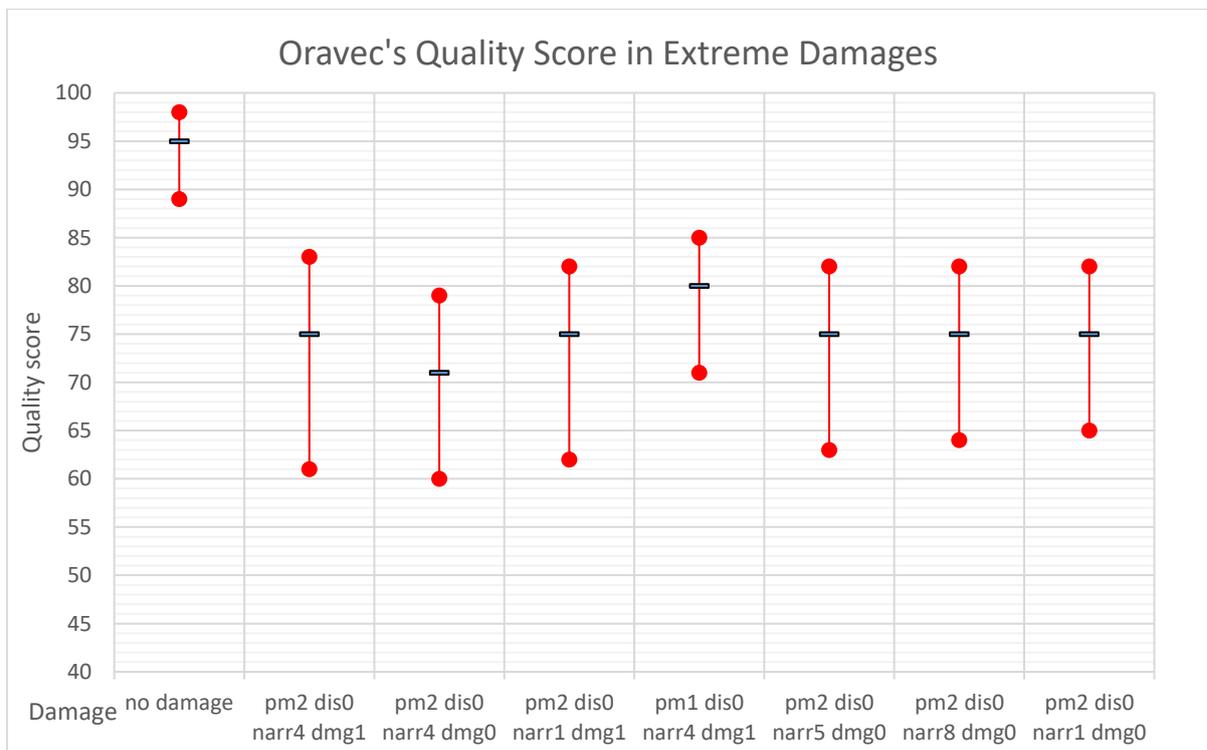


Figure 5.29: Graph of Oravec's quality score in extreme damages (narrow).

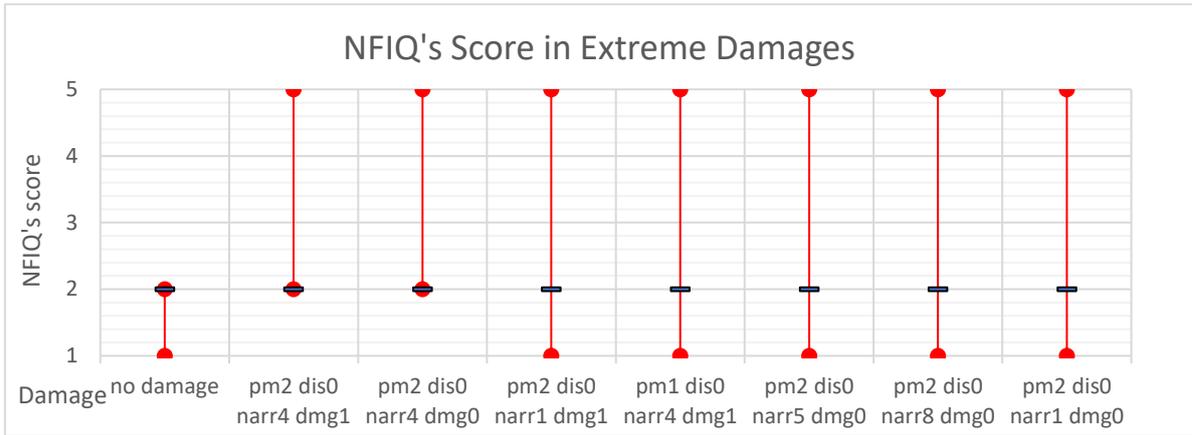


Figure 5.30: Graph of NFIQ's score in extreme damages (narrow).

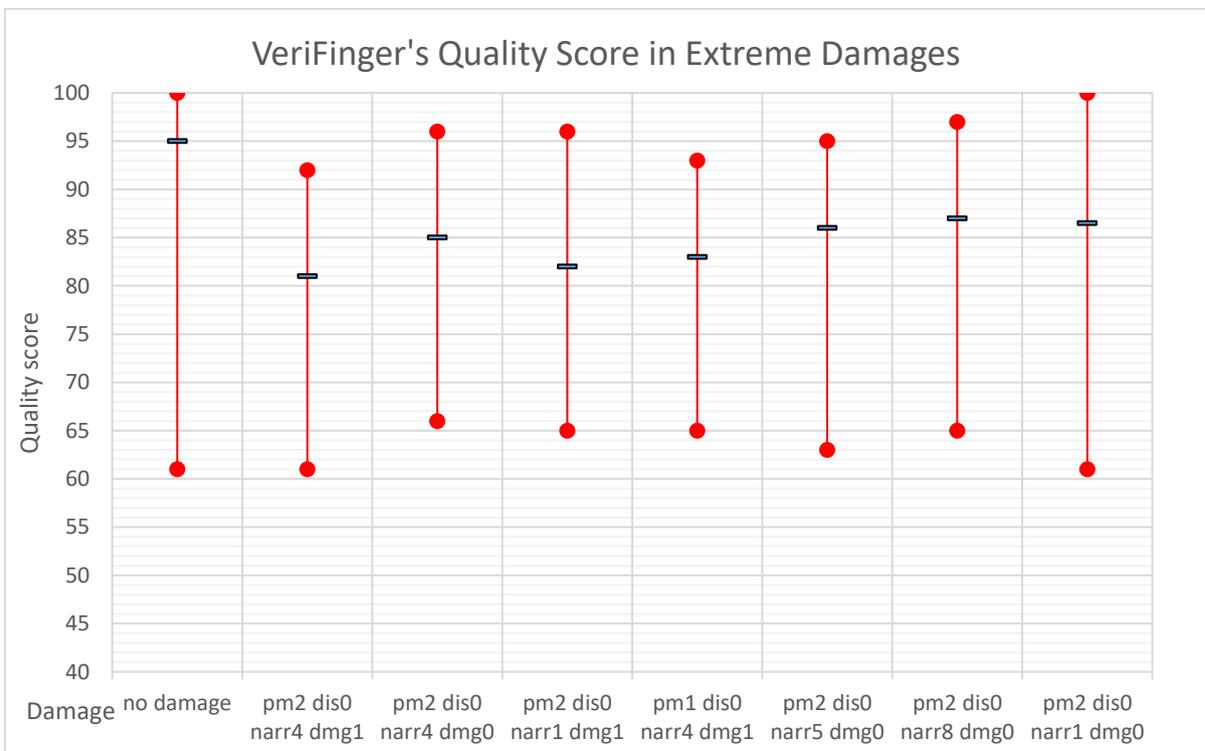


Figure 5.31: Graph of VeriFinger's quality score in extreme damages (narrow).

Same damages are at the top 3 in the comparison score (Figure 5.32) as well. Only order is different – it is “*pm2 dis0 narr4 dmg0*” followed by “*pm2 dis0 narr4 dmg1*” and “*pm1 dis0 narr4 dmg1*”. By combining results it can be said that the best damage is “*pm2 dis0 narr4 dmg1*” and “*pm2 dis0 narr4 dmg0*”. Based on the occurrences of damages in the chosen combinations it is certain that *dis0* is the most important (it appears in all combinations); the second in that regard is *pm2* (in all but one). On the contrary, *dmg0* and *dmg1* are doing some damage, but because they are evenly spread they are not so important. In the case of a narrow category, it could be said that *narr4* and *narr1* are better than others, but perhaps not so much. Basically, if the best of the individual damages are combined they create one of the most damaging combinations. There are, of course, different weights for each damage

category. Some damages which were not so good individually could excel in combinations, but there is no specific combination that would cooperate so well to make the result vastly better than looking at parts of that combination.

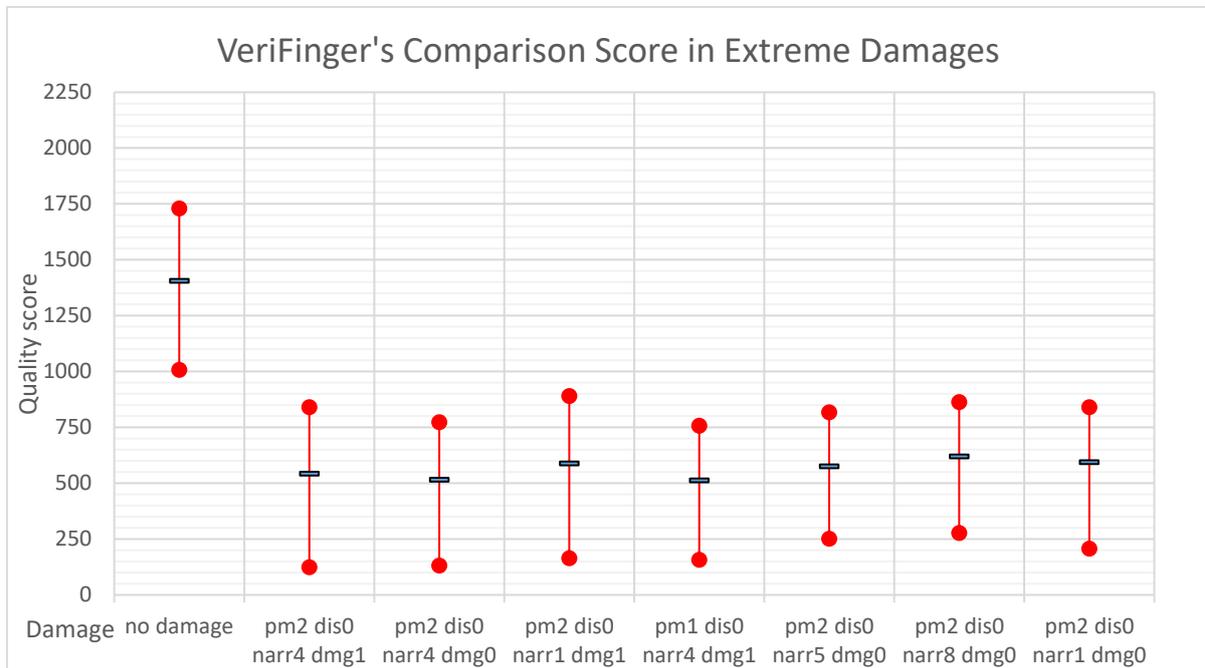


Figure 5.32: Graph of VeriFinger's comparison score in extreme damages (narrow).

5.5.4 Evaluation of Generated Normal Width Images

The evaluation of normal width images is following the same structure as the narrow one; first, the basic damages evaluation and then extreme ones made by combinations of the basics. Graphs for each quality metric with minimal, maximal, and median values are shown. The range of values for all graphs is the same as in the previous subchapters. The main reason for this evaluation is that the larger area of the fingerprint image could mean more minutiae points, and the quality measurement should also be more precise with more data. Furthermore, these results can be more easily comparable with other damages which are usually made to the normal width images.

5.5.4.1 Pressure and Moisture Damage Evaluation

Once again, Subchapter 5.4.1 holds all the important information and images of the evaluated damages. In the previous evaluation, the pressure and moisture category was the second most important factor, so it would be interesting if the same applies for the normal width images.

This metric (Figure 5.33) is almost identical to Figure 5.12. It follows that the conclusions are the same as well. The most damage done is from *pm2* (high to normal). Now the result is much clearer as the difference in the score is higher. Regardless of the visual similarities with other damages, this damage is objectively better.

Figure 5.34 and Figure 5.35 are non-unified NFIQ scores. It is unclear whether one of the implementations is from a different version or if there were some changes made. The difference is in *pm4* (normal to low) and *pm5* (recurrent normal to low). One graph shows maximal values at 3 and 2 respectively, while the other one shows a value at 4. It can be said that *pm0* (all low) and *pm6* (slowly high to low) are the worst damages here or, based on the first graph, that *pm1* (extreme), *pm2* (high to normal), and *pm3* (low to high to low) are the best ones. Examining of narrow damages shows that NFIQ is not good as a separating factor, so this is not a big issue.

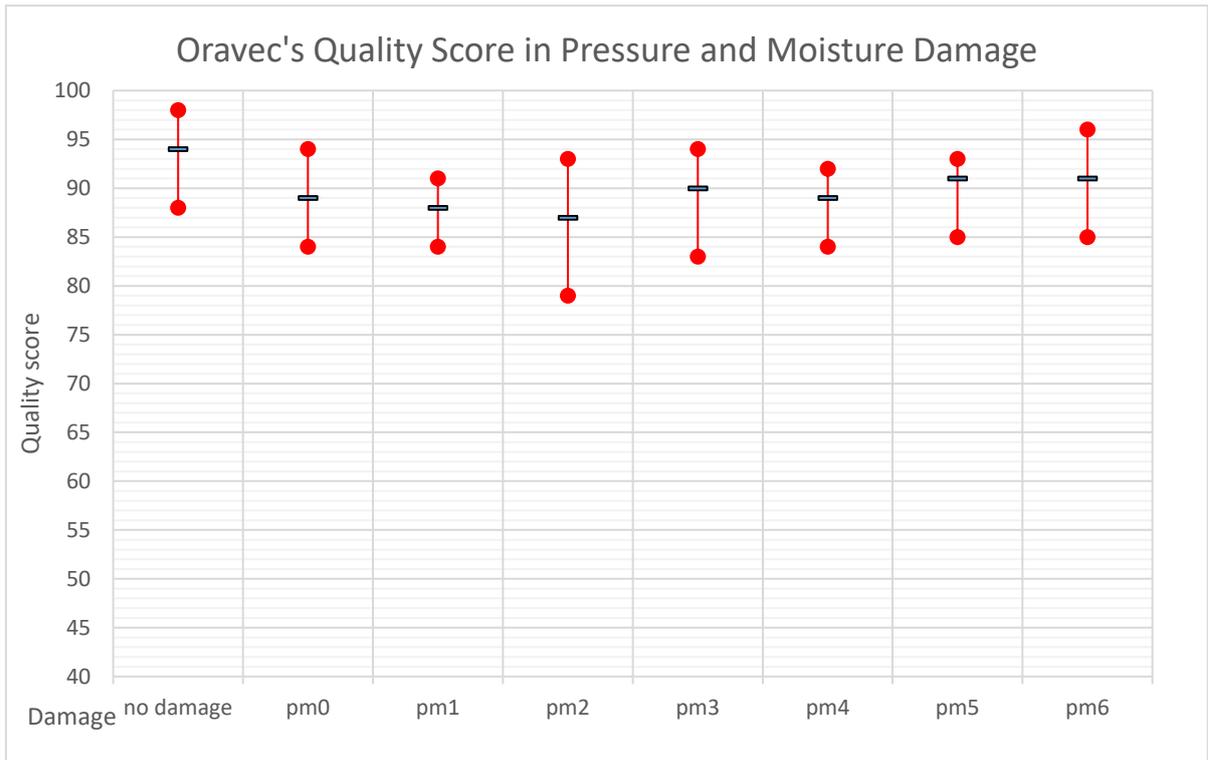


Figure 5.33: Graph of Oravec's quality score in pressure and moisture damage (normal).

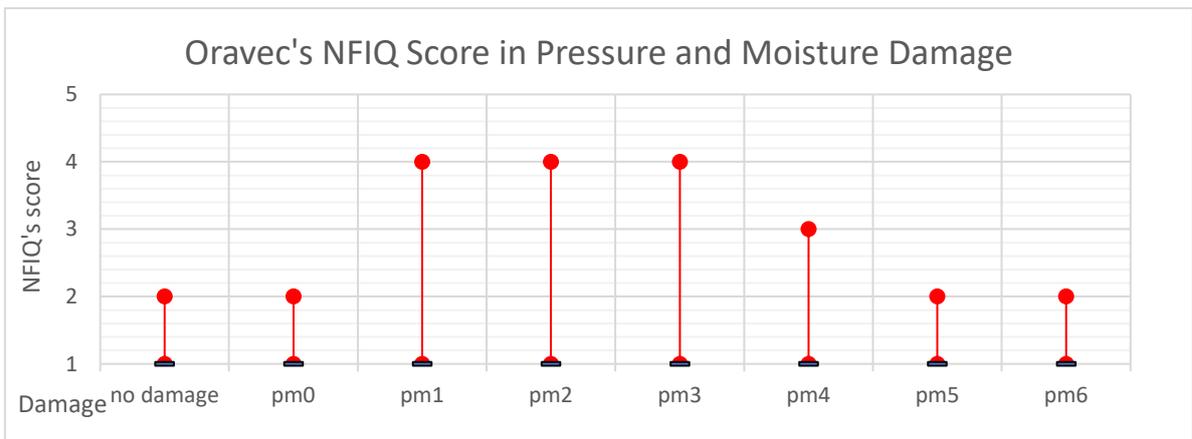


Figure 5.34: Graph of Oravec's NFIQ score in pressure and moisture damage (normal).

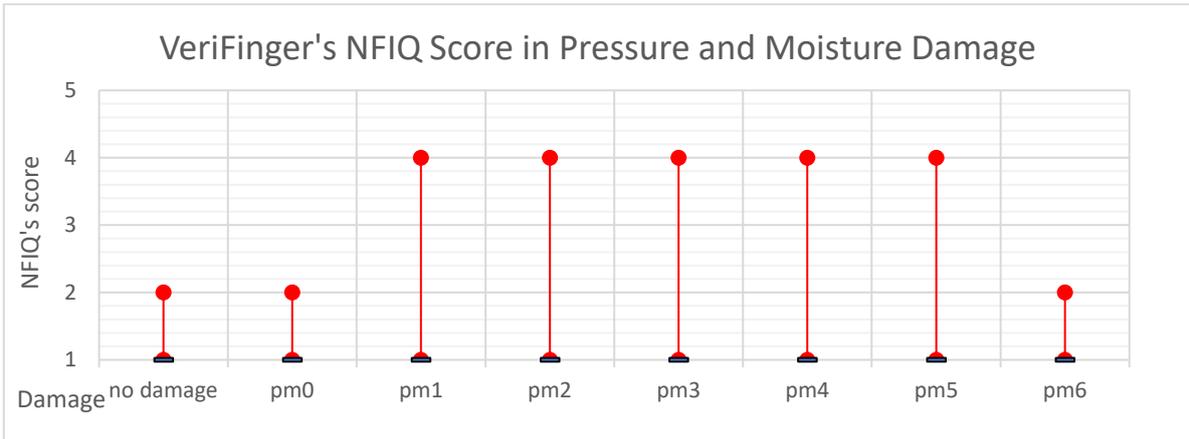


Figure 5.35: Graph of VeriFinger's NFIQ score in pressure and moisture damage (normal).

VeriFinger's quality score (as can be seen in Figure 5.36) shows very similar median scores for each damage. The lowest are *pm3* (low to high to low) and *pm6* (slowly high to low). What is worth noting is the big gap in the minimal value of *pm1* (extreme) and another interesting fact is that *pm5* (recurrent normal to low) has a better quality than *no damage*.

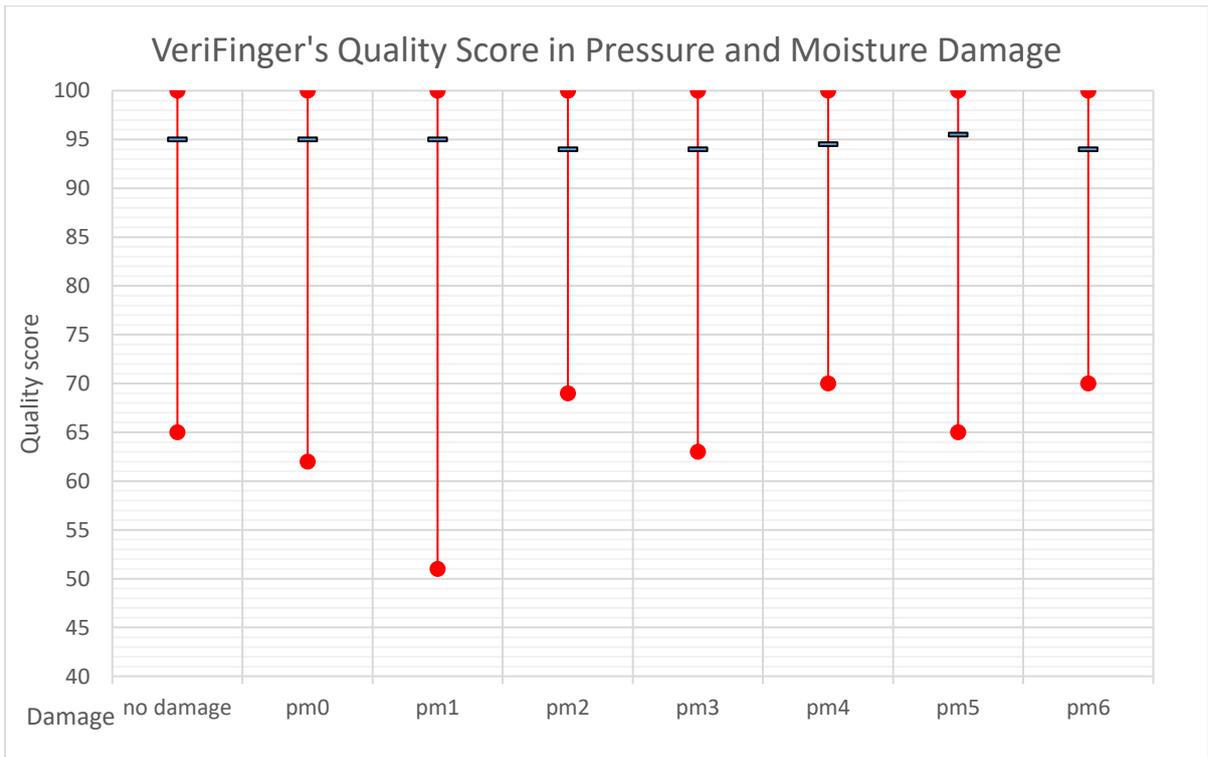


Figure 5.36: Graph of VeriFinger's quality score in pressure and moisture damage (normal).

In Figure 5.37 there are three damages that have a very similar median score. It is *pm1* (extreme) that has the lowest minimal value along with *pm3* (low to high to low) and *pm0* (all low). In the bigger picture, it is rather difficult to define the best damage for this category. The most was probably shown by *pm1* followed by *pm3*. *Pm2* (high to normal), which was dominant in narrow images, was not so successful in the normal width category. All damages have lower score than the *no damage* image.

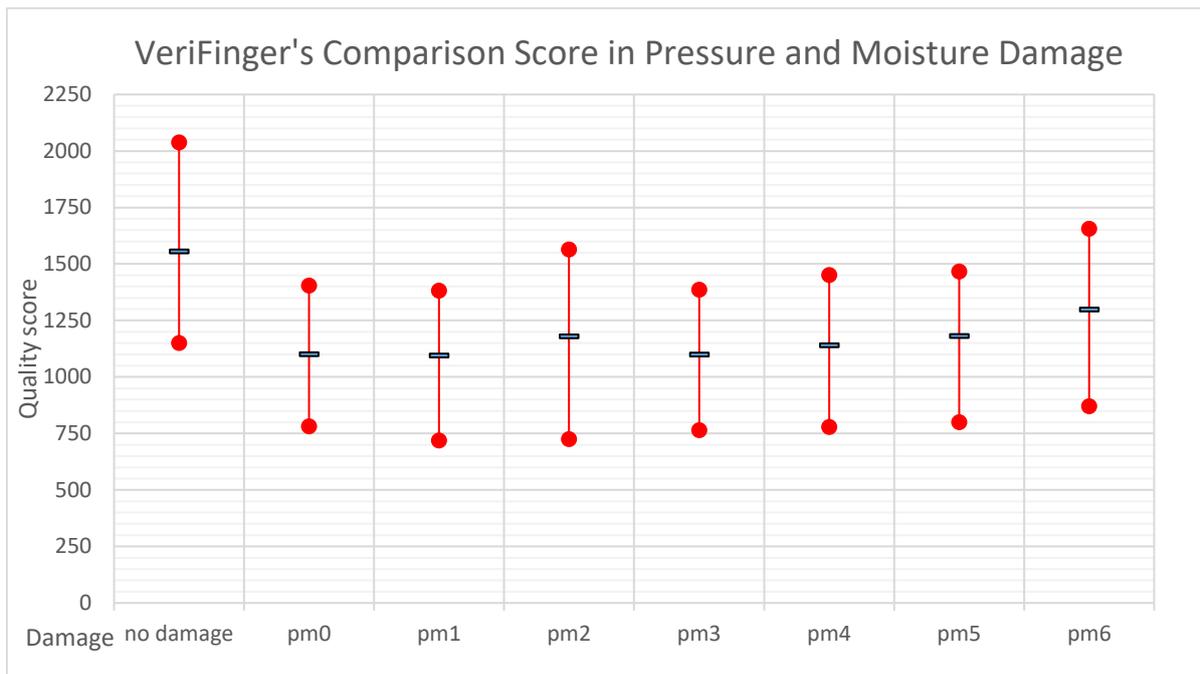


Figure 5.37: Graph of VeriFinger's comparison score in pressure and moisture damage (normal).

5.5.4.2 Distortion and Damaged Sensor Damage Evaluation

Subchapter 5.4.3 has information and images for the damaged sensor and Subchapter 5.4.4 holds the same information about the distortion. As usual, Oravec's quality score in Figure 5.38 follows to start the evaluation.

Total dominance over this metric is shown by *dis0* (extreme). What is unexpected and different is that there is a clear distinction between *dmg0* (long) and *dmg1* (short). The better one being the *dmg0*, as *dmg1* is exhibiting values very close to the *no damage*.

There were not a lot of interesting results in the NFIQ (Figure 5.39). It can be safely assumed that the median of all normal width images will be in the first class of the metric. *Dis0* (extreme) is only damage which achieved different (better) results.

Also, in VeriFinger's quality score (in Figure 5.40) *dis0* (extreme) has a median value much lower than other damages. What is interesting is that *dmg0* (long) has a higher median score than *no damage* (but lower minimal value). *Dmg0* and *dmg1* (short) essentially have the exact opposite evaluation than in the first metric.

Finally, there is VeriFinger's comparison score (Figure 5.41) where damages are sorted from the best in the left side (and *no damage* is an exception). *Dis0* (extreme) is located where extreme damages in the narrow images had been before. *Dmg0* (long) and *dmg1* (short) are very close together. In general, once again *dis0* is clearly the best damage of them all. Looking at the damaged sensor category the *dmg0* was getting a slightly better score all along, so that should be the better damage. With some small exceptions, all damages have a lower quality score than the reference (*no damage*).

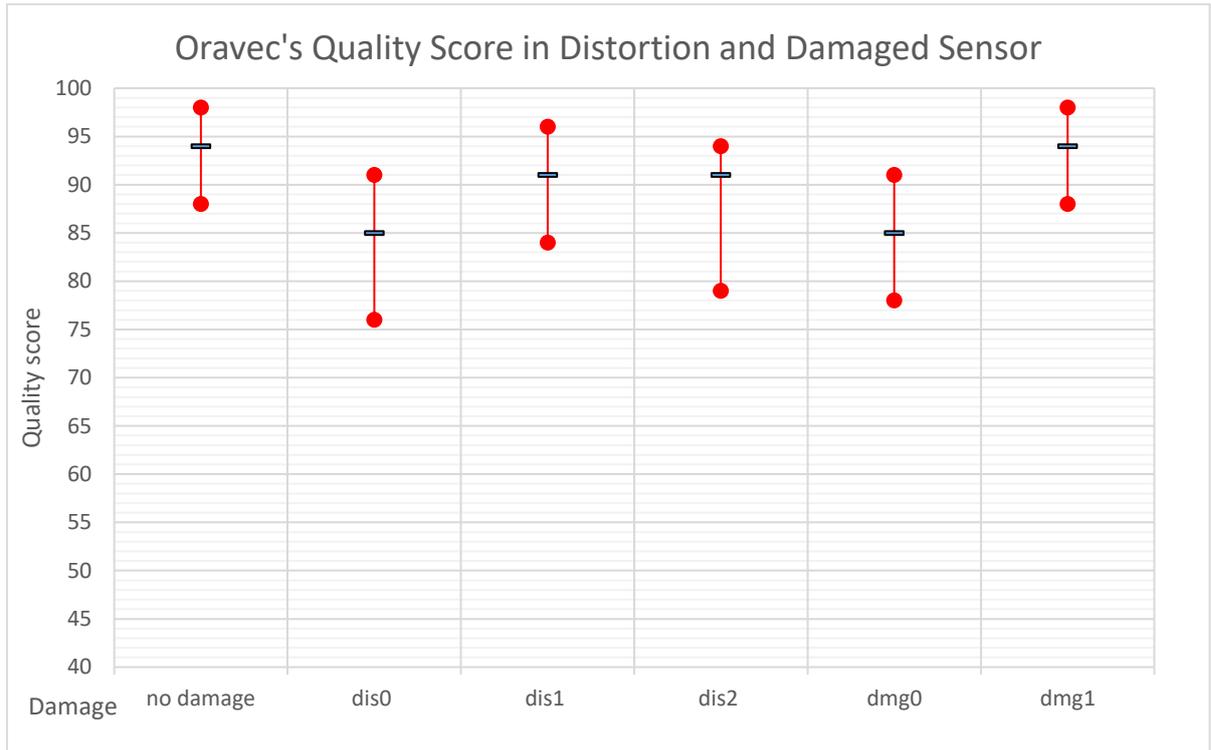


Figure 5.38: Graph of Oravec's quality score in distortion and damaged sensor (normal).

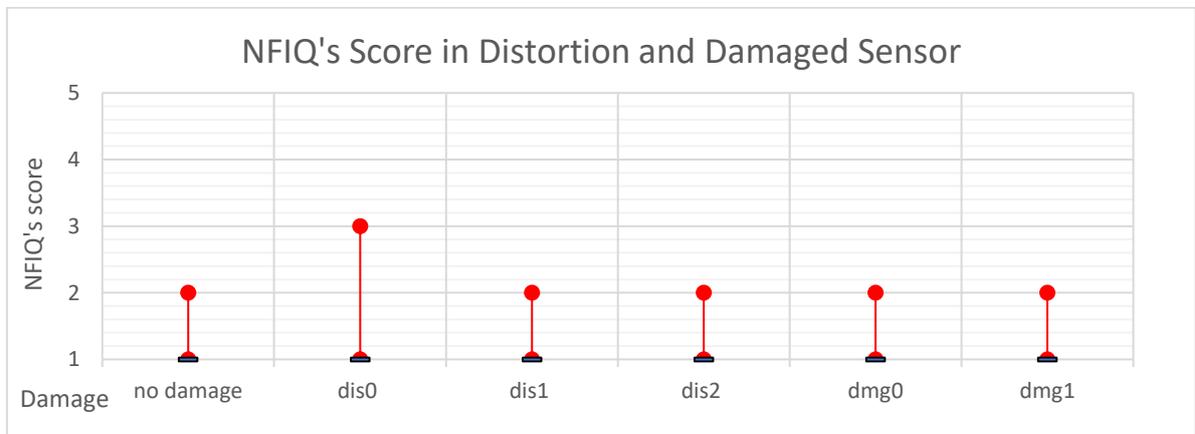


Figure 5.39: Graph of NFIQ's score in distortion and damaged sensor (normal).

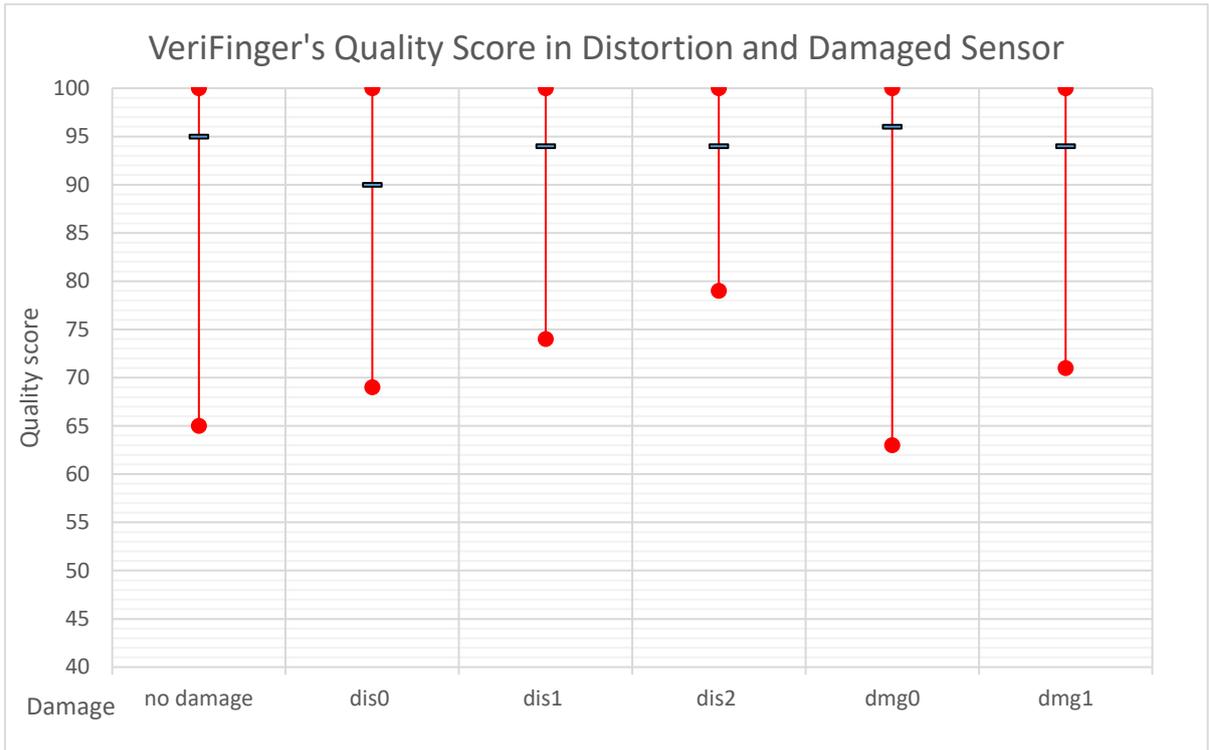


Figure 5.40: Graph of VeriFinger's quality score in distortion and damaged sensor (normal).

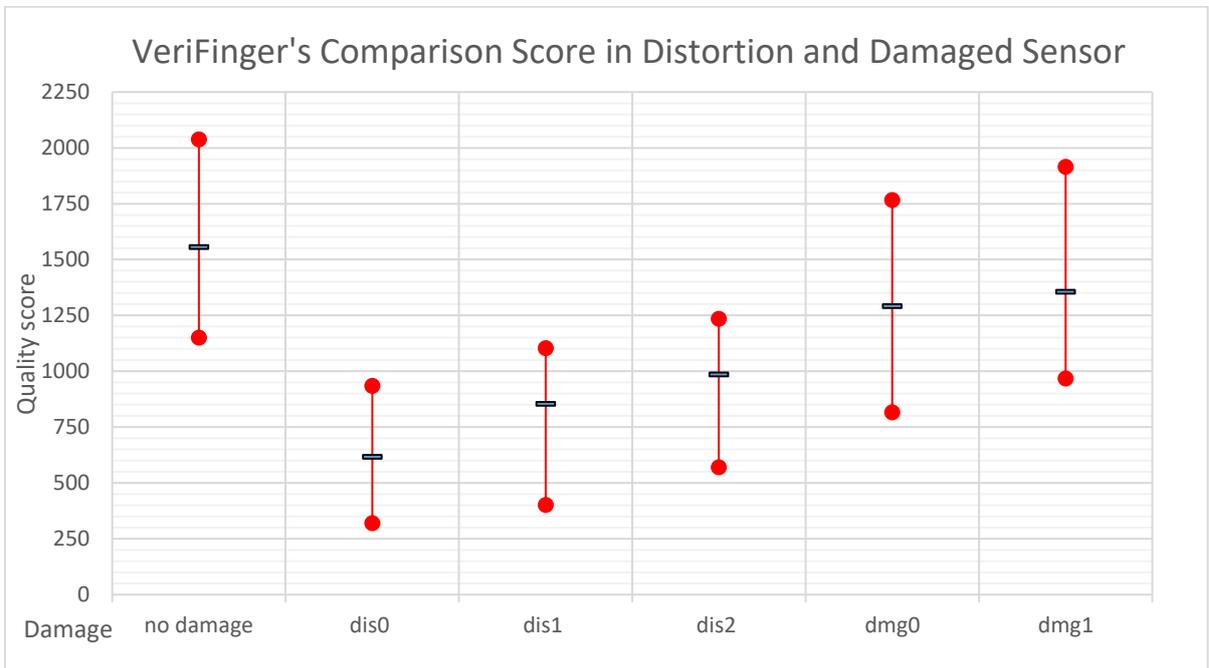


Figure 5.41: Graph of VeriFinger's comparison score in distortion and damaged sensor (normal).

5.5.4.3 Narrow Damage Evaluation

Because of the high number of damages in this category, there will be two graphs for each quality measurement. Both will be discussed at once. Example images and more information about narrow basic damages can be found in Subchapter 5.4.2.

This metric (in the Figure 5.42 and Figure 5.43) exhibits identical scores for each damage with two exceptions. That is *narr4* (side zigzags) and *narr2* (cutdown). The first of them is better than every other damage and the second has a slightly better score than the *no damage* – which is bad.

NFIQ quality scores (in Figure 5.44 and Figure 5.45) show identical values for everything. Thus contributing no useful information at all for the damage evaluation.

Median values of the next metric are shown in Figure 5.46 and Figure 5.47. They are also looking very similar to each other. *Narr4* (side zigzags) is essentially the only one better than the *no damage*. Then there are *narr3* (one side) and *narr5* (tip bottom jumpy) – which are worse than the *no damage*. Finally, there are several other damages which are basically at the same level as the reference.

In the comparison score (Figure 5.48 and Figure 5.49) there are all damages below the reference *no damage*. Significantly better are only the *narr4* (side zigzags) and *narr5* (tip bottom jumpy). The best damage in this category is without a doubt the *narr4*. But it is very sad that a lot of damages were declared as the same quality as *no damage*. With these results it can be expected that the *narr4* will later be a part of extreme damage combinations, but it would be more like voluntary damage.

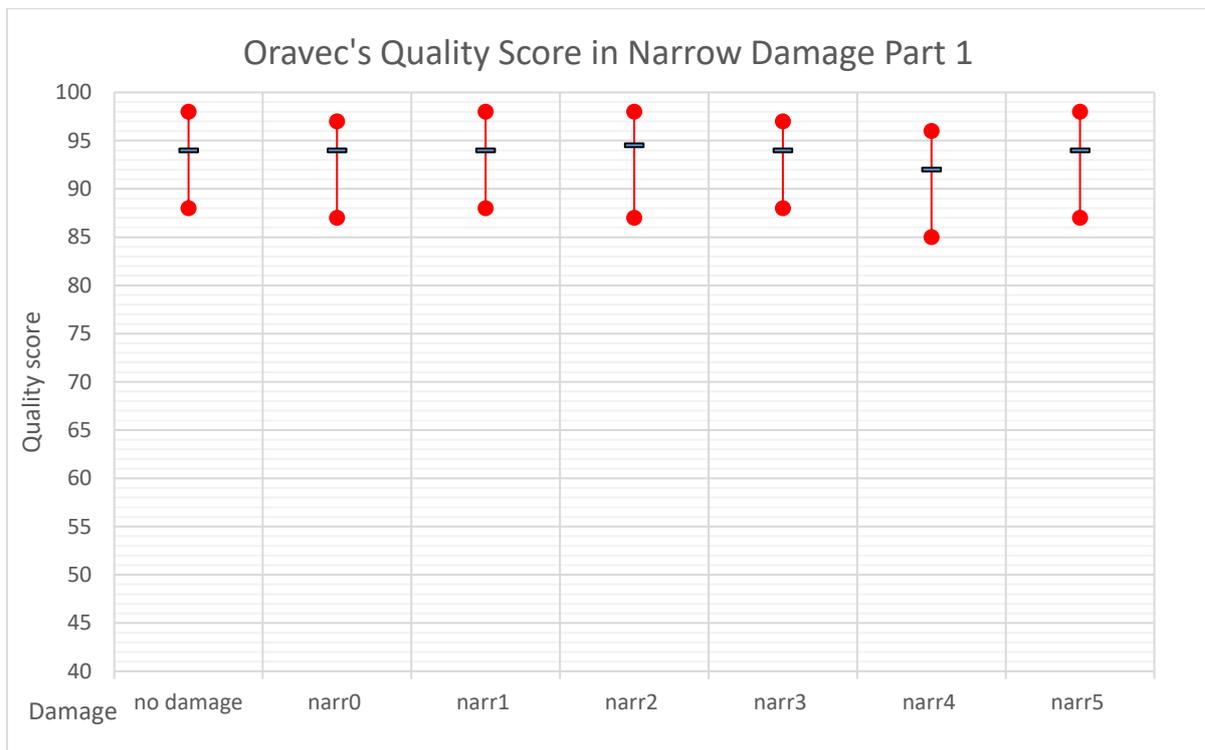


Figure 5.42: Graph of Oravec's quality score in narrow damage – part 1 (normal).

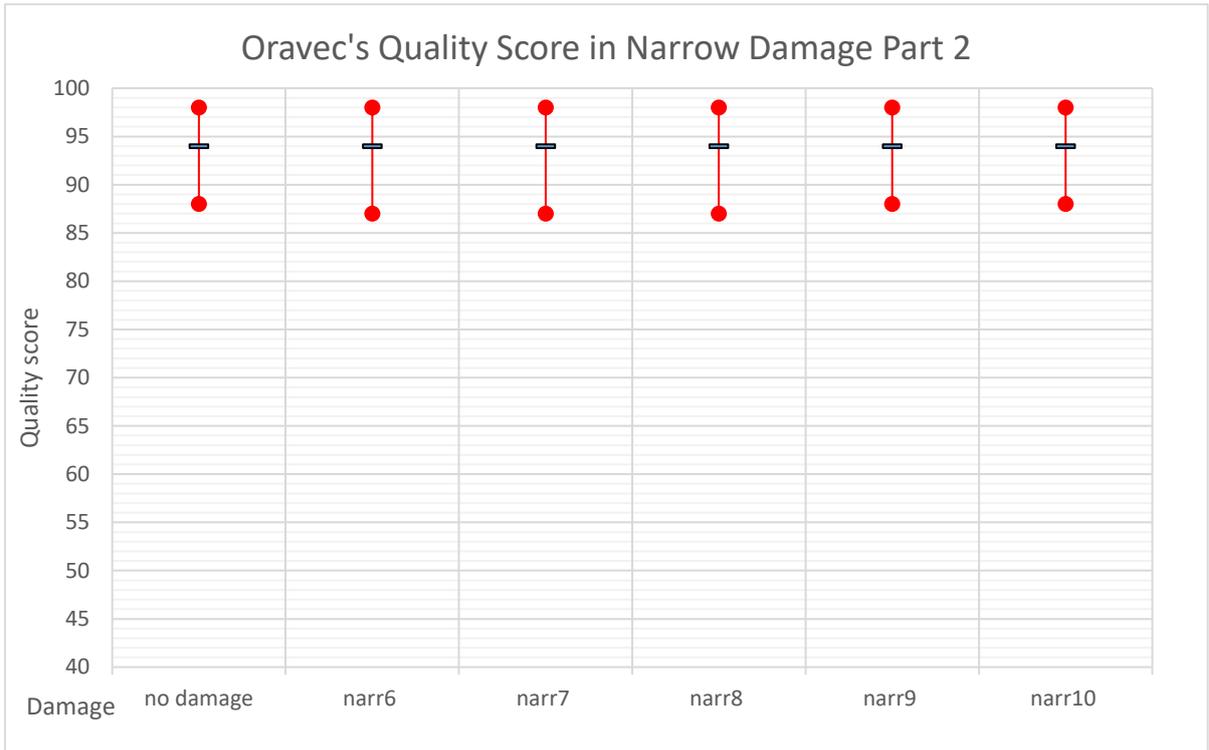


Figure 5.43: Graph of Oravec's quality score in narrow damage – part 2 (normal).

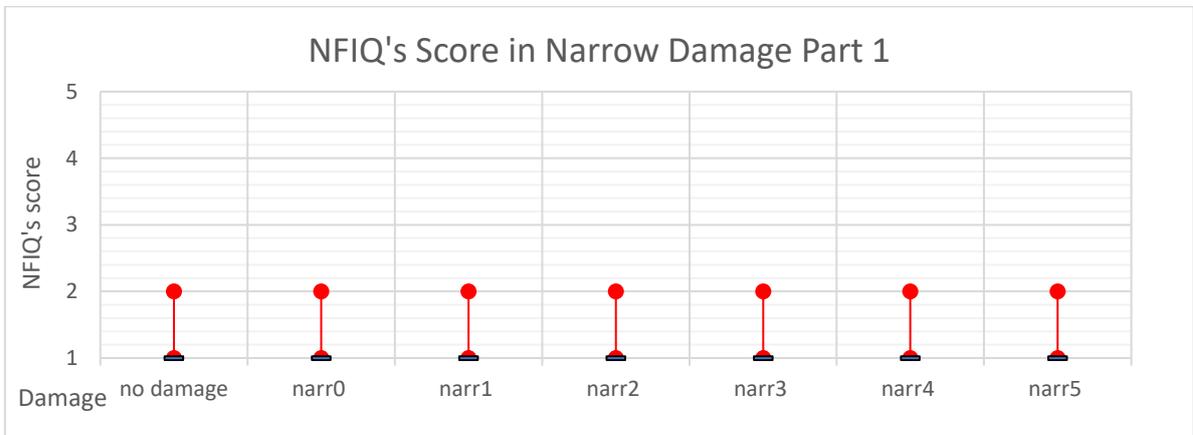


Figure 5.44: Graph of NFIQ's quality score in narrow damage – part 1 (normal).

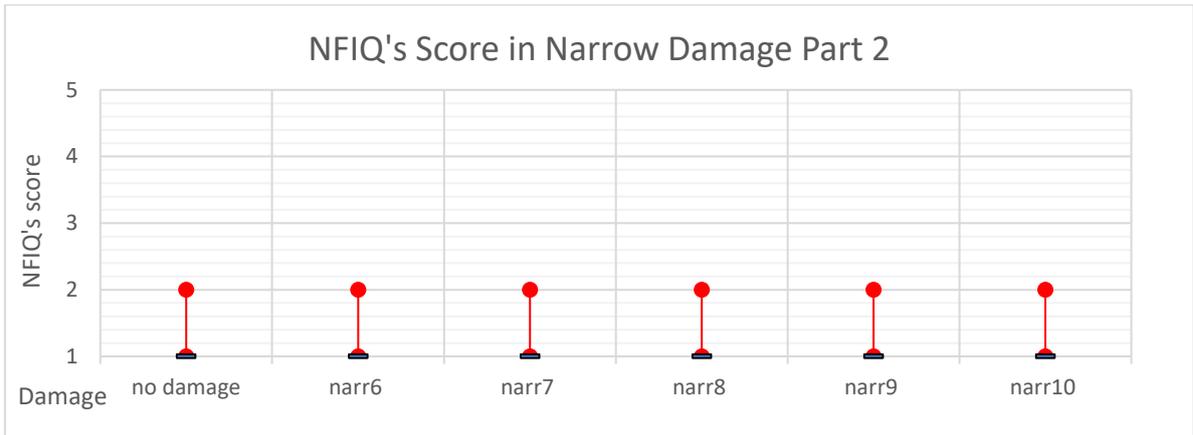


Figure 5.45: Graph of NFIQ's quality score in narrow damage – part 2 (normal).

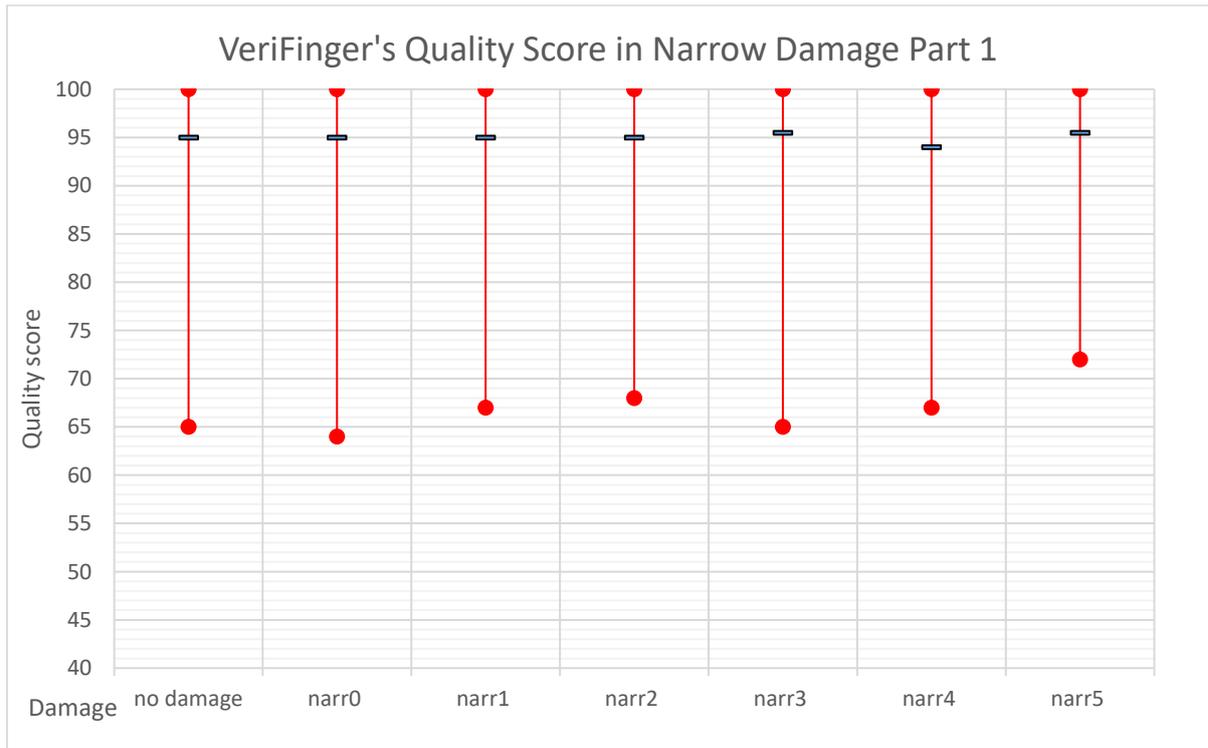


Figure 5.46: Graph of VeriFinger's quality score in narrow damage – part 1 (normal).

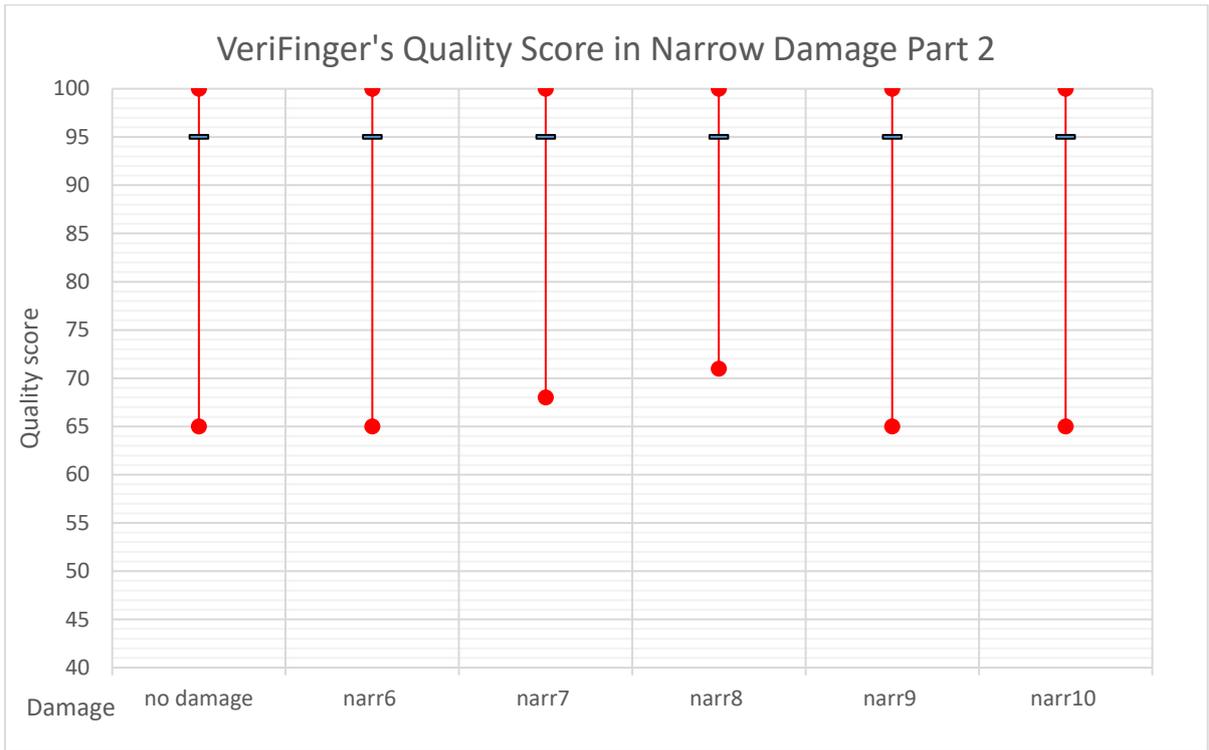


Figure 5.47: Graph of VeriFinger's quality score in narrow damage – part 2 (normal).

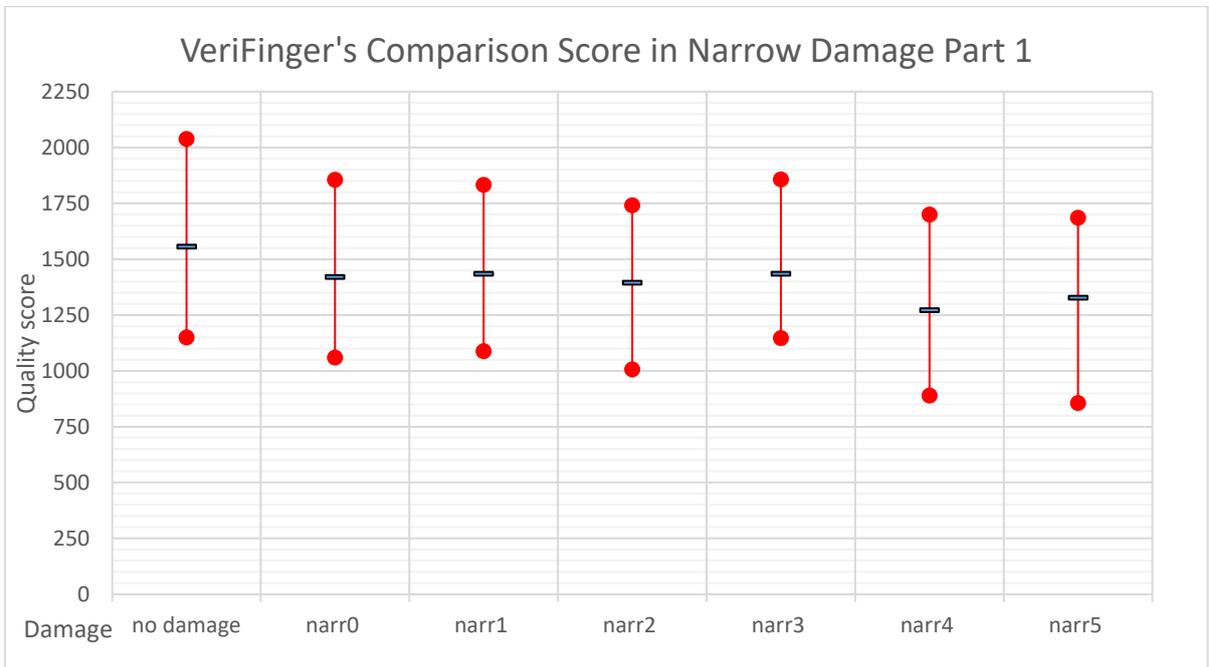


Figure 5.48: Graph of VeriFinger's comparison score in narrow damage – part 1 (normal).

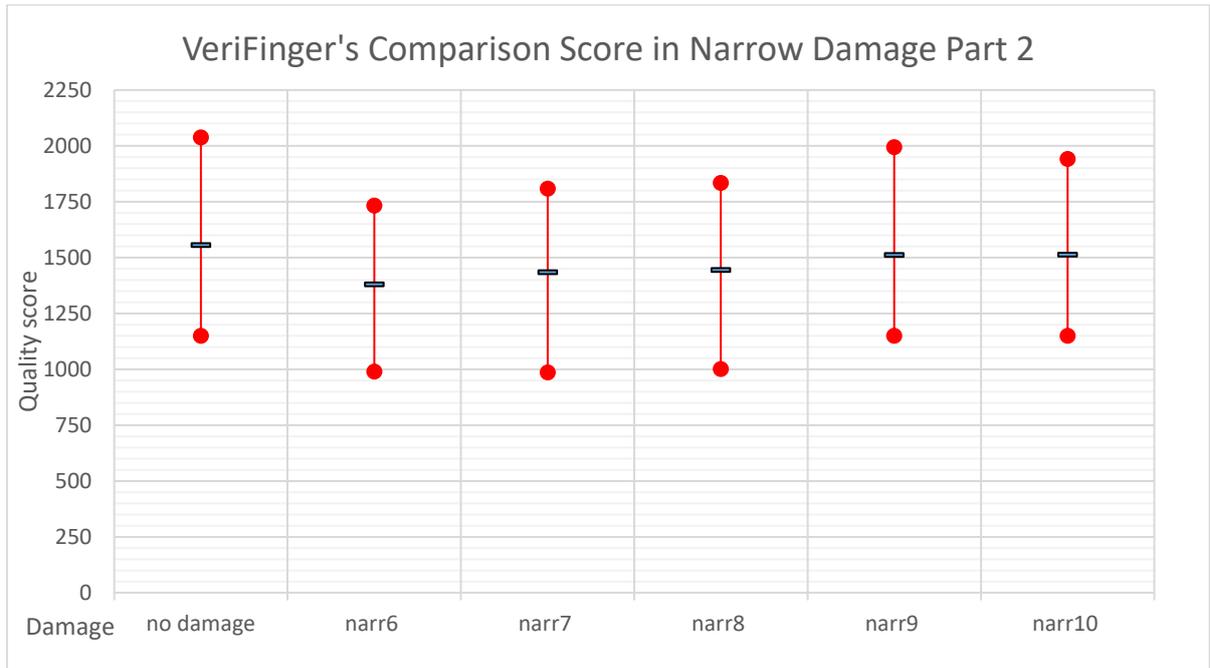


Figure 5.49: Graph of VeriFinger's comparison score in narrow damage – part 2 (normal).

5.5.4.4 Extreme Damages

The same information applies from the Subchapter 5.5.3.4. Once again 172,800 images were generated, with 1,152 different impressions on one image. The seven best combinations based on the frequency and order in the lists were used. The lists were obtained by individual metrics sorted by median and minimal (or maximal in NFIQ) values. Without any further delays, these are the seven damages that were chosen for normal width images:

- **Pm2 dis0 narr4 dmg1:** This damage is composed of high to normal pressure, extreme distortion, side zigzag narrowing, and short sensor damage and can be seen in Figure 5.50b.
- **Pm2 dis0 narr4:** This damage is composed of high to normal pressure, extreme distortion, and side zigzags narrowing and can be seen in Figure 5.50c.
- **Pm2 dis0 narr4 dmg0:** This damage is composed of high to normal pressure, extreme distortion, side zigzags narrowing, and long sensor damage and can be seen in Figure 5.50d.
- **Pm3 dis0 narr4 dmg1:** This damage is composed of low to high to low pressure, extreme distortion, side zigzags narrowing, and short sensor damage and can be seen in Figure 5.50e.
- **Pm1 dis0 narr4 dmg1:** This damage is composed of extreme pressure, extreme distortion, side zigzags narrowing, and short sensor damage and can be seen in Figure 5.50f.
- **Pm2 dis0 narr0 dmg1:** This damage is composed of high to normal pressure, extreme distortion, all sideways sharp narrowing, and short sensor damage and can be seen in Figure 5.50g.

- **Pm2 dis0 narr5 dmg1:** This damage is composed of high to normal pressure, extreme distortion, tip bottom jumpy narrowing, and short sensor damage and can be seen in Figure 5.50h.

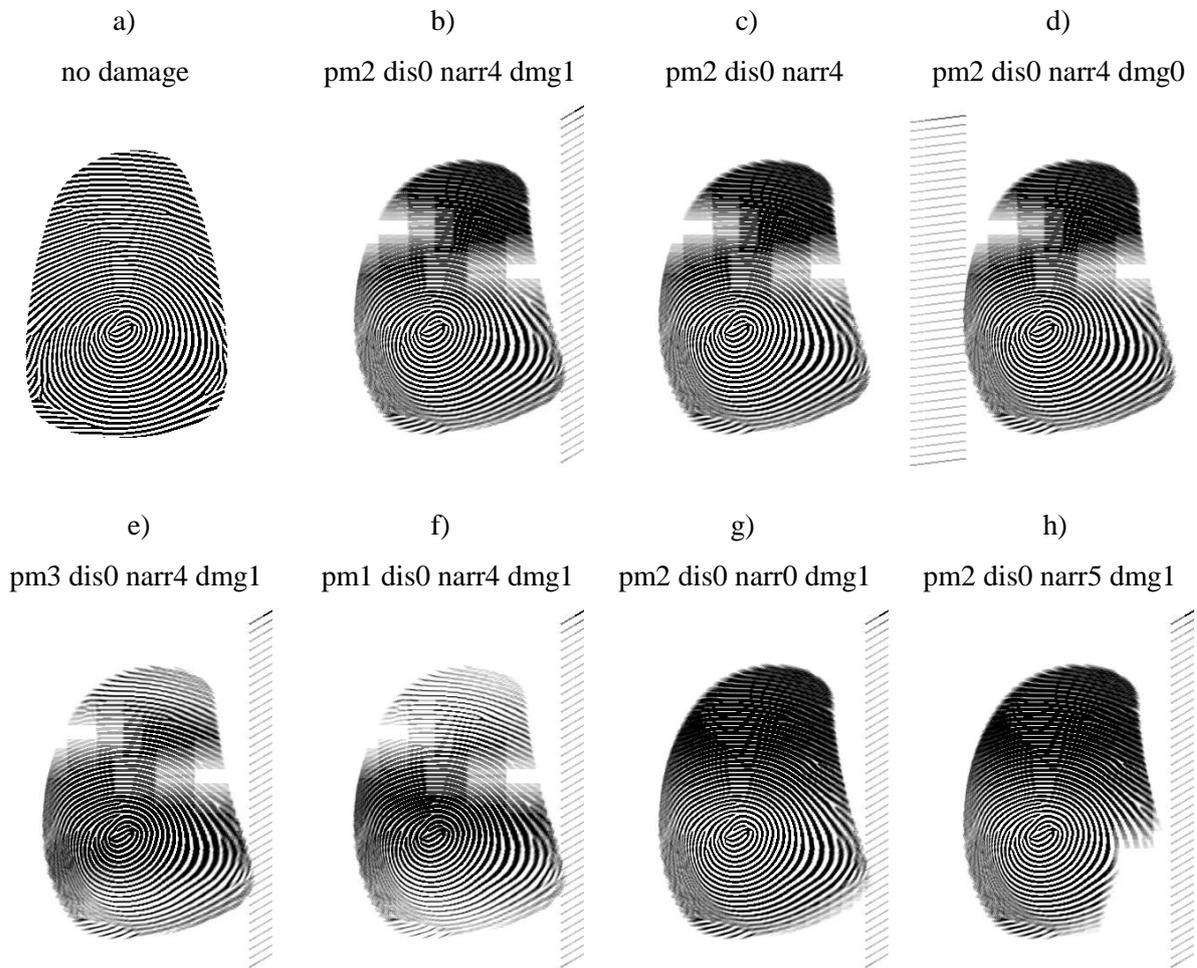


Figure 5.50: Examples of extreme damages in normal width images.

All damages are in the Oravec metric (Figure 5.51) quite below the *no damage* values. In the minimal values, the “*pm2 dis0 narr4 dmg1*” is the lowest; however, what really stands out is the median value of “*pm2 dis0 narr4 dmg0*”, which is quite low in comparison with the others.

Almost all of these combinations were able to get their median value one class higher than the *no damage* in NFIQ metric (Figure 5.52). The only exceptions are “*pm3 dis0 narr4 dmg1*” and “*pm1 dis0 narr4 dmg1*”. Excellent damage from the previous metric “*pm2 dis0 narr4 dmg0*” is the only one whose maximal value is not the worst fifth class.

In Figure 5.53, results from VeriFinger’s quality can be seen. First damage “*pm2 dis0 narr4 dmg1*” stands out in all values. Close to its minimal and median results is the “*pm3 dis0 narr4 dmg1*”, and the similar median value has “*pm1 dis0 narr4 dmg1*”.

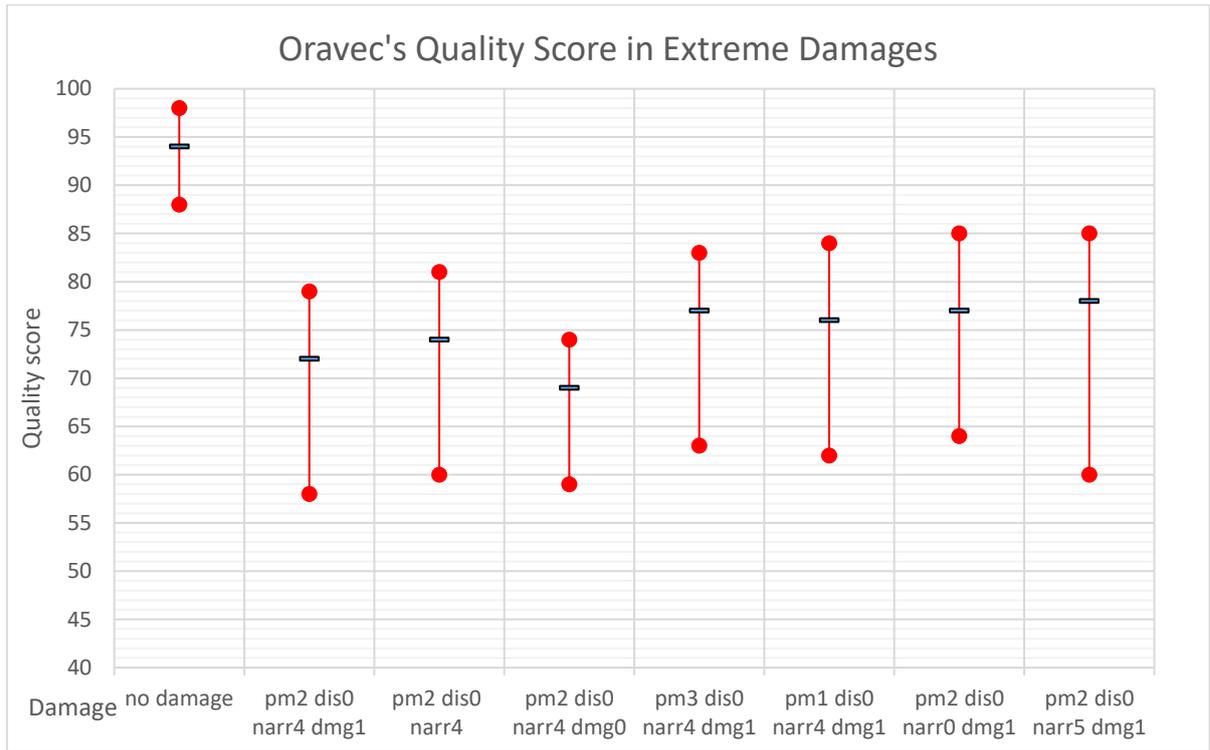


Figure 5.51: Graph of Oravec's quality score in extreme damages (normal).

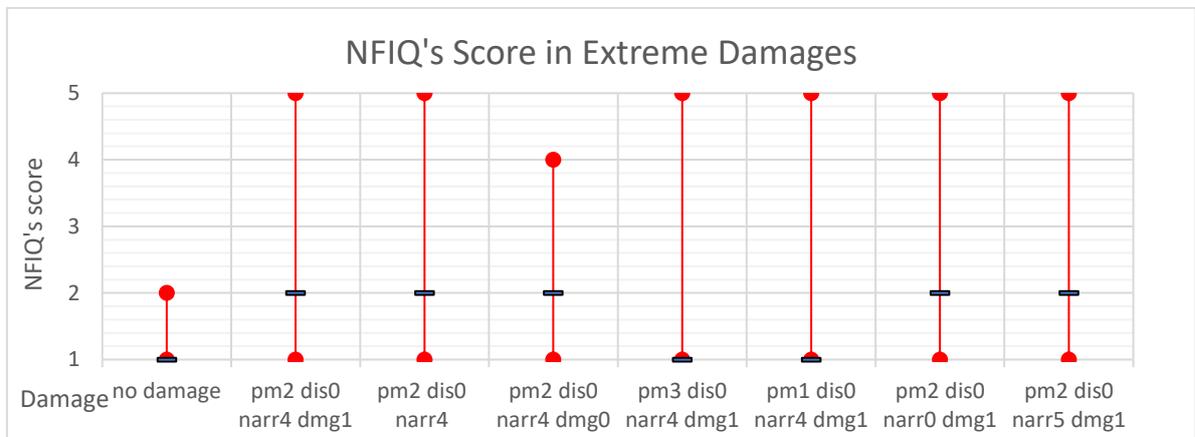


Figure 5.52: Graph of NFIQ's score in extreme damages (normal).

The last metric is the VeriFinger comparison score (Figure 5.54). All damages have really low scores. The best of them is “*pm1 dis0 narr4 dmg1*” with the lowest median and minimal score. The second best is the “*pm3 dis0 narr4 dmg1*” and basically right there, only one point lower in the median score is the “*pm2 dis0 narr4 dmg0*”. To sum up, almost every metric has pointed out different best damage. So the best damages are “*pm2 dis0 narr4 dmg1*”, “*pm2 dis0 narr4 dmg0*”, “*pm3 dis0 narr4 dmg1*”, and “*pm1 dis0 narr4 dmg1*”.

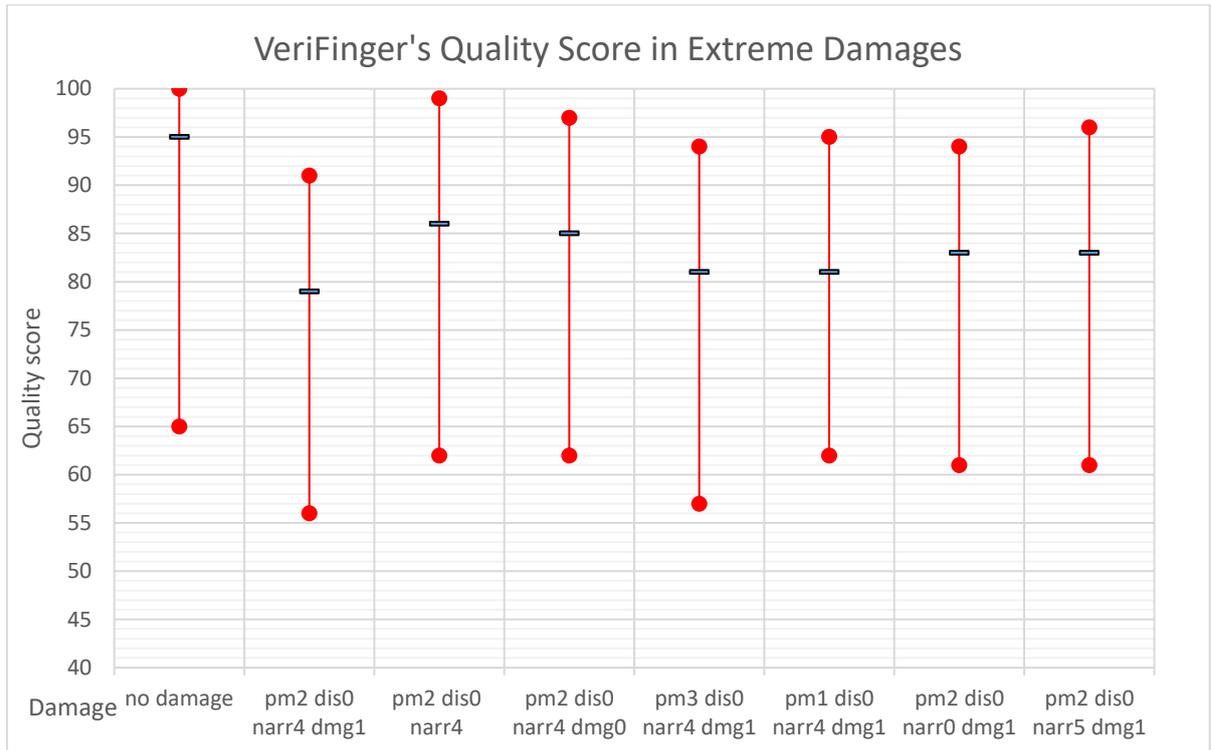


Figure 5.53: Graph of VeriFinger's quality score in extreme damages (normal).

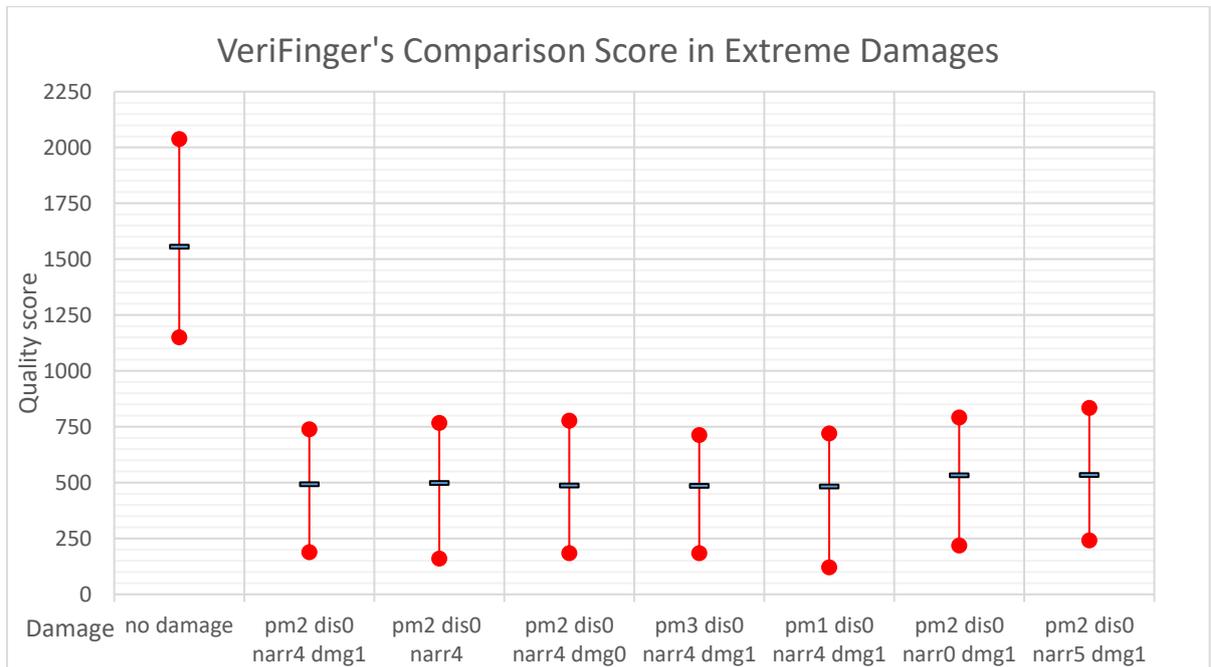


Figure 5.54: Graph of VeriFinger's comparison score in extreme damages (normal).

What is interesting is that the two of them are the same as the best damage for the narrow sensor. Looking at the occurrences in the damages it is strange that there are a lot of changes in regard to pressure and moisture. One could presume that this specific damage is not so important and other

damages are making these good results. On the other hand, *pm2* was not picked as the best in its category and was still able to get to these top results. So it could also be the case that *pm2* works well together with the other damages, but not on its own. The dominant position of *dis0* is clear, but it is hard to enlist *pm2* with the similar dominance. As it was stated in the narrow category, the *narr4* is probably not so dominant, but better than the others in the group. Finally, the damaged sensor was against the presumption that only *dmg1* damage gets to the top. Similarly, as in the previous evaluation the combination of damage is much better than the individual basic damages. Also the damages were almost always lower than the reference *no damage*, so it stands out as the second proof of damaging the fingerprint images.

6 Skin Disease Simulation

The skin is the largest organ in the body, having a surface area of 1.8 m² and making up to 12-15 % of an adult's total body weight. It consists of three layers (see Figure 2.1) [73]: the *epidermis* (the outer layer), *dermis* („true skin”) and *subcutaneous* (fatty) layer. Structure and thickness of the skin vary by site (e.g. thick epidermis on palms and soles due to mechanical protection – up to 1.4 mm). [12] [13] [16] [17] [53]

Skin diseases represent a very important, but often neglected factor regarding fingerprint acquirement. It is not possible to say how many people suffer from skin diseases, as there are so many various skin diseases out there [74] [75]. In a general medical practice, there are about 20-25 % of patients with skin complaints. When using fingerprint recognition technology, one should always keep in mind those potential users who suffer from some skin disease. The situation after successful recovery of a potential user from such skin diseases is, however, very important for the possible further use of fingerprint recognition devices. If the disease has attacked and destroyed the structure of the ridges in the epidermis and underlying dermis (the so-called dermoepidermal junction – the connection of the top two layers of the skin), the ridges will not grow in the same form as before (if at all), therefore this user could be restricted in their future life by being excluded from the use of fingerprint recognition systems, even though their fingers no longer have any symptoms of the skin disease. [12] [13] [16] [17] [53]

Skin is constantly being regenerated. A keratinocyte (“skin cell”) starts its life at the lower layer of epidermis (the basal layer), which is nourished by blood vessels and is supplied with nerve endings from the dermis. The cell migrates upwards from the basal layer to the stratum corneum (the outermost skin layer). During these four weeks the cell undergoes a series of changes, gradually flattening out and moving toward the surface. Then it dies and it is shed. This physiological process can be negatively affected by many skin diseases. The epidermis is not supplied with blood vessels, but it does have nerve endings. The shape of the dermoepidermal junction basically forms the structure of ridges. In the most cases of dermatological disorders one could find a lot of changes in the ultrastructure of the skin, including the epidermis and dermis. There is often inflammation (inflammatory cells), atrophy or hypertrophy, fibrotisation, and many other changes visible under the microscope. These differences result in colour changes (optical characteristics), changes of dermal vessels and capillaries (blood perfusion), and changes of elasticity and thickness of the skin (optical characteristics after pressure change). [12] [13] [16] [17] [53]

The first category of diseases that influence fingerprints are diseases causing **histopathological changes of the epidermis and dermis**. These diseases usually cause problems for all kinds of fingerprint sensors, because they can influence either the colour or internal structure of the skin. The most common representatives of this group are [73] [74]: *hand and fingertip eczema*, *dyshidrosis*, *tinea*, *pyoderma*, *pitted keratolysis*, *pyogenic granuloma*, *systemic sclerosis*, or *Raynaud's phenomenon*. [12] [13] [16] [17] [52] [53] [S1]

The second group are diseases that cause **skin discoloration**. These diseases may cause problems for optical fingerprint sensors and also for sensors that use a fingerprint liveness detection checks based

on the colour or spectral analysis of the human skin [15]. Typical representatives are [73] [74]: *Macular drug eruptions and rashes in infectious diseases (hand, foot and mouth disease, scarlet fever, secondary syphilis, Kawasaki's disease), pitted keratolysis, Raynaud's phenomenon, xanthomas, carotenosis, or hereditary hemorrhagic teleangiectasia.* [12] [13] [16] [17] [52] [53] [S1]

The third and final category consists of diseases that cause **histopathological changes in the junction of the epidermis and dermis**. These diseases could cause structure changes underneath the skin in the junction between the dermis and epidermis – i.e. in the area from which ultrasonic fingerprint sensors acquire fingerprint pattern images. Typical representatives are [73] [74]: *hand eczema, verruca vulgaris (warts), psoriasis, or epidermolysis bullosa.* [12] [13] [16] [17] [52] [53] [S1]

6.1 Database of Fingerprints with Skin Diseases

It is rather difficult to get a fingerprint database with skin-diseased users along with information about their disease. The creation of this type of database is even harder because of the cooperation required of technicians, medical doctors, and patients. On the other hand, there is no other reasonable method of testing how recognition algorithms can cope with skin diseases. [16]

As it has been shown before, working with fingerprints that come from users with skin diseases can be very difficult. Creating synthetic skin disease databases could encourage the development of methods to detect or reconstruct those types of fingerprints. Fortunately, there is a database that can be used. It was created in cooperation with the University Hospital in Olomouc (CZ), the St. Anne's University Hospital in Brno (CZ), and the Darmstadt Hospital in Germany. In Figure 6.1, the workspace which was sent to each institution to acquire fingerprints can be seen. It contains a three-dimensional touchless and touch optical sensor, a swipe and touch capacitive sensor, and a digital microscope. Some institutions also acquired fingerprints using a dactyloscopic card. Each image in the database has anonymized information about the patient, severity, and type of disease. There are 2,165 fingerprints from 44 patients suffering from 12 different diseases in the database. To acquire the database, six different technologies were used – Dinolite (digital microscope), Sagem MSO 300 (optical sensor),

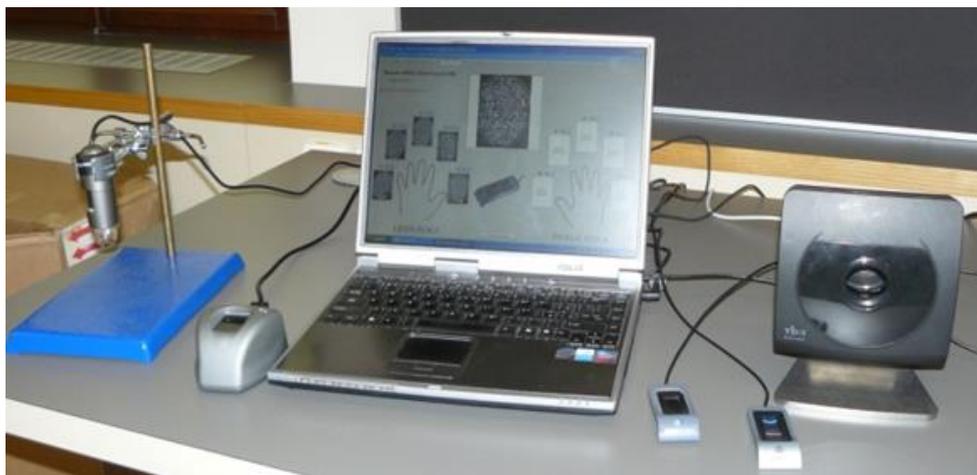


Figure 6.1: Workplace for acquisition of diseased fingerprints.

UPEK EikonTouch500 (touch capacitive), UPEK Eikon II (swipe capacitive), TBS 3D Enroll 2011 (contactless optical multispectral), and a scan of the dactyloscopic card. With the exception of the dactyloscopic card, in that order – from left to right – the sensors can be seen in the Figure 6.1. Images from all sources are shown in Figure 6.2. The size of the database and the described unique information about them is the reason why the database is one of the few, if not the only one in the world. Therefore, it is perfect for studying what damage each disease does and it can also be used for validation. [16] [17] [52] [53] [76] [77]

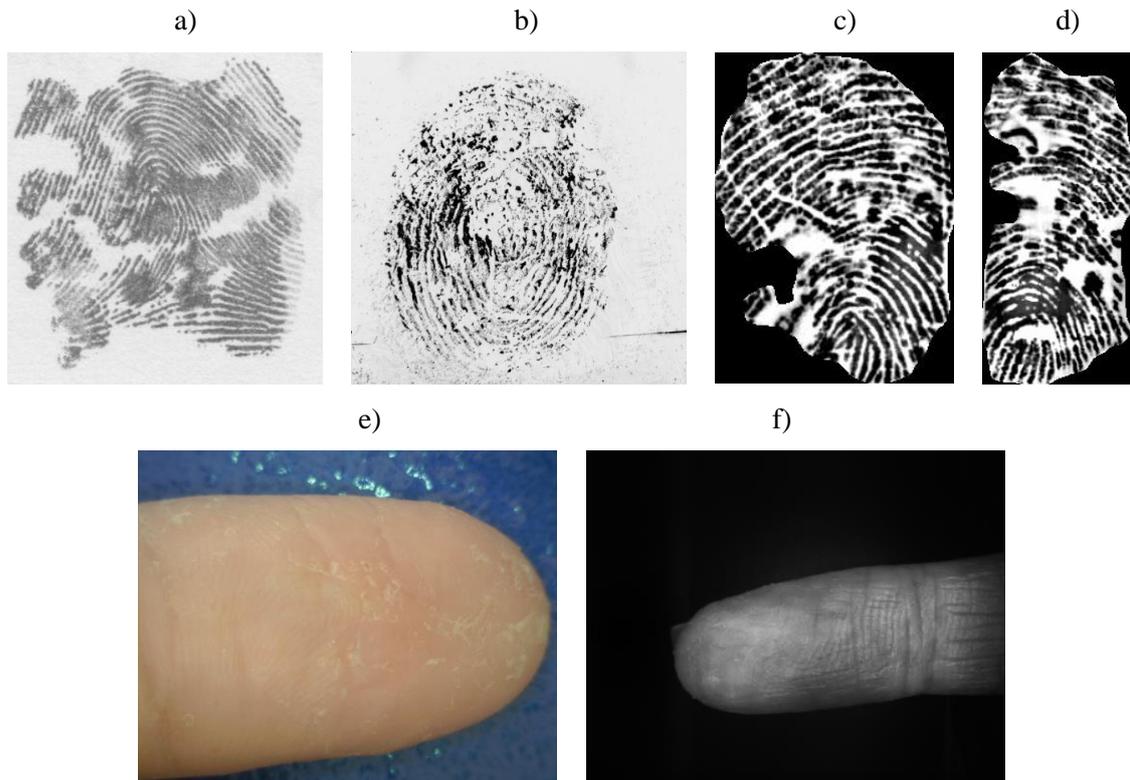


Figure 6.2: Dyshidrotic eczema from a) dactyloscopic card, b) Sagem, c) EikonTouch, d) Eikon II, e) Dinolite, f) TBS 3D Enroll Station.

6.1.1 Database Analysis

The raw, diseased fingerprint database was first analysed in order to provide a solid foundation for future research. For every disease, common signs among all fingerprint images affected by this disease were found, and a general description of each disease and its influences were defined. Based on these descriptions and sets of common signs and their frequencies, the diseased fingerprint images were classified into five categories. These categories are later used in the actual detection of the damaged areas in a fingerprint image and they help to divide the large detection task into smaller bearable parts. Most of the fingerprint images come from a dactyloscopic card. The numbers of fingerprints of each disease is displayed in Table 6.1. [53] [76] [78]

Table 6.1: Database content.

Disease	Number of fingerprints in the DB	Percentages [%]	Number of patients
Fingertip eczema	1,107	51.132	17
Psoriasis vulgaris	326	15.058	9
Dyshidrotic eczema	247	11.409	4
Hyperkeratotic eczema	118	5.450	2
Verruca vulgaris	96	4.434	4
Scleroderma	50	2.310	1
Acrodermatitis continua	40	1.848	1
Colagenosis	36	1.663	1
Raynaud's phenomenon	9	0.416	1
Effusion of fingers	35	1.617	1
Cut wound	18	0.831	2
"Unknown" disease	83	3.834	1
Sum	2165	100	44

By observing and comparing the fingerprint images, 12 common features were defined and seven of them are local features [76]: *straight lines* (SL), *grid* (G), *small ridges disruptions* (SRD), *small "cheetah" spots* (CS), *larger round/oblong spots* (ROS), *large irregular spots* (IS) and *dark places* (DP). The other five are global image patterns [76]: *blurriness of (parts of) the image* (B), *significantly high contrast of the image* (HC), *the entire fingerprint area affected* (EA), *total deformation of the fingerprint image* (TD), and *significantly high-quality and healthy fingerprint* (HQ). For every skin disease, its image features were summarized (see Table 6.2 and Table 6.3). Fingerprint images obtained from optical scanners were excluded as their character is significantly dissimilar to the others. The actual number of images taken into account is stated in the column "sum". [56] [76] [A1]

Table 6.2: Local features of damaged fingerprint images.

Disease	Percentages of particular features [%]							Sum
	SL	G	SRD	CS	ROS	IS	DP	
Fingertip eczema	72.03	24.65	15.91	12.24	32.34	16.61	15.73	572
Psoriasis vulgaris	40.37	6.42	2.75	12.84	48.17	32.57	62.84	218
Dyshidrotic eczema	63.11	7.38	14.75	18.03	78.69	29.51	32.79	122
Hyperkeratotic eczema	3.92	0.00	66.67	15.69	74.51	3.92	5.88	51
Verruca vulgaris	3.17	0.00	14.29	12.70	74.60	0.00	25.40	53
Scleroderma	0.00	0.00	0.00	0.00	0.00	0.00	30.43	23
Acrodermatitis continua	14.29	0.00	0.00	85.71	60.00	14.29	65.71	35
Colagenosis	100.00	78.13	0.00	0.00	15.63	0.00	25.00	32
Raynaud's phenomenon	0.00	0.00	100.00	0.00	0.00	0.00	0.00	8
Effusion of fingers	10.00	0.00	73.33	43.33	63.33	6.67	13.33	30
Cut wound	93.75	0.00	0.00	0.00	18.75	0.00	12.50	16
"Unknown" disease	100.00	86.67	0.00	0.00	76.67	30.00	73.33	30

Table 6.3: Global features of damaged fingerprint images.

Disease	Percentages of particular features [%]					Sum
	B	HC	EA	TD	HQ	
Fingertip eczema	18.01	21.50	40.38	36.36	29.02	572
Psoriasis vulgaris	34.86	27.06	61.93	58.72	18.35	218
Dyshidrotic eczema	30.33	30.33	31.97	29.51	9.84	122
Hyperkeratotic eczema	31.37	29.41	9.80	0.00	37.25	51
Verruca vulgaris	19.05	80.95	7.94	7.94	76.19	63
Scleroderma	0.00	0.00	0.00	0.00	100.00	23
Acrodermatitis continua	48.57	25.71	100.00	100.00	0.00	35
Colagenosis	9.38	40.63	0.00	0.00	25.00	32
Raynaud's phenomenon	0.00	0.00	0.00	0.00	100.00	8
Effusion of fingers	23.33	16.67	40.00	16.67	3.33	30
Cut wound	37.50	68.75	0.00	0.00	50.00	16
"Unknown" disease	30.00	20.00	90.00	83.33	0.00	30

6.1.2 Description of Diseases in the Database

Fingertip eczema (see Figure 6.2), or **atopic eczema**, is a very dry, inflammatory, non-infectious disease that occurs on the palmar surface, or the fingertips. The skin becomes cracked and scaly and usually starts peeling off, which results in the exposure of red and tender skin surfaces [73] [74] [79]. Since the number of fingerprints with fingertip eczema in the database is large, a wide range of typical features were observed. There are two groups of these fingerprints: less and more severely damaged. In the first group of fingerprints, the occurrence of thin lines of different directions was typical. These lines often connect or cross each other. In some cases, small round white spots were present, and in others occasional dark areas made the ridges partially unreadable. In the second group, the damage is more severe. Fingerprints are usually almost completely damaged; straight lines cover the entire fingerprint area and create grids by crossing each other. The background is darker and large irregular spots can be seen. As the ridges cannot be seen at all, this type of damage is by no means recoverable. [12] [13] [53] [76] [80] [A1]



Figure 6.2: Fingertip eczema. Source: database and [73].

Psoriasis vulgaris (see Figure 6.3) is a common, chronic, and inflammatory disease of the skin that is often indistinguishable from a serious form of hand eczema. It is characterized by dry and scaling plaques covered with dry scales that peel in layers [73] [74] [79]. The vast majority of fingerprints affected by psoriasis are completely damaged. Ridges are almost entirely unreadable. The most frequent feature is a large irregular dark spot bounded by a white border. Apart from this feature, the presence of larger dark areas or thick lines is also common, as well as round and oblong spots. [12] [13] [53] [76] [80] [81]

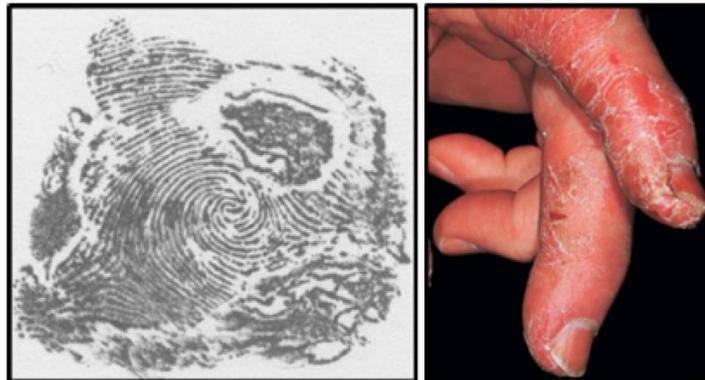


Figure 6.3: Psoriasis vulgaris. Source: database and [73].

Dyshidrotic eczema, also known as **pompholyx** (see Figure 6.4), is a variant of hand and foot dermatitis that makes skin extremely dry. Its typical features are itching vesicles and scales located on the palms and sides of fingers. [73] Fingerprint images damaged by dyshidrotic eczema are typically covered with irregular blurred shapes with no specific form. Another typical feature is a thick line. These fingerprints were divided into two groups, according to how severe the damage is. In the first group of less severely affected fingerprints, the entire area of a fingerprint is often covered, but ridges remain visible. Ridges are usually disrupted at multiple places and irregular, blurred white spots may appear. Fingerprints in the second group are seriously damaged and cannot be repaired. The image area is typically covered by thicker lines in combination with large blurred white spots. Ridges are not sufficiently visible. [12] [13] [53] [76] [80] [A1]

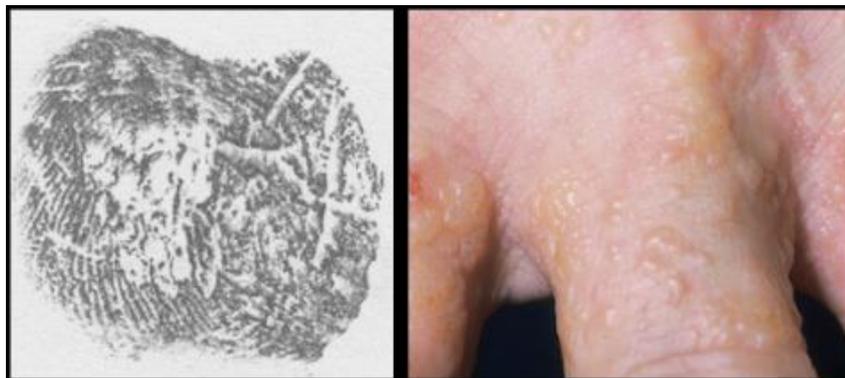


Figure 6.4: Dyshidrotic eczema. Source: database and [73].

Hyperkeratotic eczema (see Figure 6.5) is a chronic form of hand eczema characterized by the occurrence of orange and brown scales with cracks between them [73] [74] [79]. Usually, only one-third to one-half of the fingerprint area is affected and sometimes only the ridges are disrupted. In other cases, however, ridges are distorted and their direction is difficult to determine. Small to medium round spots are likely to be present. [12] [13] [53] [76] [80] [A1]



Figure 6.5: Hyperkeratotic eczema. Source: database and [73].

Verruca vulgaris (warts) (see Figure 6.6) is a very common skin disease, characterized by the presence of stiff elevated bumps on the skin's surface. They grow in size, which is on average about 5 mm, but it can reach up to more than 1 cm. On their surface, tiny black dots may appear [73] [74] [79]. The influence of this disease on the fingerprint images is minor and easily removable. Typically, one to four round white spots occur, sometimes with black dots in their centre. [12] [13] [53] [76] [80] [A1]



Figure 6.6: Verruca vulgaris (warts). Source: database and [73].

Systemic scleroderma (see Figure 6.7) is characterized by the appearance of hard, smooth, and ivory-coloured areas. In the early stage, affected areas are red and swollen; later, they become completely immobile and lose their natural peaked contour [73] [74] [79]. The fingerprints in the database did not show any signs of damage. Therefore, it can be concluded that the number of acquired fingerprints was not sufficient enough to describe the disease's influence on fingerprint images. [12] [13] [53] [76] [80] [A1]



Figure 6.7: Systemic scleroderma. Source: database and [73].

Acrodermatitis continua, or **dermatitis repens** (see Figure 6.8), is a chronic inflammatory disease of the hands and feet and one of the less frequent types of psoriasis vulgaris. The outbreak of the disease is accompanied by the asymmetric formation of pustules of the fingertips and continues with the eruption of fresh pustules with hyperkeratosis and crusting. As the disease progresses, nails can even come off [73] [74] [79]. Fingerprint images are typical for the occurrence of small round spots that look like cheetah skin and usually cover the whole fingerprint area. Larger oblong or round spots occur as well and straight lines or cracks are also not uncommon. Ridges cannot be recognized at all and the original structure of the fingerprint is completely covered. Larger dark areas are often present and the spots can be blurred together. In almost all cases, the fingerprint image is completely damaged and cannot be repaired. [12] [13] [53] [76] [80] [A1]

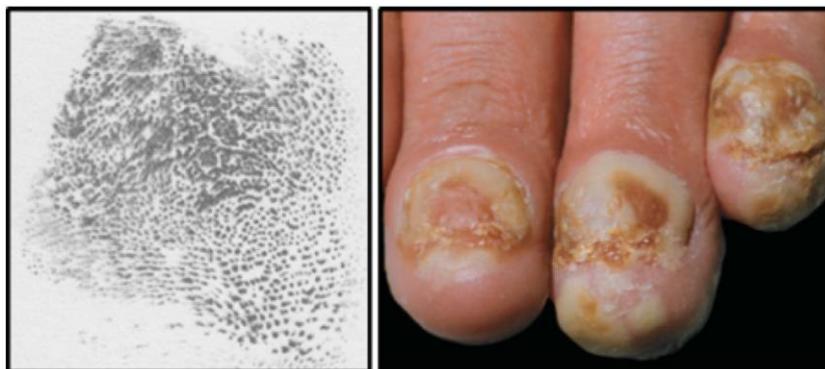


Figure 6.8: Acrodermatitis continua. Source: database and [73].

Colagenosis (see Figure 6.9) is a connective tissue and inflammatory autoimmune disease [73] [74] [79]. The only typical feature of fingerprints with this disease is the thin lines that cross each other. Under these lines, ridges are well visible. [12] [13] [53] [76] [80] [A1]

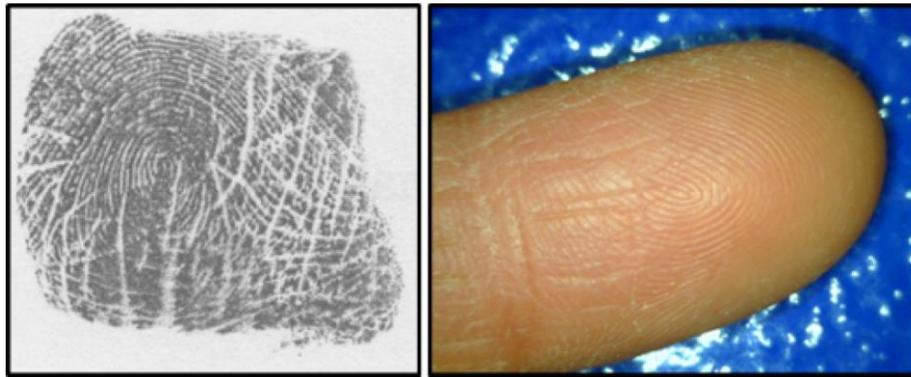


Figure 6.9: Colagenosis. Source: database and [73].

Raynaud’s phenomenon (see Figure 6.10) is a vascular skin disease that often accompanies an associated disease (most often scleroderma). The fingers have sequential discolorations: they first become pale and cold, then white, blue, and finally red. This is caused by constrictions of the small arteries and arterioles in fingers [73] [74] [79]. As Raynaud’s phenomenon causes discoloration only, fingerprints in the database are always undamaged and appear healthy. [12] [13] [53] [76] [80] [A1]



Figure 6.10: Raynaud’s phenomenon. Source: database and [73].

Although being stated as a disease in the database, the **effusion of fingers** (see Figure 6.11a) is only a syndrome, which manifests itself by a strong swelling. It is one of the symptoms of systemic scleroderma, for instance. Ridges are typically disrupted in many places and small to medium spots are present. In general, the ridges are clearly visible, however, sometimes white spots make them unreadable. A **cut wound** (see Figure 6.10b) typically causes either a straight line in a fingerprint image or a more blurred white area. The damage is minor and should not be difficult to remove. Fingerprints of the “**unknown**” or “**unnamed**” disease (see Figure 6.10c) are totally covered with lines of different thickness and length, therefore unreadable. They are very similar to those with fingertip eczema. [12] [13] [53] [76] [80] [A1]

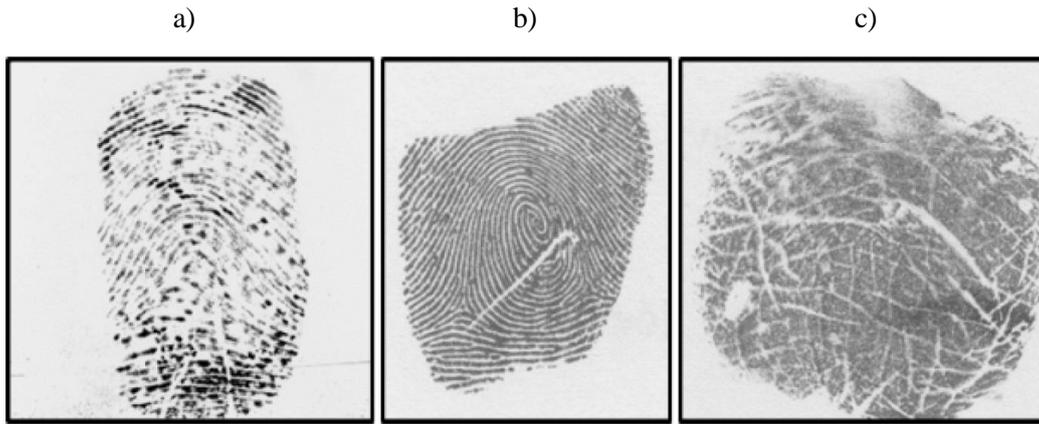


Figure 6.11: a) Effusion of fingers, b) cut wound and c) “unknown” disease.

6.1.3 Skin Diseases Detection

In this subchapter, the specific algorithms used in the disease detector along with their pros and cons are discussed, as well as the core methods essential for the detection functionality. For the *detector*, there are three major algorithms that are used for the detection part: *block orientation field*, *histogram analysis*, and *flood fill*. Their combination provides valuable information about the fingerprint quality and character of the possible disease. [17] [53] [76] [A1]

The computation of the *block orientation field* is commonly used in the fingerprint recognition process for the purposes of estimating the ridges direction and classifying the fingerprint image into one of the several fingerprint classes [6] [17]. Because a typical fingerprint pattern consists of alternating dark and white lines, this information can be easily processed by a gradient operator that estimates the image gradient for each pixel. This low-level information is gathered and averaged for each $w \times w$ block in the image [76]. The transformation can result in a relatively smooth and continual image of the ridges direction estimates – for a healthy fingerprint, of course. If the computation of the block orientation field for a damaged or partially damaged fingerprint is attempted, there can be easily recognizable areas that contain possible damage, because the orientation field in those areas will be discontinuous. These discontinuities can be detected by scanning the field for differences in the direction angles. In the detector’s pipeline, a gradient-based method of the block orientation field computation is used [76]. The resulting block orientation field is afterwards analysed for any discontinuities that may occur. The analysis is done using a row-wise and column-wise scanning approach that reveals areas of possible damage in the fingerprint. Sometimes, the method detects single discontinuities that may be erroneous, but on the other hand (under different circumstances), one unmarked block may appear in the midst of discontinuous blocks. In order to make the algorithm as accurate as possible, although mistakes never disappear completely, these cases are taken into account. The algorithm handles them by copying the properties of their neighbouring blocks (marking the single ones either as alright or as a discontinuity, depending on the neighbourhood). An example of the detection is shown in Figure 6.12. [17] [53] [76] [A1]



Figure 6.12: Damaged area detected using the block orientation field. [76]

The method of *histogram analysis* is based on the presumption that a quality fingerprint image consists of equally distributed ridges and valleys. If it is assumed that ridges are roughly the same dark colour while valleys are light-coloured, a histogram computed from each subfield of the fingerprint's area should ideally consist of two peaks of approximately the same height and one valley between them. Examples of good (bimodal) and bad histograms are presented on Figure 6.13. On the other hand, the intensity distribution in a fingerprint image part that belongs to a damaged area is not always as equal as in the quality one. Thus, if a histogram is computed for this subfield, it is very likely that it will not have the ideal bimodal appearance as described above. Experiments showed that the majority of damaged areas break the rules of the bimodal histogram. The lower the quality, the less the histogram resembles the ideal one. A non-bimodal histogram always implies a damaged or low-quality area. Figure 6.14 shows an example output of this method, along with the particular histograms that were being analysed. A red background implies an invalid histogram, green means valid, and blue stands for background. [17] [53] [76] [A1]

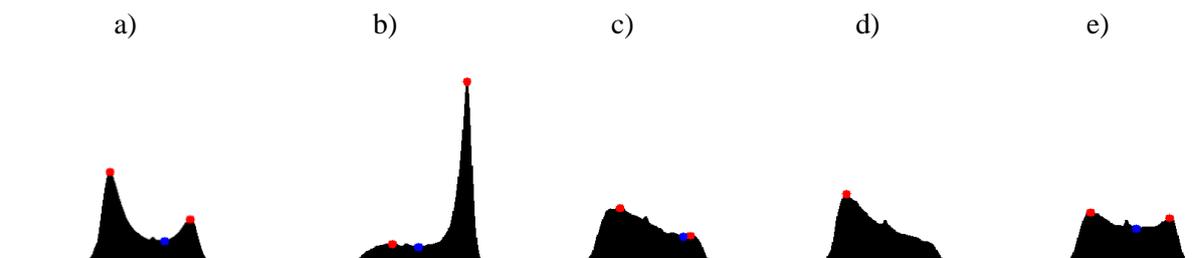


Figure 6.13: An ideal bimodal histogram (a) and examples of bad histograms (b, c, d, e). [76]

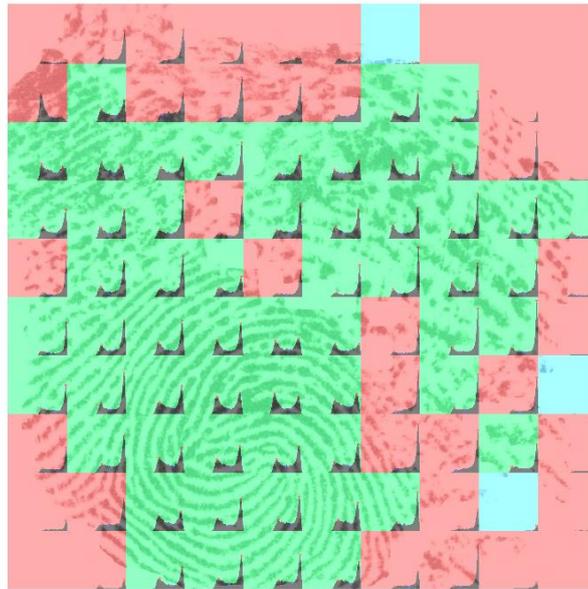


Figure 6.14: Histogram analysis result with particular histograms. [76]

Flood fill is commonly used algorithm used for graphical purposes [82], and it is especially handy for detecting and filling the connected, single-coloured areas of an image. This characteristic was used in order to find local features of damaged fingerprints, such as straight lines or spots. The *flood fill* algorithm has three parameters: a target colour, a replacement colour, and a start pixel. It is based on examining the colour of all pixels in the 4- or 8-neighborhood of the start pixel and changing the colour of those pixels that have the target colour to the replacement colour. Using either recursion or stack/queue, the coloured pixels become the next start pixels and the process is repeated. In the end, the entire single-coloured area is filled. The result from this algorithm is shown in Figure 6.15. [17] [53] [76] [A1]



Figure 6.15: Extraction of straight white lines. [76]

All three of these methods detect a different kind of damage in the image and only flood fill provides logically structured results that can be used in classification. However, connecting the three methods together results in a surprisingly accurate description of the extent of damage in the entire area of a fingerprint image. At the end of each detection process, every image pixel is assigned a value

between -1 and 1. Negative values stand for background, zero means a healthy area, and positive values indicate damage. The higher the value, the more damaged the area to which the pixel belongs. [17] [53] [76] [A1]

Each of the three detection methods separately provides interesting outputs, but it is their connection that makes the resulting application so notable. Thanks to the connection, very satisfactory results have been achieved in locating the damaged areas – see the results on Figure 6.16 (the colour representation is as follows – green marks the healthy areas, blue highlights the background, and for the damaged areas a scale from yellow to red is used; yellow stands for minor damage, whereas red implies extremely damaged places). [17] [53] [76] [A1]

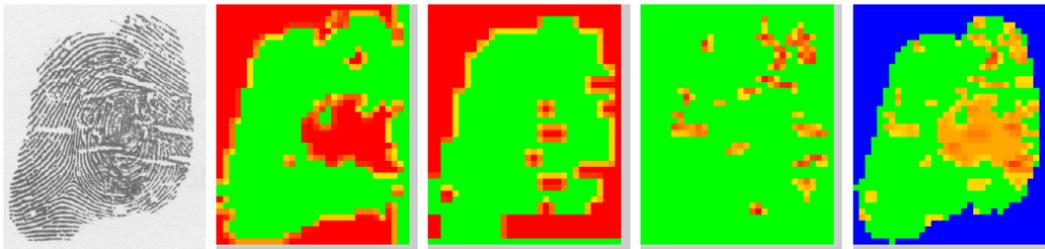


Figure 6.16: Example of the final distribution of damage in the image with atopic eczema. [76]

The *Classifier* decides based on the features extracted by the flood fill algorithm and classifies the fingerprint image according to the feature’s numbers, sizes, and shapes into one of these six categories: acrodermatitis, atopic eczema, psoriasis, verruca vulgaris, unknown disease, or healthy. The results of the described detector are in Table 6.4 and 6.5. [17] [53] [76] [A1]

The classification accuracy reached high values for acrodermatitis (83.5 %) and verruca vulgaris recognition (75.0 %), whereas it was lower for atopic eczema (45.3 %) and psoriasis (58.3 %). Better performance could be gained by improving the classification decision rules, as well as coming up with new types of feature detection. [53] [76] [A1]

Table 6.4: Rejected and accepted samples.

	True Positives	False Negatives	False Positives	True Negatives
Acrodermatitis	12	18	103	478
Atopic eczema	134	289	25	163
Verruca vulgaris	23	17	314	257
Total	611	611	611	611

Table 6.5: Classifier accuracy measures.

	False Accept Rate	False Reject Rate	F1 score [83]	Accuracy
Acrodermatitis	0.1394	0.6667	0.1655	0.8347
Atopic eczema	0.1968	0.7021	0.4300	0.4533
Verruca vulgaris	0.3408	0.7373	0.1956	0.5827
Total	0.2329	0.5000	0.2073	0.7496

6.2 Directly Simulated Diseases

To create the impression of fingerprints having a skin disease it is necessary to implement an algorithm that is designed to damage the master fingerprint and make it look like the fingerprint from a diseased finger. The first method is to base the algorithm on the findings from Subchapter 6.1.1. That means simulating the seven local and five global markings that can be found on the diseased fingerprints. After that, the damage done to a fingerprint with these markings is based on the **probabilistic distribution** of markings in the specific diseases. This way is very dependent on the analysis of the available database, but it can create every disease that is in the database. The second method for creating algorithms that will damage the fingerprints is based on **study of the diseases** one by one. By conducting a thorough analysis of a specific type of damage a unique algorithm can be created. There could even be a few algorithms based on, for example, disease severity. Using the second method will create more precise results for the damage, but only for a few of diseases. Sometimes the effects of diseases are difficult to generalize. In that case, it might be enough to **adapt damage from the existing** fingerprint images to a synthetic one. This subchapter focuses on the algorithms based on the second approach.

6.2.1 Verruca Vulgaris (Warts)

The first skin disease chosen for simulation are warts, specifically common warts (*verruca vulgaris*). In the following subchapters, the disease is described in detail (adapted mainly from [67] [79] [84]), the analysis of selected representative set of wart-affected fingerprints is conducted, and a method of simulating similar fingerprint images is proposed.

6.2.1.1 Detailed Disease Description

Warts are caused by the human papillomavirus (HPV), which belongs to a group of papovaviruses. There is more than a hundred types of HPV, and the gene sequences of HPVs throughout the world are similar. Most of them cause specific types of warts and favour certain anatomic locations, such as plantar warts, common warts, genital warts, and so on. HPV infection is very common amongst the global population, as most people will experience a form of it during their lifetime. HPV has coexisted with humans for millennia, and humans are also their primary host. HPV has been a successful pathogen for humans because they evade their immune response. Common warts are the most-spread variant of warts (affecting approximately 10 % of the population [85]) and are usually cause frustration on the part of the patient. Social activities can be affected, lesions can be uncomfortable or bleed, and treatment is often painful and frustratingly ineffective [86]. The frequent immersion of hands in water is a large risk factor for common warts. People working with raw meat also have a high incidence of common warts of the hands. [12] [13] [67] [80] [84]

Common warts are usually located on the hands, favouring the fingers and palms (see Figure 6.17). Periungual warts are more common in nail biters. Fissuring may lead to bleeding and tenderness. Lesions range in size from pinpoint to more than 1 cm, most averaging about 5 mm. They grow in size for weeks to months and usually present as elevated, rounded papules with a rough, greyish surface. In some instances, a single wart (mother wart) appears and grows slowly for a long time, and then suddenly many new warts erupt. On the surface of the wart, tiny black dots may be visible, representing thrombosed, dilated capillaries. Warts do not have fingerprint folds, as opposed to calluses, in which these lines are accentuated. [12] [13] [67] [80] [84]



Figure 6.17: Common warts on hands and fingers [79].

The treatment for common warts involves two basic approaches: destruction of the wart and the induction of local immune reactions. Destructive methods are most commonly used as initial therapy by most practitioners. Cryotherapy is a reasonable first-line therapy for the most common warts. The wart should be frozen adequately to produce a blister after one or two days. An alternative method of treatment gaining on popularity is a pulsed dye laser treatment, which is both effective and safe for the patient. [12] [13] [67] [80] [85]

6.2.1.2 Warts-affected Fingerprint Analysis

The database contains fingerprint images acquired by various methods and sensors. To study the possible differences between images acquired by different sensors, three fingerprints of the same finger affected by warts have been chosen (see Figure 6.18). In Figure 6.18ab, it can be seen that the wart is located just on top of the whorl. It is a white oval with an irregular border. Inside the oval are black dots. The ridge structure is completely disrupted by the wart. However, the ridge flow continues normally around the border of the wart. Figure 6.18c, acquired by the UPEK sensor, cuts the wart out of the image completely. [67] [84]

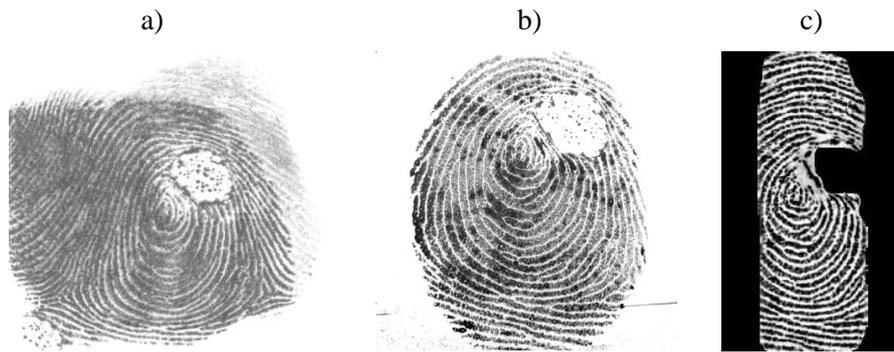


Figure 6.18: Same fingerprint affected by warts acquired by different sensors – a) dactyloscopic card, b) Sagem MSO 300 and c) UPEK Eikon II.

Three different fingerprint images affected by warts (see Figure 6.19) that have been captured by a single sensor, in this case the Sagem MSO 300, are being compared. The first image in Figure 6.19a shows a fingerprint with a clean ridge structure except for the part where the wart is located. The wart is located near the right border of the image and it is represented by a white circle-shaped object with several black dots irregularly spread over its surface. In the upper part of the fingerprint image, another small oval-shaped structure can be seen. It could be a small wart that has spread from the larger one. When the wart is relatively small, usually it contains little or no black dots at all. In the second part (Figure 6.19b), a single large wart is located near the whorl. Black dots on top of it represent the hard and scaly skin of the wart. The wart is irregularly shaped, and its border is well-defined. The ridge structure around the wart is mildly deformed and the ridges are compressed. However, except for the close surroundings of the wart, the ridge structure of the fingerprint is unaffected. The final image in Figure 6.19c shows a fingerprint that has been affected by warts in a large area of its surface. There are at least three large, white, and oval-like objects with irregular borders near each other. The warts affect the ridge structure near their edges in a similar manner to the wart described previously. [67] [84]

From the database of fingerprints, it has been found that the size of warts on fingers varies from very small to ones that are as large as half of the hypothetical radius of the fingerprint. The location of warts on a fingerprint is completely random and oftentimes, one wart produces other so-called satellite warts in its proximity. [67] [84]



Figure 6.19: Different fingerprints affected by warts acquired by Sagem MSO 300.

6.2.1.3 Design of a Method for Warts Simulation

Based on the analysis of the existing fingerprints with warts, a design of a method of disease simulation is proposed. The algorithm consists of the following steps:

1. Localise the fingerprint area in the image.
2. Determine the new wart size and locate its centre point on the fingerprint.
3. Draw the wart into an image buffer.
 - a. Create an empty image buffer.
 - b. Generate a number of small circles around the centre point of the wart.
 - c. Draw the generated circles into the buffer.
 - d. Draw dark dots inside the wart.
 - e. Determine the final colour of each wart pixel.
 - f. Blur the wart in the buffer
4. Draw the wart from the image buffer into the fingerprint image.
5. Generate possible secondary warts.

In order to generate warts into the image with a synthetic fingerprint, the fingerprint has to be localised in the input image first. This is done in *step 1*. First, an adaptive thresholding is applied in order to clearly separate the fingerprint structure from the background. Next, the image is blurred so that the fingerprint ridges connect and the contours can be localised in the image. The contour of the largest area is then selected as the fingerprint contour (see Figure 6.20). This contour defines the border of the fingerprint. In *step 2*, a centre of the new wart is localised. The point coordinates are randomly generated and are used only if they comply with the requirements (location inside of the fingerprint, minimal distance from the fingerprint border). Also in this step, the size of the generated wart is randomly determined within the set boundaries. [67] [84]

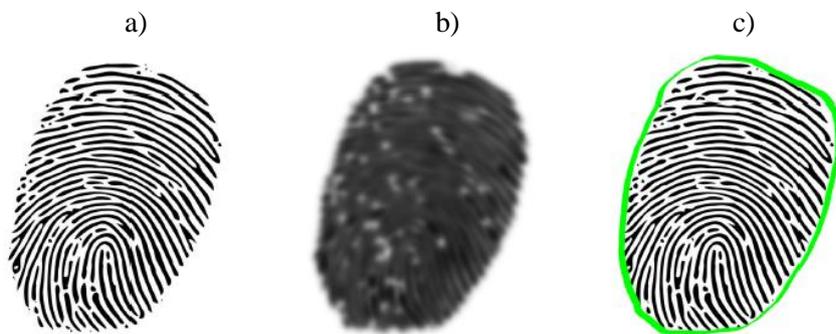


Figure 6.20: Fingerprint area localisation (*step 1*) – a) original fingerprint, b) blurring applied and c) fingerprint contour. [67]

Each new wart is drawn into its own image buffer, so it does not interfere with the rest of the fingerprint. This is done in *step 3* (see Figure 6.21). First, an empty buffer large enough for the new wart to fit in is created (*step 3a*). In the following *step 3b*, a number of small circles of varying radii are generated with their centres being distributed with an exponentially large distance from the wart's centre point. Each of the circles are defined by their radii and centre point. The centre point is determined by a calculation of coordinates using a randomly generated angle α and a randomly

generated radius r . Using a simple equation (Eq. 6.1), the centre point $[x_{circle}, y_{circle}]$ of a new circle is calculated: [67] [84]

$$\begin{aligned} x_{circle} &= x_{centre} + r \cdot \cos \alpha \\ y_{circle} &= y_{centre} + r \cdot \sin \alpha \end{aligned} \quad (\text{Eq. 6.1})$$

where $[x_{centre}, y_{centre}]$ is the wart's centre point. The drawing of the generated small circles is then done in two steps. First, the circles are drawn in a distinctive colour representing the outline of the circles. Afterwards, the same circles are drawn only with their radius smaller by one to two pixels. This creates a desired effect of a wart composed of a number of small circles with only the outline of the whole wart object visible. [67] [84]

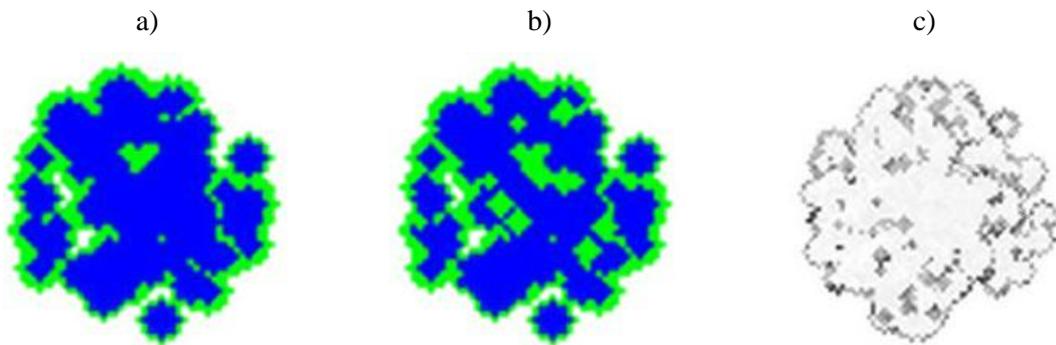


Figure 6.21: Process of wart drawing into an image buffer (enlarged) – a) circles, b) dark dots, c) pixel colouring. [67]

In *step 3c*, the previously generated circles are drawn into the buffer. The final drawing step is *step 3d* in which the dots with randomly generated coordinates are drawn onto the warts surface. The dots are drawn with the same colour as the border of the small circles in the previous step (Figure 6.21b). [67] [84]

With the wart shape drawn in the buffer, the algorithm proceeds with *step 3e* where the final colour of each pixel of the wart is determined (Figure 6.21c). Depending on if the pixel is drawn by the colour of the border or the colour of the inside of the wart, the colour of the neighbouring pixels in the

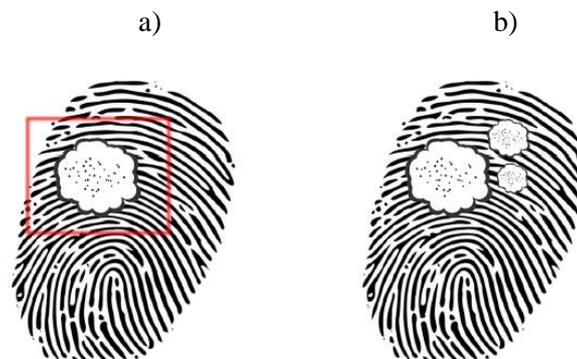


Figure 6.22: Final wart drawing – a) wart drawn into original image, b) secondary warts. [67]

original fingerprint image is acquired (in this case, dark pixels for the border and light ones for the inside of the wart). The final pixel colour is then determined by one of the two following methods. The first method picks a random neighbouring pixel and copies its colour. The second method computes the mean colour of all the neighbouring pixels and then the mean colour is computed and applied to the pixel. Blurring of the wart with a kernel size of 3×3 in the image buffer follows, making the image less sharp and more realistic. Finally, in *step 4* the buffer is drawn into the original fingerprint image, taking into consideration the transparency of the pixels in the buffer and blending them into the original image appropriately (Figure 6.22a). Eventually, secondary warts are drawn into the fingerprint if required in *step 5*, following the same steps of the algorithm as for the main wart (Figure 6.22b). The only difference is an added requirement not to overdraw the already existing warts in the image. There is a 50 % chance of secondary warts until the defined maximum number is reached. [67] [84]

6.2.1.4 Examples of Warts Damage

Five different settings of warts were simulated. The list contains the damage name, testing shortcuts, and a short description with settings. The settings for warts contains secondary wart maximal count (SMC), distance from border (BD), minimal size (MIN), and maximal size (MAX) – the last three values are in percentage of the fingerprint radius. For simplicity, the minimal border distance is at 20 % and the neighbour colour method is set to random. Real images of warts can be seen in Figure 6.18 and Figure 6.19, with generated impressions in Figure 6.23.

- **Small warts with no secondary warts (warts0):** The synthetic image can be seen in Figure 6.23a. Settings: SMC 0, MIN 5, MAX 10.
- **Small warts with up to two secondary warts (warts1):** The synthetic images can be seen in Figure 6.23bc. Settings: SMC 2, MIN 5, MAX 10.
- **Large warts with no secondary warts (warts2):** The synthetic image can be seen in Figure 6.23d. Settings: SMC 0, MIN 10, MAX 15.
- **Large warts with up to two secondary warts (warts3):** The synthetic images can be seen in Figure 6.23ef. Settings: SMC 2, MIN 10, MAX 15.
- **Extreme (warts4):** An extreme case is set to generate even larger warts than in previous examples. Up to three secondary warts can also be present. The synthetic images can be seen in Figure 6.23gh. Settings: SMC 3, MIN 15, MAX 20.

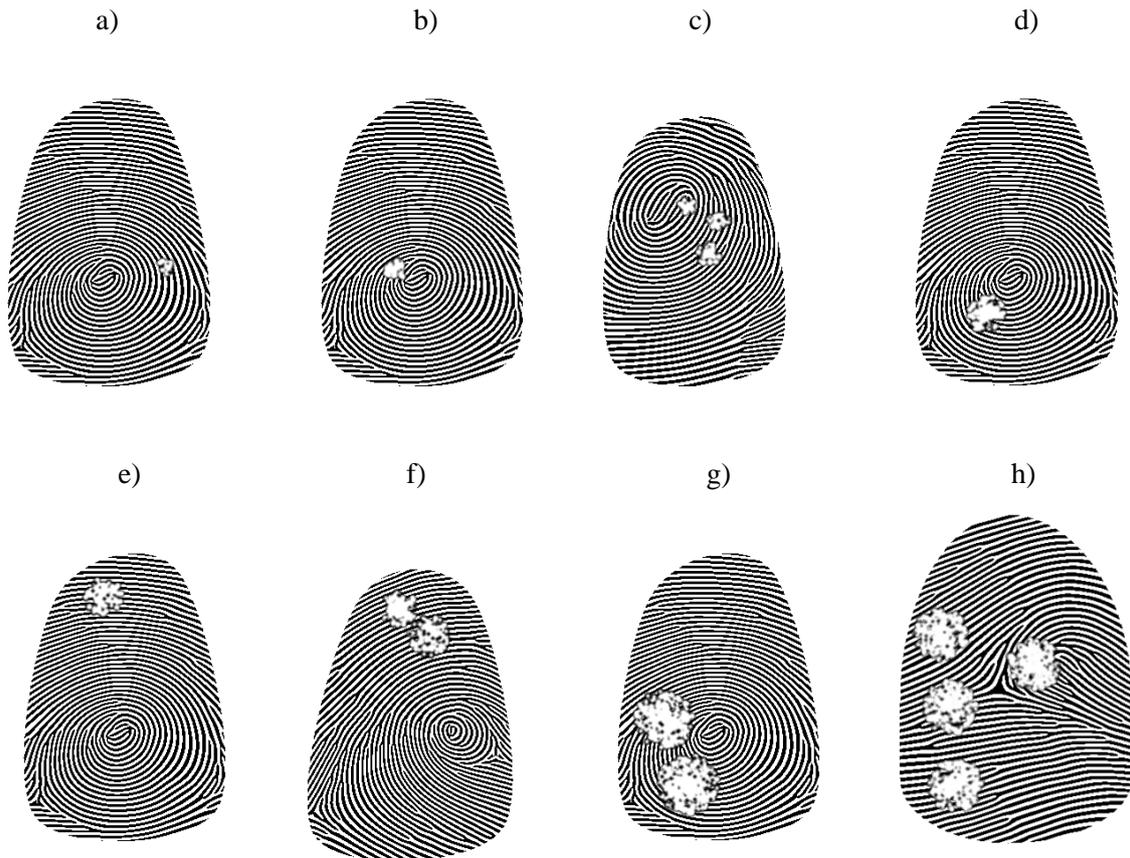


Figure 6.23: Examples of warts disease damage (a) warts0, b) c) warts1, d) warts2, e) f) warts3, g) h) warts4).

6.2.2 Atopic Dermatitis

The second chosen skin disease is atopic dermatitis (also known as atopic eczema). In the following subchapters, this disease is described in detail with a focus on hand eczema (adapted from [79] [67] [84]). The analysis of a selected representative set of atopic dermatitis affected fingerprints is conducted and a method of simulating fingerprint images is described.

6.2.2.1 Detailed Disease Description

Atopic dermatitis (AD) [79] is a chronic, inflammatory skin disease that is characterized by pruritus and a chronic course of exacerbations and remissions. The prevalence of AD increased dramatically in the last half of the twentieth century, becoming a severe health problem in many countries. Rates of AD are around 15-20 % worldwide, with up to 20 % of children affected by the disease. [12] [13] [67] [80] [84] [87]

The skin, in general, is dry and somewhat erythematous. Lichenification and prurigo-like papules are common. Papular lesions tend to be dry, slightly elevated, and flat-topped. They are nearly always excoriated and often coalesce to form plaque. [12] [13] [67] [80] [84]



Figure 6.24: Hand eczema [79].

The hands, including the wrists, are frequently affected in adults and hand dermatitis is a common problem for adults with a history of AD. Hand eczema (Figure 6.24) is the most common occupational skin condition, accounting for more than 80 % of all occupational dermatitis. Women have an increased risk for the development of hand eczema. Most of this increased risk is accounted for by a spike in the rate of hand eczema at the age of 20-29, because of increased environmental exposure. [12] [13] [67] [80] [84]

There are five different types of hand eczema [79]: allergic contact dermatitis, irritant hand dermatitis, atopic hand eczema, vesicular endogenous hand eczema, hyperkeratotic endogenous hand eczema.

6.2.2.2 Atopic Dermatitis-affected Fingerprint Analysis

As in the case of wart-affected fingerprints, in order to study the differences between the images acquired by different sensors, three fingerprint images of the same finger have been selected (see Figure 6.25). Figure 6.25a has been acquired using the dactyloscopic card, to capture Figure 6.25b, the Sagem MSO 300 sensor has been used, and in the case of Figure 6.25c, the UPEK Eikon II sensor was used.

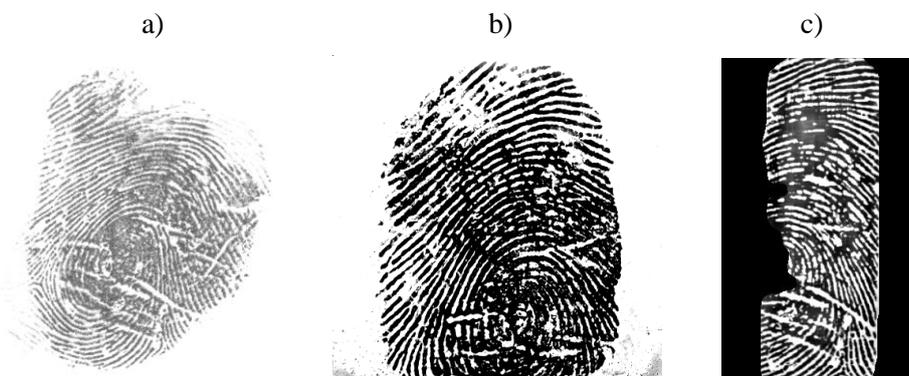


Figure 6.25: Same fingerprint affected by atopic dermatitis acquired by different sensors – a) dactyloscopic card, b) Sagem MSO 300 and c) UPEK Eikon II.

The comparison of the three images shows no significant difference in the quality among them. Abnormal white lines can be seen on all three of them, as well as patches of light and dark colours. Light patches are located mainly on the outer parts of the fingerprint, while dark areas are concentrated mostly in the centre of the fingerprint. [67] [84]

The four different fingerprint images affected by atopic dermatitis (see Figure 6.26) that have been acquired using a single sensor Sagem MSO 300 are compared. The first image in Figure 6.26 shows a clear, wide, and long white lines running throughout the whole fingerprint. The lines are mostly horizontal. The finger is dry and the ridge structure is, in some areas of the fingerprint image, less visible than in an image of a healthy finger. On the other hand, other parts of the fingerprint show unusually dark areas with a damaged ridge structure. Figure 6.26b is similar in the structure of the abnormal white lines to the previously described one. The lines run predominantly in a horizontal direction with their length as large as the width of the fingerprint. Other thinner and shorter white lines can be seen in both horizontal and vertical directions. The ridge structure is clearer than that of the previous fingerprint image; however, the patches are present there as well. The fingerprint in Figure 6.26c contains many large white-only patches with no ridge structure whatsoever. In the centre of the image is a wide line running from the bottom-left corner through the centre of the fingerprint to the upper-right corner of the image. Other thinner lines can be seen in the left half of the fingerprint. As the ridge structure is mostly badly damaged, this fingerprint image can be hardly used in an authentication system. Figure 6.26d is the last fingerprint of the described set. It is similar to the first analysed fingerprint image with white lines running throughout it. In contrast with the other images, this one's lines are not as wide and run only in a horizontal direction. The white lines are also considerably shorter. Several small white patches covering the ridge structure can be seen in the upper part of the fingerprint image. [67] [84]

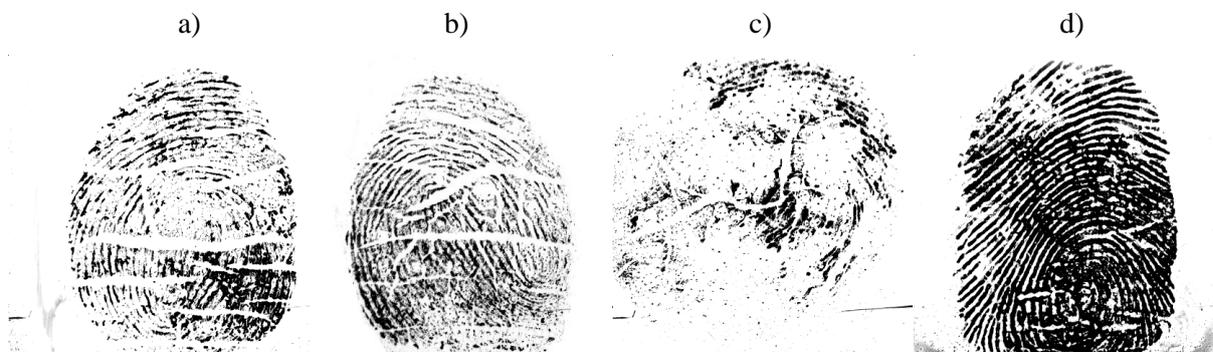


Figure 6.26: Different fingerprints affected by atopic dermatitis acquired by Sagem MSO 300.

The two types of damage from atopic dermatitis are abnormal white lines and light and dark patches. According to [88], the patches represent dystrophy of the skin and the median percentage of the surface area of dystrophy in their study was 22.80 %. The abnormal white lines usually run in a horizontal or vertical direction and their length ranges from very short to lines that run throughout the whole fingerprint. Once again referring to the study [88], the median number of white lines per fingerprint was 12 and the short horizontal lines prevailed (with occurrence in 73.0 %), followed by short vertical lines (56.5 %), long horizontal lines (52.5 %), and long vertical lines (18.0 %). [67] [84] [88]

6.2.2.3 Design of a Method for Atopic Dermatitis Simulation

Based on the analysis of existing fingerprints affected by atopic eczema, the design of a method for generating similarly damaged synthetic fingerprint images is proposed in this section. The algorithm consists of the following steps: [67]

1. Localise the fingerprint area on the image.
2. Create an empty image buffer.
3. Draw eczema patches into a buffer.
 - a. Determine the centre and size of the patch.
 - b. Draw the patch of the determined type (light or dark).
4. Determine the final colour of each pixel of the patches.
5. Blur the patches in the image buffer.
6. Draw eczema white lines into the buffer.
 - a. Determine the starting point, direction, and length of the line.
 - b. Generate line points in the given direction and length.
 - c. Interpolate the generated line points.
 - d. Draw the lines in the determined thickness.
7. Blur the lines in the image buffer.
8. Draw the buffer into the fingerprint image.

Step 1 of the algorithm is identical to the first step of the algorithm for generating warts. The result of this step is a contour of the fingerprint in the input image. Knowing precisely where on the image the fingerprint is located is necessary in order to only draw into the fingerprint area and not outside of it. In *step 2*, a new image buffer is created. The size of the buffer is the same as the size of the input image. The patches and white lines shall be drawn into the buffer separately as not to interfere with the original image. First, the light and dark patches are drawn into the buffer in *step 3*. The number of patches is generated randomly within the set boundary values. Afterwards, the type (light or dark), size, and the centre point for each patch is determined in *step 3a*. If the centre point lies within the fingerprint boundaries, the algorithm proceeds to draw the patch into the buffer. This is *step 3b* (see

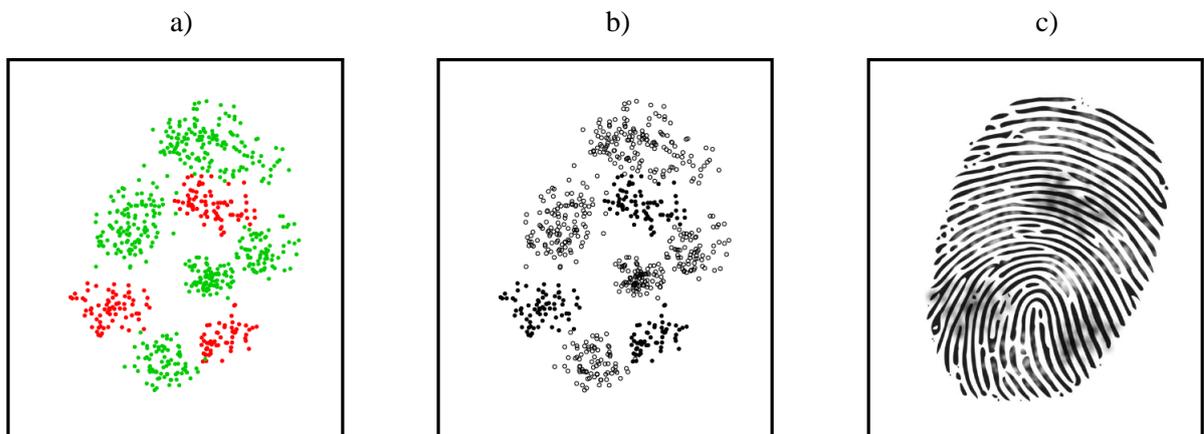


Figure 6.27: Eczema patches drawing – a) patches drawn in distinctive colours, b) patches drawn in final colours, c) blurred patches drawn into the fingerprint. [67]

Figure 6.27a). In this step, pixels are generated with an exponential distribution from the centre point and they are drawn into the buffer in a distinctive colour (e.g. red or blue). It is randomly chosen if the pixels of the patch will later be in a light or dark colour. [67] [84]

When all of the patches are drawn into the buffer in a distinctive colour, the final colour of each pixel is determined in *step 4* (see Figure 6.27b). First, the neighbouring pixels of each pixel in the patch are collected from the input image. Then, based on the selected algorithm, the final pixel's colour is either one of a randomly chosen neighbouring pixel or the mean of all of its neighbour's colours. After this, the patches in the buffer are blurred (with kernel of size 3×3) in *step 5* (see Figure 6.27c). [67] [84]

The second significant part of the algorithm takes place in *step 6* where white lines are drawn into the buffer. Each part of the process is described in the following paragraphs. In *step 6a*, the parameters of each line are determined. The length of the line is determined within the set boundary values and the line direction (either vertical or horizontal) is set. The starting point for line generation is found using random coordinate generation. The starting point must be sufficiently far from all other starting points of all other lines of the same type. Line points are generated in *step 6b*. Beginning with the starting point, other leading points are generated based on the length of the line, the direction of the line, and a random generated angle within a pre-defined range (see Figure 6.28a). To make the lines look more realistic, in *step 6c*, the line leading point count is doubled and spline interpolation of the first order is applied. This makes the line appear less edgy and smoother (see Figure 6.27b). [67] [84]

Finally, each line is drawn into the buffer in *step 6d* (see Figure 6.27c). The thickness of the line is set and the line is drawn in several steps, starting with the whole length drawn in the smallest thickness. Then the first and last leading points are removed and the line is drawn over with a higher thickness. This process repeats until the final set thickness has been reached. This ensures that the line's width decreases towards the line's ends. In the last two steps, the buffer is once again blurred (same kernel of size 3×3) in *step 7* and then in the following *step 8*, the buffer is drawn into the original fingerprint image, taking in consideration the transparency of the pixels in the buffer and blending them into the original image appropriately. [67] [84]

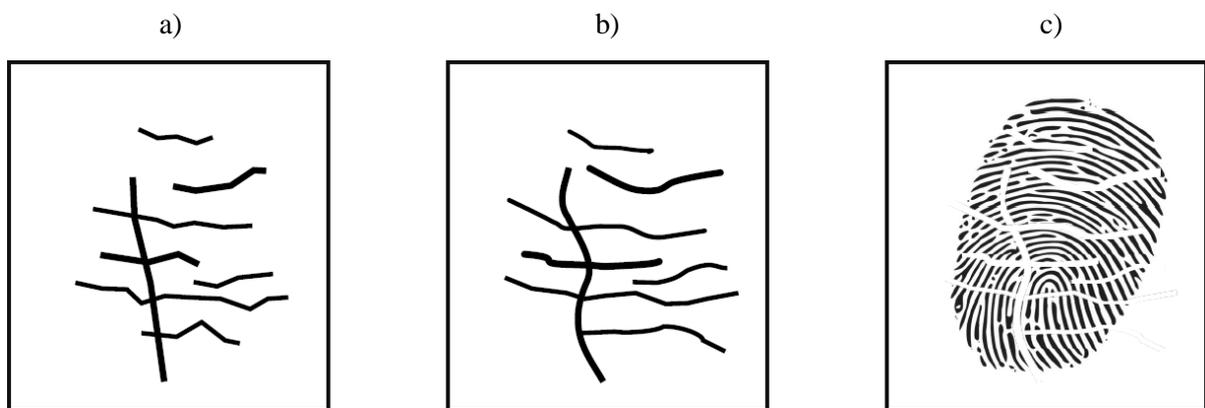


Figure 6.28: Eczema lines drawing – a) line leading points, b) interpolated line leading points, c) white lines drawn into the fingerprint. [67]

6.2.2.4 Examples of Atopic Dermatitis Damage

Nine different settings of atopic dermatitis have been simulated. The list of them contains damage name, testing shortcuts, and a short description with settings. The settings for atopic dermatitis contain the minimal and maximal number of horizontal lines (HL), minimal and maximal the number of vertical lines (VL), and the minimal and maximal length of lines (LL) – in percentage of the fingerprint radius. The possible values for the minimal LL are 0-200 and the possible values for the maximal LL are from 0 to infinity. Angles that influence both horizontal line torsion (HLT) and vertical line torsion (VLT) are from 0 to 180°, maximal line thickness (LT), minimal and maximal number of colour patches (CP), and minimal and maximal size of colour patches (SCP) in range of 0-100. For simplicity, LL is set to 50-200, HLT to 20, VLT to 30, SCP to 20-80, and the neighbour colour method is set to random. Real images of atopic dermatitis can be seen in Figure 6.25 and Figure 6.26, generated impressions in Figure 6.29, 6.30 and 6.31.

- **Few horizontal lines only (eczem0):** The synthetic image can be seen in Figure 6.29a. Settings: HL 4-8, VL 0-0, LT 8, CP 0-0.
- **A lot of horizontal lines only (eczem1):** The synthetic image can be seen in Figure 6.29b. Settings: HL 8-12, VL 0-0, LT 8, CP 0-0.
- **Few vertical lines only (eczem2):** The synthetic image can be seen in Figure 6.29c. Settings: HL 0-0, VL 2-5, LT 8, CP 0-0.
- **A lot of vertical lines only (eczem3):** The synthetic image can be seen in Figure 6.30a. Settings: HL 0-0, VL 5-8, LT 8, CP 0-0.
- **Horizontal and vertical lines only (eczem4):** The synthetic image can be seen in Figure 6.30b. Settings: HL 4-12, VL 2-8, LT 8, CP 0-0.
- **Horizontal and vertical lines, half thickness (eczem5):** The synthetic image can be seen in Figure 6.30c. Settings: HL 4-12, VL 2-8, LT 4, CP 0-0.
- **Small number of patches only (eczem6):** The synthetic image can be seen in Figure 6.31a. Settings: HL 0-0, VL 0-0, LT 8, CP 2-11.

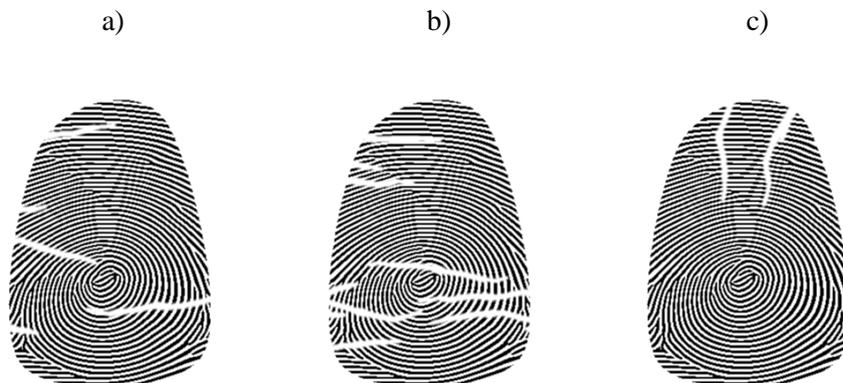


Figure 6.29: Examples of atopic dermatitis disease damage (a) *eczem0*, b) *eczem1*, c) *eczem2*.

- **High number of patches only (eczem7):** The synthetic image can be seen in Figure 6.31b. Settings: HL 0-0, VL 0-0, LT 8, CP 11-20.
- **All factors together (eczem8):** The synthetic image can be seen in Figure 6.31c. Settings: HL 4-12, VL 2-8, LT 8, CP 2-20.

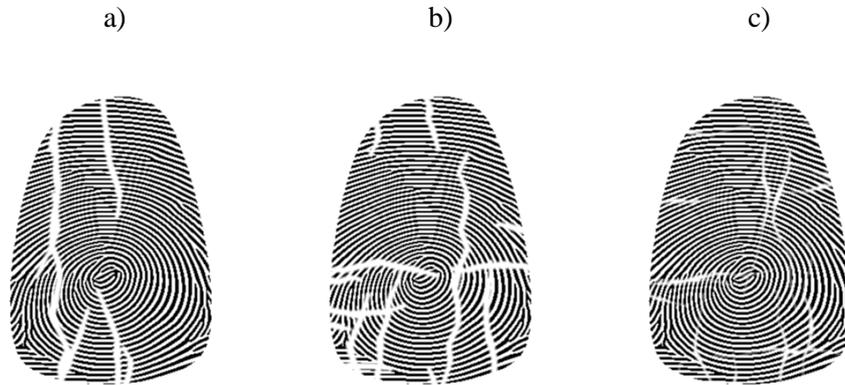


Figure 6.30: Examples of atopic dermatitis disease damage (a) *eczem3*, b) *eczem4*, c) *eczem5*.



Figure 6.31: Examples of atopic dermatitis disease damage (a) *eczem6*, b) *eczem7*, c) *eczem8*.

6.3 Simulation Based on Learning from Diseased Images

As stated in Subchapter 6.2, some diseases are hard to describe and have very different effects based on disease severity. It would be obvious to use some neural network or machine-learning methods for their simulation. On the other hand, as stated in Subchapter 6.1, it is really difficult to get diseased fingerprint images. The database used is not big enough to utilize these kinds of methods (e.g. deep neural networks). Nevertheless, the core idea of these methods could still be used.

6.3.1 Psoriasis

To test this concept, psoriasis was chosen as a sample disease. It is the second most frequent disease in the database. Also, it is one of the most frequent skin diseases. Around 2-3 % of the population suffers from this disease. Psoriasis is caused by a failure of the immune system. It is too active, so skin cells are created not in 28-30 days, but in three to four days. The body is not prepared for such an influx of cells, so the old cells are accumulated on the skin's surface as a result. Itchy, silver flakes known as plaque are created. The more severe the disease is, the more plaque is created and the more the fingerprint is damaged. [12] [13] [80] [81] [89]

Unlike other disease simulations, there is no need of a deep description of individual damages done to a fingerprint. Anyway, in Figure 6.32 the images acquired by the dactyloscopic card are shown. They are ordered by damage severity. In Figure 6.32a, only a small white part in the bottom left of the image is damaged. Figure 6.32b shows more plaque and more damage. Not only white places are present but also white lines. Inside of some white subjects are black dots. In Figure 6.32c, a very damaged fingerprint can be seen. Half of the image is white with black dots and the other half is almost entirely black. There are some remainders of the ridges at the bottom of the image. The last image, Figure 6.32d, shows only black noise with some white spaces where the ridges originally were.

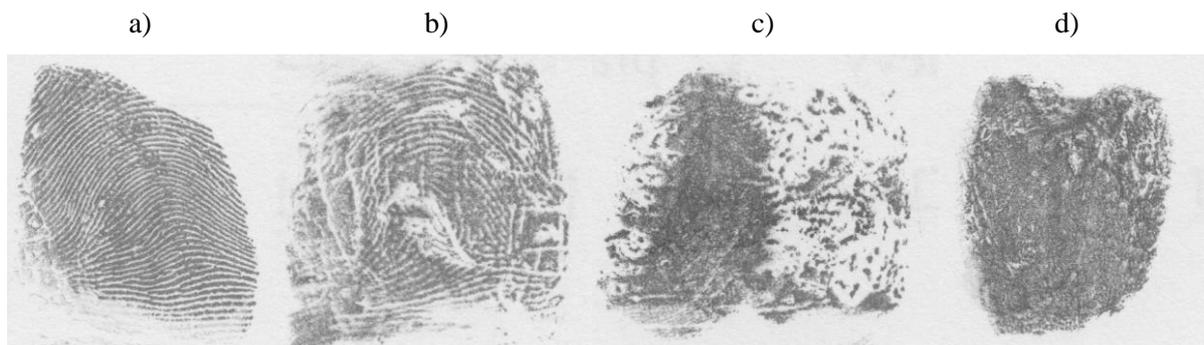


Figure 6.32: Different fingerprints affected by psoriasis acquired by dactyloscopic card.

6.3.1.1 Design of a Method for Psoriasis Simulation

Based on the described details of the disease, the idea of an algorithm for the damage extraction from existing images can be designed. This consists of the following steps:

1. Load an input (real) image.
2. Detection, extraction, processing, and storage of subjects from the image.
3. Repeat steps 1-3 until there are no input images.
4. Load the synthetic image.
5. Localize the fingerprint area.
6. Load the damage subject.
7. Insert subject into the image.
8. Until there is a defined number of subjects in the image, repeat step 6-8.

The idea is to extract subjects (individual damages) from real fingerprint images. That is *step 1*; it loads all images that can be learned from one by one. The image gets a five-pixel border and the

background is filled with one colour. This image is now ready for *step 2*; the extraction of subjects. This is done by using threshold, colour conversion, blur, erosion, dilatation, and a Canny operator. The three main classes of a subject are defined. *Black subjects* (size 5,000-34,000 px), *small white subjects* (size 370-6,000 px), and *large white subjects* (size 6,000-25,000 px). The size values of different subjects were found out through experimentation. Because the background is filtered, it is relatively easy to take bigger areas in the image as a source of damage. Examples of extracted subjects can be seen in Figure 6.33. [89]

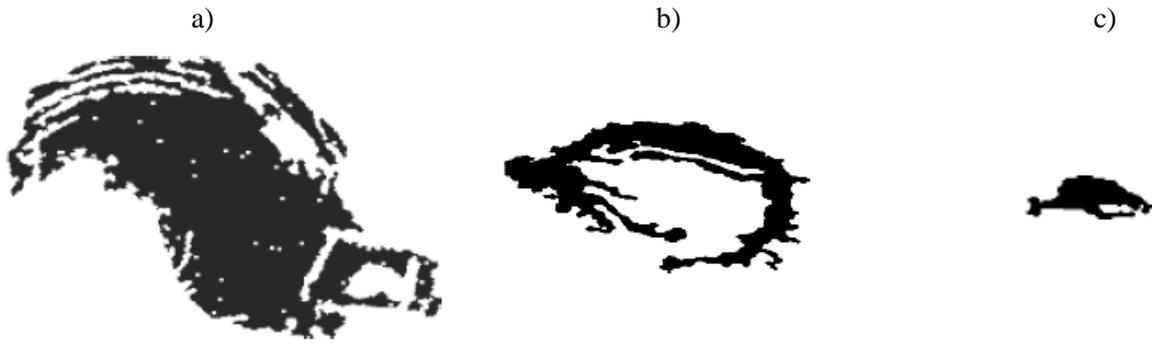


Figure 6.33: Enlarged and coloured subjects – a) black subject, b) large white subject, c) small white subject.

Subjects have their alpha channel set, otherwise their white background colour would be used instead of transparency. Because of the possible various image sizes in the input database, all subjects have to be normalized. That is done by using Eq. 6.2. [89]

$$\begin{aligned} subject_{height} &= \frac{target_image_{height}}{input_image_{height}} \\ subject_{width} &= \frac{target_image_{width}}{input_image_{width}} \end{aligned} \quad (\text{Eq. 6.2})$$

Subjects are stored in the database. In *step 4*, the target (synthetic image) is loaded. By mapping the non-zero pixels, the area of the fingerprint is localized in *step 5*. In *step 6*, the subject from the database is loaded. Based on the defined number of subjects that should be inserted into the image, the count of each subject class is determined. For the *black subject* the rules are: [89]

- Defined number ≤ 15 – no subject from this class,
- defined number ≤ 30 – 1 subject from this class,
- defined number ≤ 40 – 2 subjects from this class,
- defined number ≤ 80 – 3 subjects from this class,
- defined number ≤ 150 – 4 subjects from this class,
- defined number ≤ 250 – 5 subjects from this class,
- defined number > 250 – 6 subjects from this class.

For the *larger white subjects*, the rules are:

- Defined number ≤ 15 – no subject from this class,
- defined number ≤ 100 – 1 subject from this class,
- defined number ≤ 250 – 2 subjects from this class,
- defined number > 250 – 3 subjects from this class.

For the *small white subjects*, the defined number is the count of subjects from that class. Also, this small object is inserted into a random location, but it has to be ensured that the whole subject would be in the fingerprint area. Individual damages could be overlapping. For the black and larger white objects, the following optimization is used. There is a 50 % chance of using a mirror image of the stored subjects (practically doubling the number of subjects available). These subjects are inserted closer to the centre of the image. Random coordinates are generated from the interval of 1-90 px for vertical axis and 1-60 px for horizontal axis. Obviously, the whole subject has to fit into the fingerprint area. [89]

For the simpler processing of input images, only one of the acquirement methods was chosen. The chosen method is the dactyloscopic card (as can be seen on Figure 6.32). This is because more than half of the psoriasis images in the database are from this method. Overall, 174 images of psoriasis from the dactyloscopic card were used to extract subjects. From this input, 122 black subjects, eight large white subjects, and 1,022 small white subjects were extracted.

6.3.1.2 Examples of Psoriasis Damage

In this case, six different settings of psoriasis are simulated. The list contains damage name, testing shortcuts, and a short description with settings. The settings for this type of simulation is only the number of subjects inserted into the image. The real images of psoriasis can be seen in Figure 6.32, generated impressions in Figure 6.34.

- **Small number of subjects (psor0):** The synthetic image can be seen in Figure 6.34a. Settings: 15 subjects.
- **Small number of subjects plus one black subject (psor1):** The synthetic image can be seen in Figure 6.34b. Settings: 30 subjects.
- **Small number of subjects plus two black subjects (psor2):** The synthetic image can be seen in Figure 6.34c. Settings: 40 subjects.
- **Moderate number of subjects plus three black subjects (psor3):** The synthetic image can be seen in Figure 6.34d. Settings: 60 subjects.
- **High number of subjects plus three black subjects (psor4):** The synthetic image can be seen in Figure 6.34e. Settings: 80 subjects.
- **Enormous number of subjects plus four black and one larger white subject (psor5):** The synthetic image can be seen in Figure 6.34f. Settings: 100 subjects.

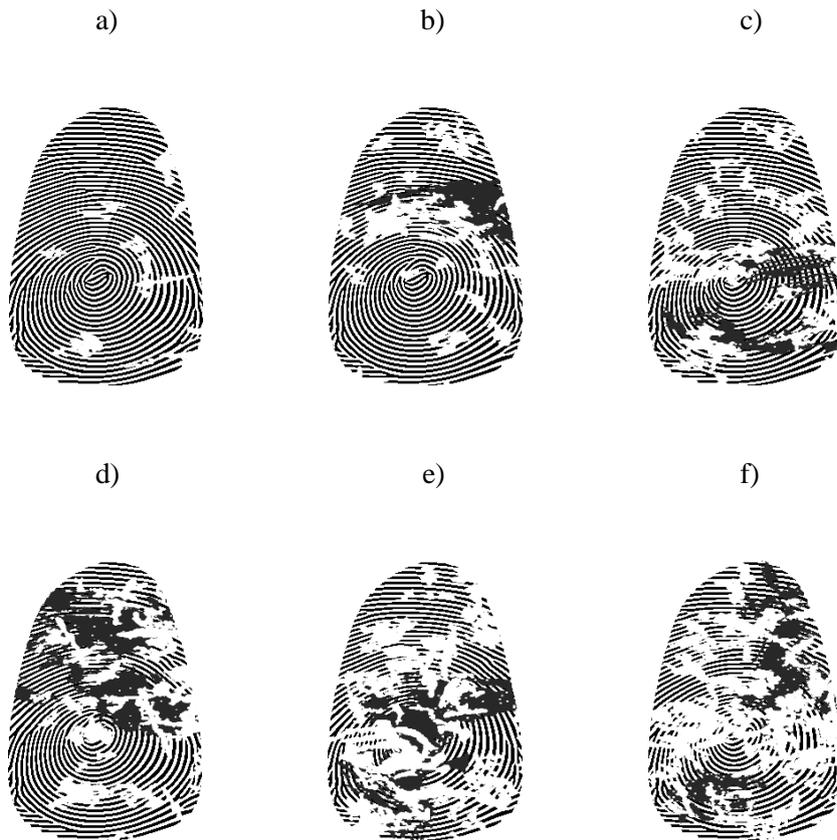


Figure 6.34: Examples of psoriasis disease damage (a) *psor0*, b) *psor1*, c) *psor2*, d) *psor3*, e) *psor4*, f) *psor5*).

6.4 Evaluation

Introduction to the evaluation process is the same as it was in Subchapter 5.5, respectively Subchapters 5.5.1, 5.5.2, and 5.5.3. Because of that, this text is just a summary of the most important information; for a more detailed description, please look at the aforementioned subchapters. The only difference is that for the evaluation of the diseases, the range of some graphs have to be expanded upon. If a direct comparison of the graphs is needed, bear this difference of ranges in mind. Also, only basic damages are evaluated.

The database of 150 synthetic images from three synthetic generators were used. Each generator created images with natural fingerprint class distribution. Images were scaled so that their resolution is similar. Four quality measurement metrics are used for the evaluation of the generated images. The standardized NFIQ (NIST – National Institute of Standards and Technology, Fingerprint Image Quality) solution, where quality is determined by five classes, where first class is the best quality and fifth being the worst. There is the commercial VeriFinger software with its quality measurement and

verification comparison score. Quality here is in the range of 0-100, where 100 is the best quality. The comparison score, where damaged fingerprint images and their respective source images are used for verification, is roughly from 0 to 2,250, where the higher number means higher quality (better verification to be exact). The last metric is quality measurement from Mr. Oravec, which is an experimental algorithm that is currently being researched. Its range is from 0-100, where 100 signifies the best quality.

6.4.1 Evaluation by the Dermatologist

The first part of the evaluation process can be found in the previous subchapters. That is a visual comparison of the damaged fingerprint images and source fingerprint images with skin diseases. To ensure that not only the images look similar to real one, but that they really resemble the image of the disease, they were also discussed with medical doctors. Methodologies and some example images were consulted with a dermatologist, namely with MUDr. Eva Březinová Ph.D. Bear in mind that dermatologists are used to working with the whole hand, finger, or body. There was, for example, a discussion that fingerprint (or fingertip) images are not enough to distinguish between psoriasis or eczema. There are a lot of accompanying factors that help medical doctors with their task of disease recognition and setting up the right treatment. A lot of diseases show their initial manifestation very similarly. Sometimes it was decided to directly simulate the later phases or more severe variants of the diseases. That was the case of atopic eczema. In the chronic phase it is typical to have cracked skin (thus creating lines on the fingerprint) and the darker or lighter spots on the fingerprint images are from the physical damage of diseased skin. Sometimes small white stains show, which are the manifestation of blisters. This description is specific, clear, and easily recognizable, unlike some blurry parts because of the malnutrition of diseased skin, which is a factor for all kinds of eczemas. To sum up, methodologies and results were consulted, and they were approved as similar to real diseases.

6.4.2 Warts Damage Evaluation

Information about the specifics of the damage and example images can be found in Subchapter 6.2.1.4. The focus here is to find out if the damaged fingerprints are worse than the reference images without the damage and what is the best damage (meaning it has the worst quality). An important fact is that the generation of warts have some randomized factors. It is possible to place the wart on different spots which could have an effect on the final score.

All the values for the warts in the Oravec quality score (in Figure 6.35) are lower than *no damage* images. Otherwise damages are sorted from left to right so that *warts4* (extreme) are the best. The value *warts4* is also significantly better than the others.

In previous evaluations, the NFIQ has proved to have a very low classification power. This graph (Figure 6.36) demonstrates why.

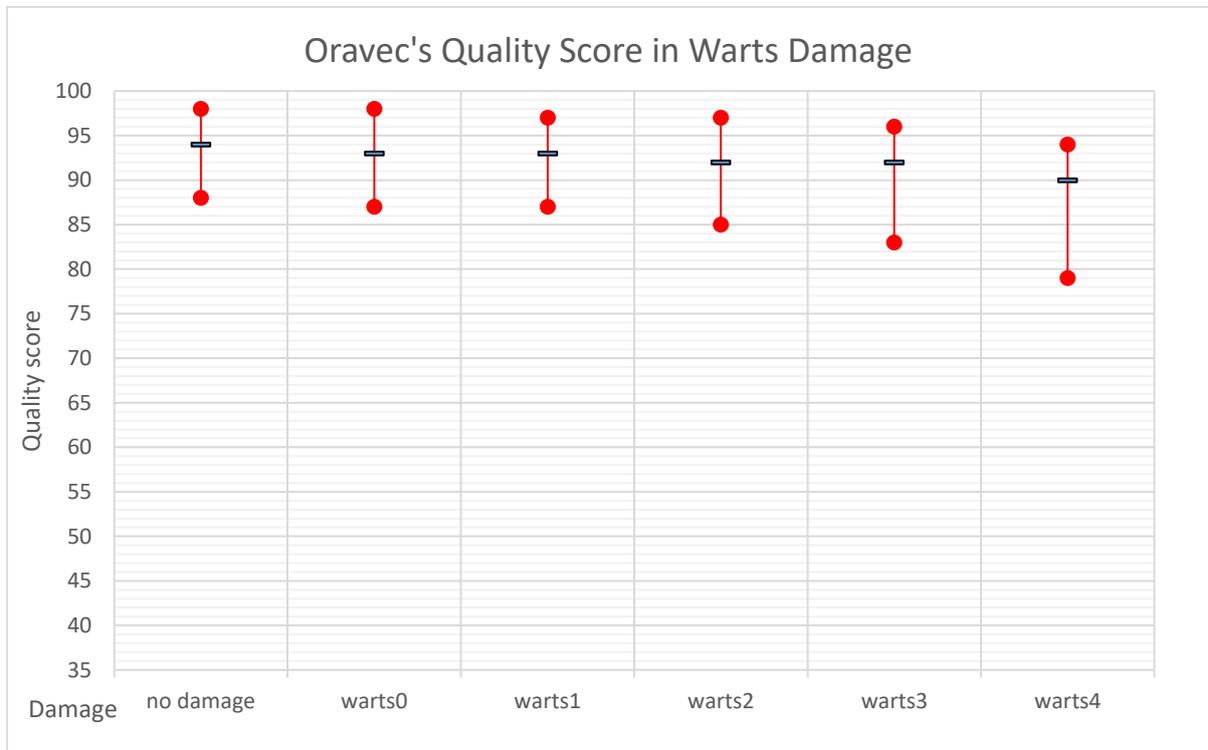


Figure 6.35: Graph of Oravec's quality score in warts damage.

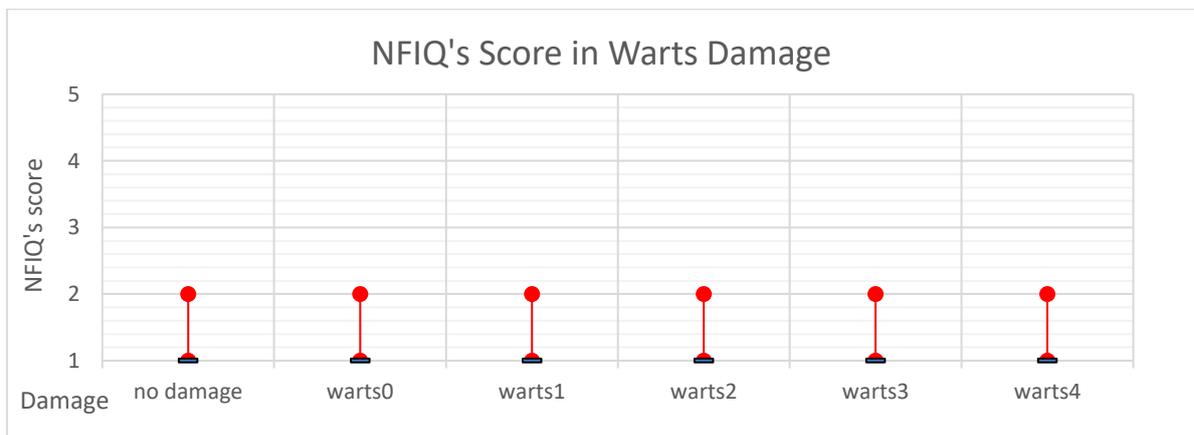


Figure 6.36: Graph of NFIQ's quality score in warts damage.

VeriFinger's quality score (in Figure 6.37) shows a very minor reduction of these values. Starting with *warts0* (small warts with no secondary), which is basically the same as the *no damage*, and ending with *warts4* (extreme), which exhibits a visible change in median values.

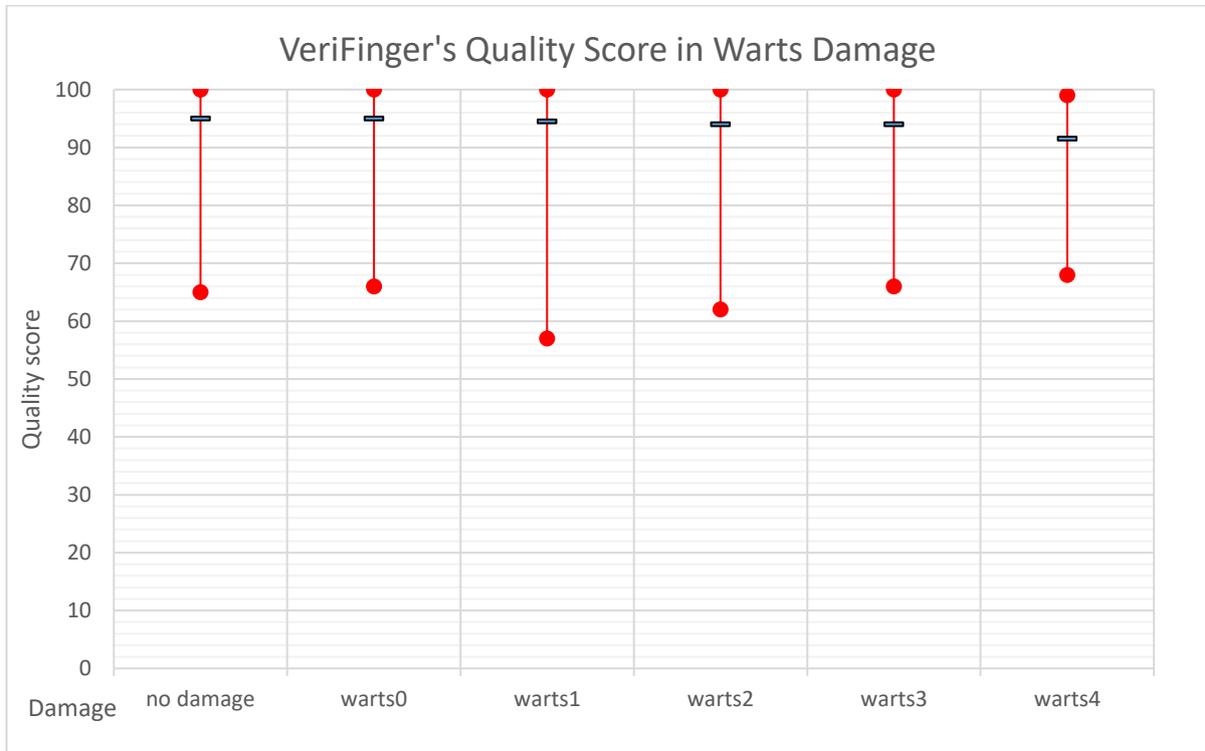


Figure 6.37: Graph of VeriFinger's quality score in warts damage.

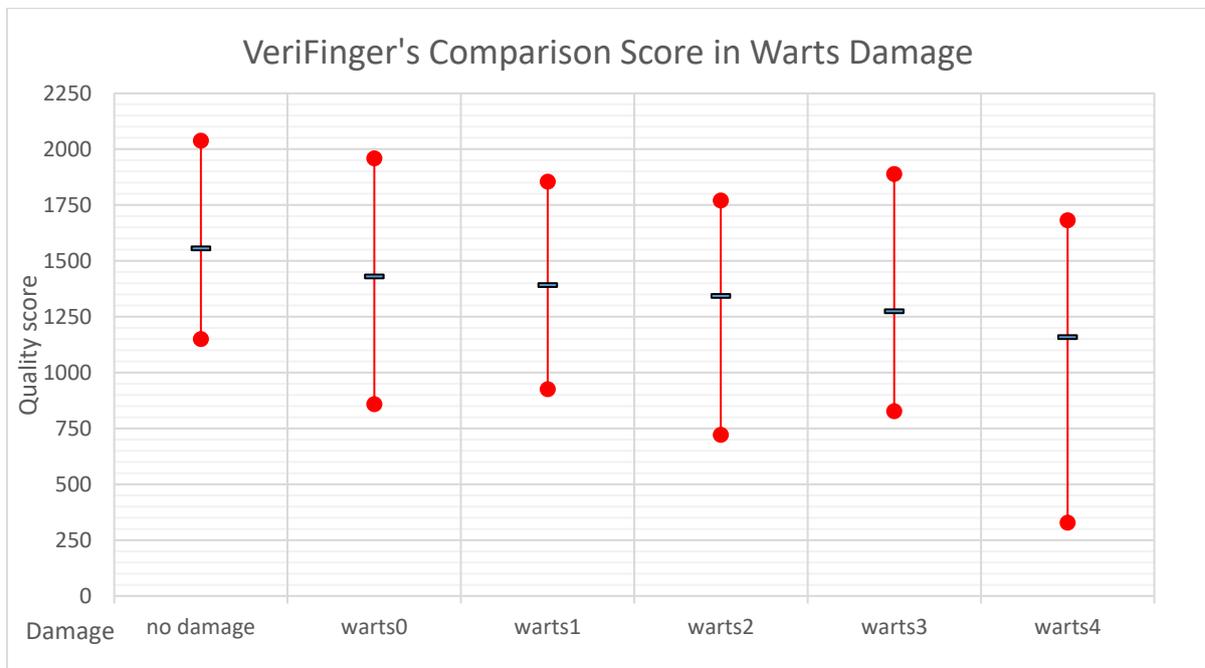


Figure 6.38: Graph of VeriFinger's comparison score in warts damage.

This graph (Figure 6.38) is a nice presentation of random effect evaluation. There is a stable reduction in median quality score based on the severity of the damage. However, the minimal and maximal values are scattered probably just because of the randomized parts of the algorithm. *Warts4* (extreme) confirmed the role of the best damage. In conclusion, all damages have the same or lower quality scores than the reference damage.

6.4.3 Atopic Eczema Damage Evaluation

The structure of the atopic eczema damage is a little bit different than the other damages. The last damage (*eczem8*) combines all previous damages. All this information and example images are in Subchapter 6.2.2.4 Algorithms that create these damages have some parts that are stochastic. This means that one damage type can have different effects on various fingerprints. For example, in one fingerprint it could randomly damage the core with a lot of minutiae (and cause an extremely low-quality score), but in another fingerprint it could randomly damage only a marginal portion (and cause an extremely high-quality score). Nine (or ten different damages if *no damage* is also counted) is too much for one graph to fit in, so there are two graphs for each metric.

The system of damage composition can be clearly seen in Figure 6.39 and Figure 6.40. First, the median reduction *eczem4* (horizontal and vertical lines only), as the name suggests, combines previously tested horizontal (*eczem0* and *eczem1*) and vertical (*eczem2* and *eczem3*) lines. Second, a high reduction in quality is to be expected in *eczem8* (all factors together) because it combines the previous damages. This also means that lines and spots are posing different damages to the images, so their quality reduction can be almost added together.

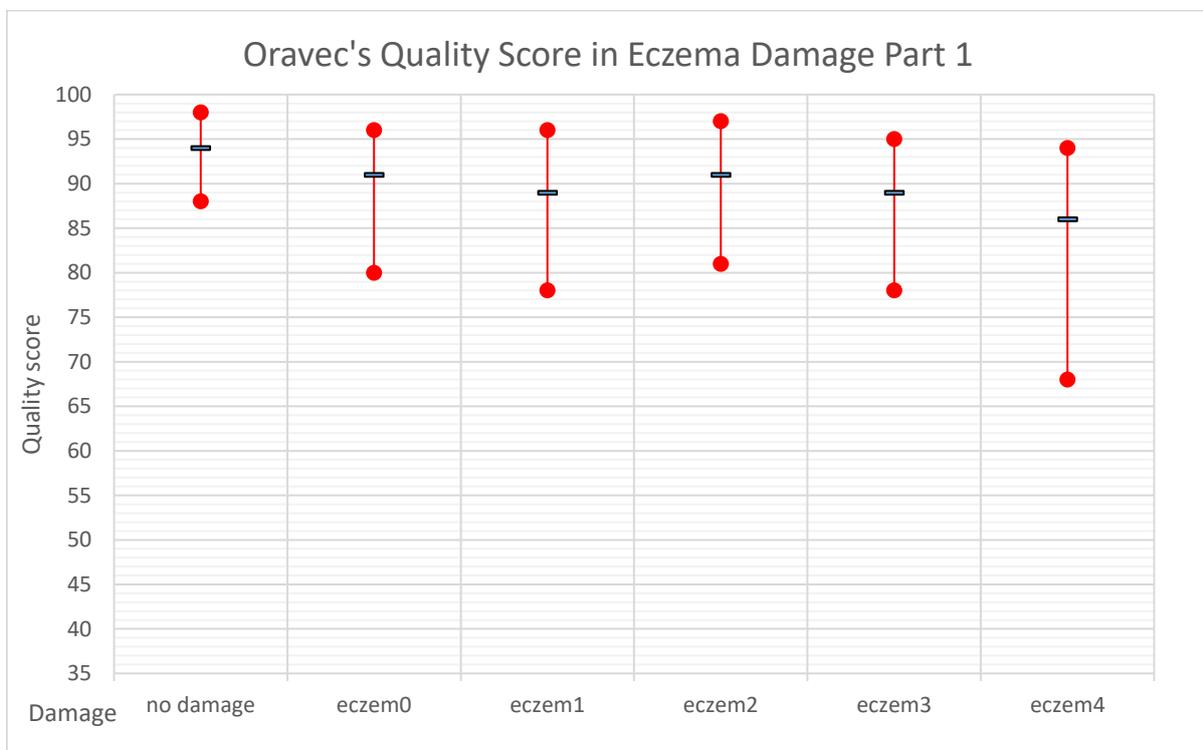


Figure 6.39: Graph of Oravec's quality score in atopic eczema damage – part 1.

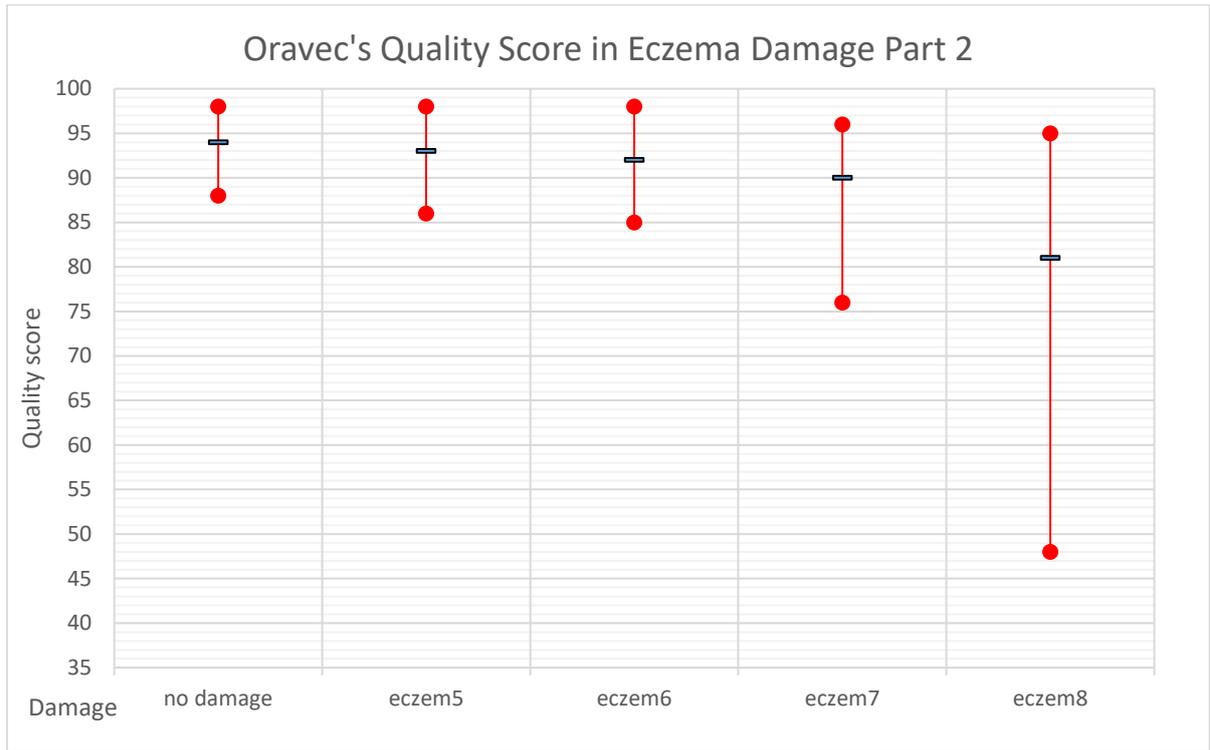


Figure 6.40: Graph of Oravec's quality score in atopic eczema damage – part 2.

NFIQ score graphs (in Figure 6.41 and Figure 6.42) are just showing that the *eczem8* (all factors together) is better than every other damage in this category. It was not enough to move the median score, just the maximal value.

VeriFinger's quality score (as can be seen in Figure 6.43 and Figure 6.44) exhibits the same damage structure as Oravec's quality metrics, but the differences are quite lower. *Eczem0* (few horizontal lines only), *eczem2* (few vertical lines only), and *eczem5* (horizontal and vertical lines, half thickness) essentially have the same values as the *no damage*. *Eczem8* (all factors combined) has lower quality than *no damage*, but it is not a big difference.

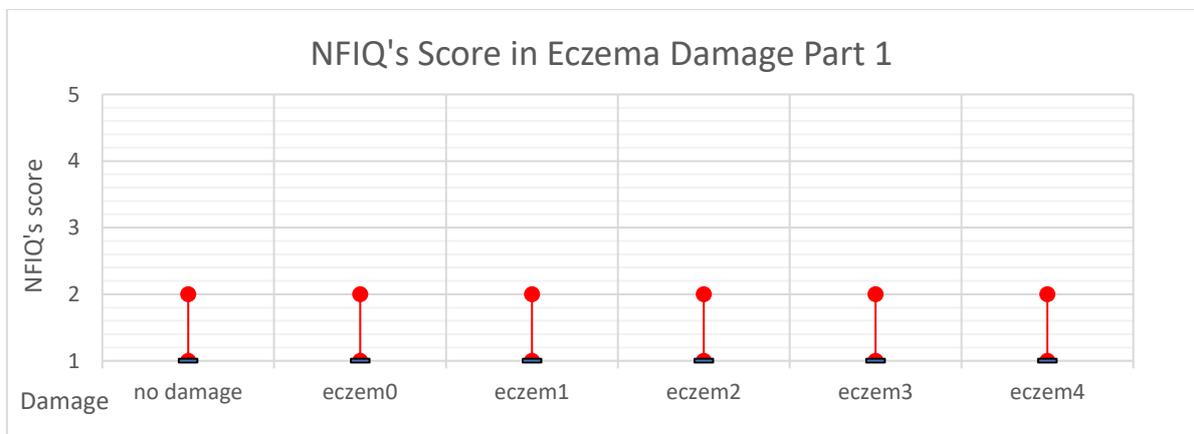


Figure 6.41: Graph of NFIQ's score in atopic eczema damage – part 1.

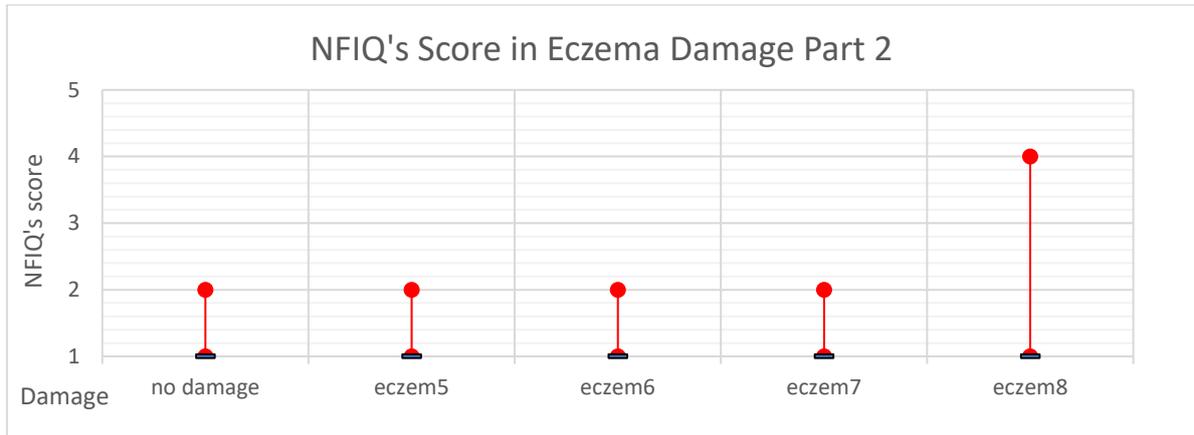


Figure 6.42: Graph of NFIQ's score in atopic eczema damage – part 2.

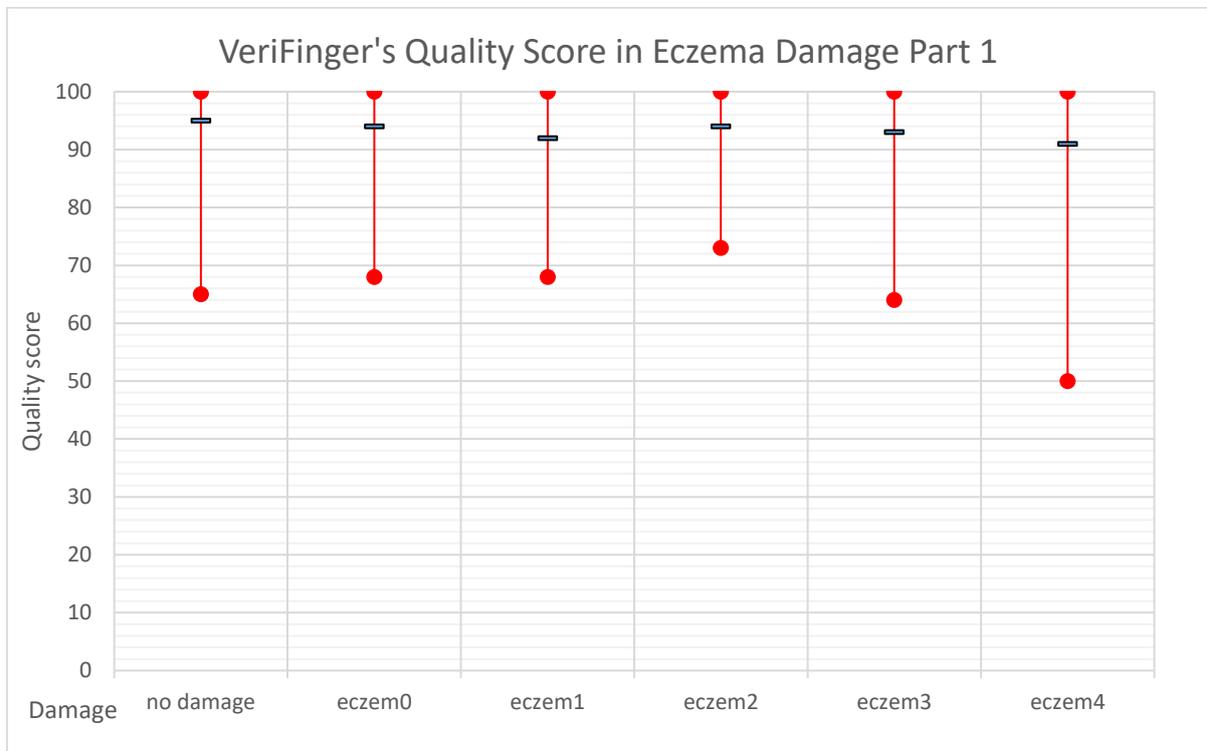


Figure 6.43: Graph of VeriFinger's quality score in atopic eczema damage – part 1.

The comparison score (Figure 6.45 and Figure 6.46) confirms the results from the Oravec quality score. It is a different scale, but still very similar percentage changes in the same values. The assumption is that these two measurements are acting very similarly to the damage done to the images. It is safe to say that damage is done to the images because the scores are lower than the reference. As was expected, the best damage is *eczem8* (all factors combined).

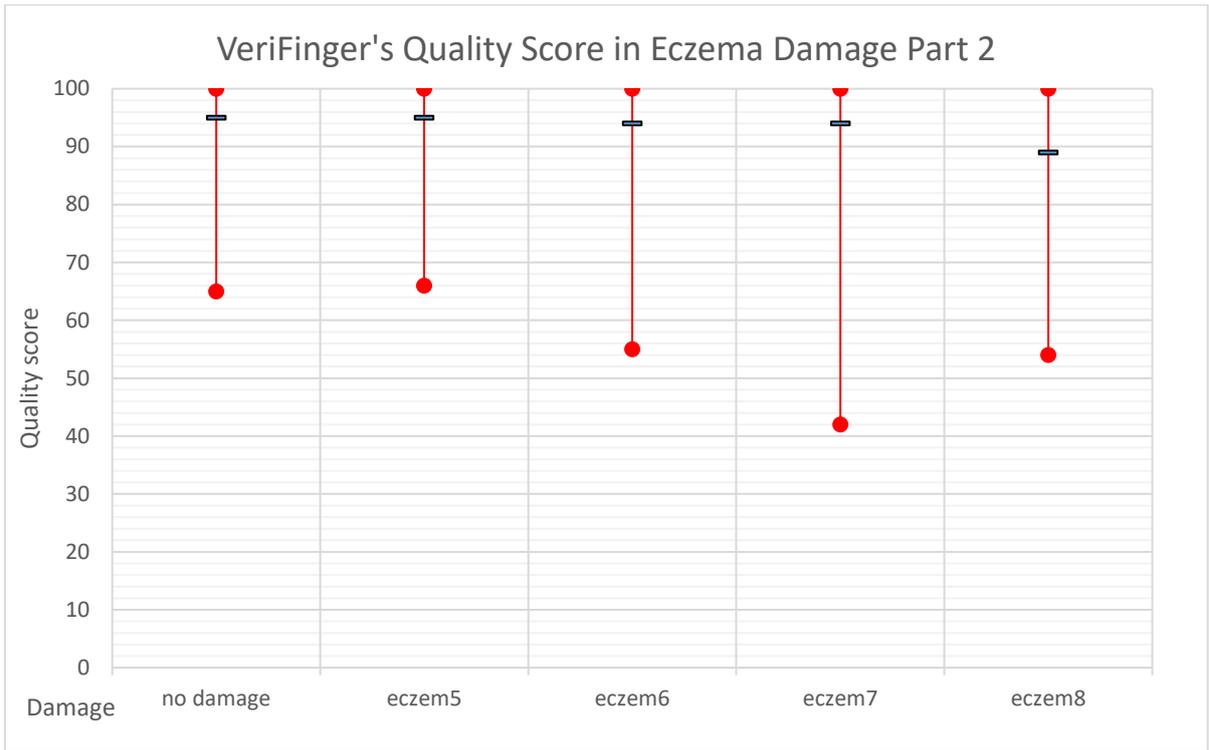


Figure 6.44: Graph of VeriFinger's quality score in atopic eczema damage – part 2.

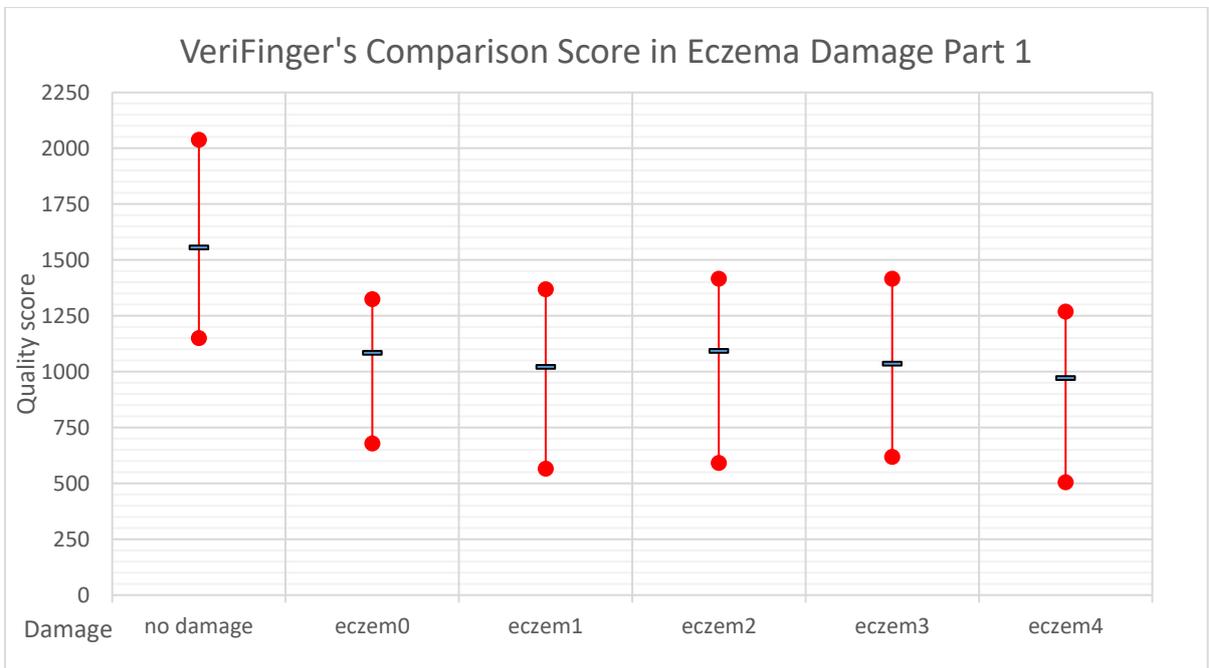


Figure 6.45: Graph of VeriFinger's comparison score in atopic eczema damage – part 1.

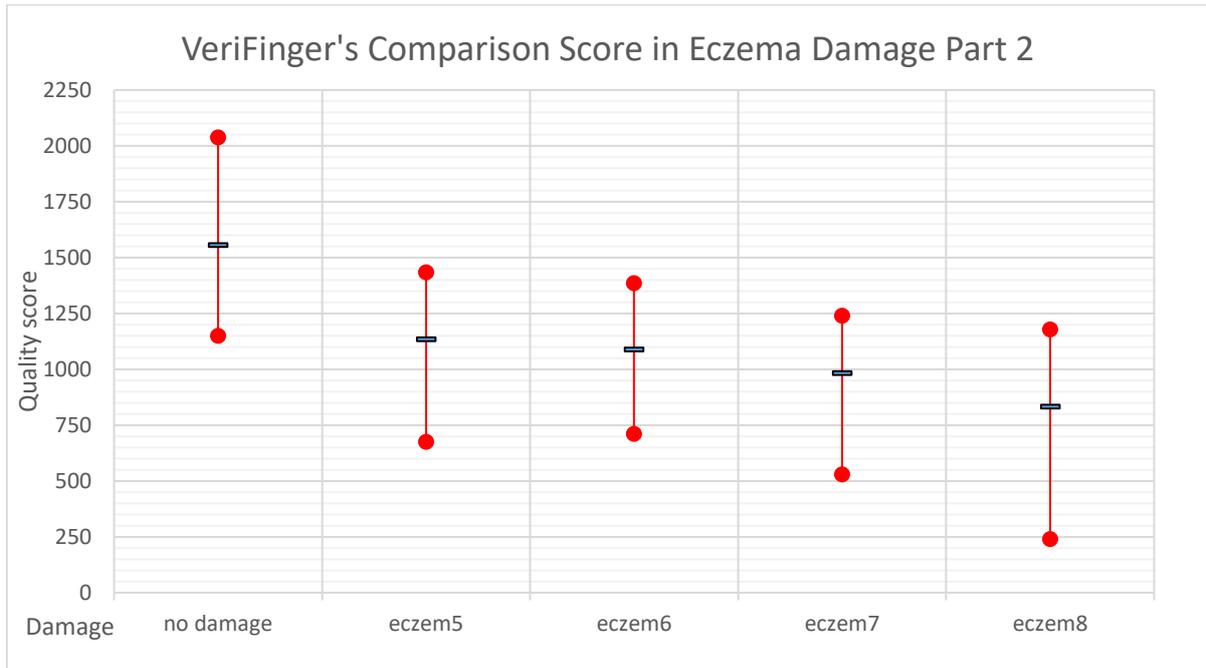


Figure 6.46: Graph of VeriFinger's comparison score in atopic eczema damage – part 2.

6.4.4 Psoriasis Damage Evaluation

Detailed information and example images for the psoriasis damage category are summarized in Subchapter 6.3.1.2. The severity of the damage should be sorted out (the higher number in the damage shortcut should mean more severe damage). The choice and position of generated subjects are stochastic, so again, a little bit of volatility in maximal and minimal values can be expected.

An almost linear dependence for the median quality can be seen in Figure 6.47. The only outlier value is *psor0* (small number of subjects), which is not low enough to fit into the pattern. What is impressive is the minimal value of *psor5* (enormous number of subjects plus four black and one larger white subject). This is just one of the reasons why the range of the graphs must be wider.

Even the NFIQ graph (Figure 6.48) shows some interesting results. The median values for almost all of the damages, *psor2* (small number of subjects plus two black subjects), *psor3* (moderate number of subjects plus three black subjects), *psor4* (high number of subjects plus three black subjects), and *psor5* (enormous number of subjects plus four black and one larger white subject) is at the fourth class. This is very interesting when considering the weak results in the previous subchapters. It is very likely that the NFIQ is trained to be very sensitive to this kind of damage. *Psor4* and *psor5* even achieved minimal values in the second class.

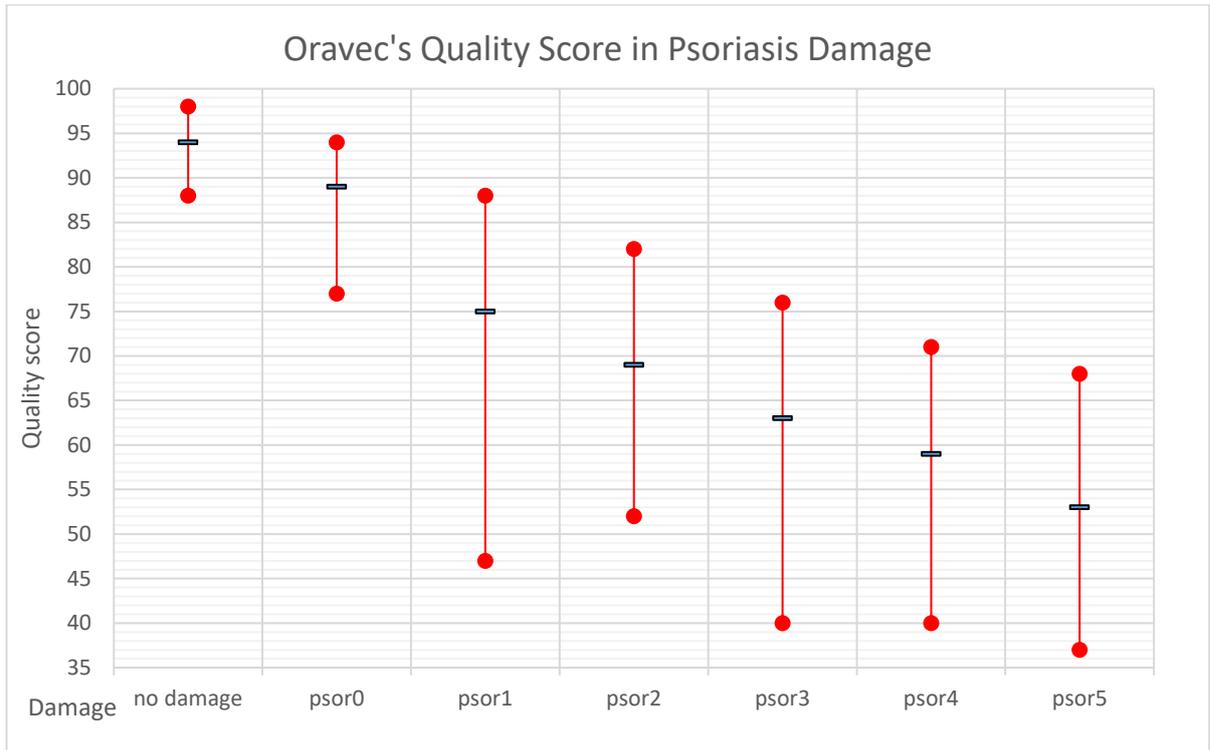


Figure 6.47: Graph of Oravec's quality score in psoriasis damage.

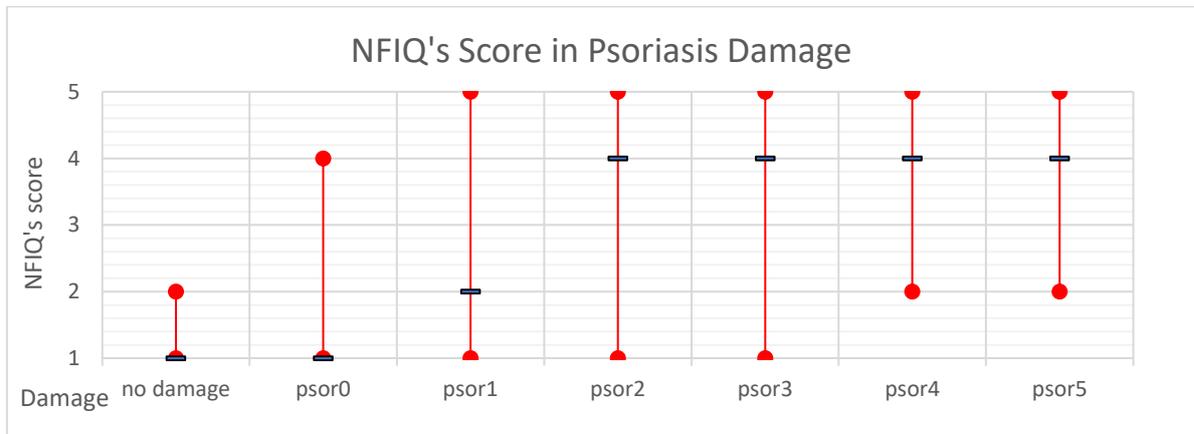


Figure 6.48: Graph of NFIQ's score in psoriasis damage.

VeriFinger's quality in Figure 6.49 shows the same behaviour as the Oravec quality metric. It can be said that now it is more linear than before. The same small changes in the minimal and maximal value also confirm the fact that they are created because of some random effects.

Lastly, Figure 6.50 shows a familiar looking graph. The median values have a more quadratic distribution. Nevertheless, the extreme minimal value for the *psor1* (small number of subjects plus one black subject) can be seen. By the similarity of the graphs, it can be said that all quality metrics are sensitive to this kind of damage and that this is why all of them show similar scores and behaviour.

As a final note, all damages are lower than the reference *no damage* values. In the comparison with the swipe mode, the *psor2* is on a par with the best swipe mode damage (normal width image “*pm1 dis0 narr4 dmg1*”). *Psor3*, *psor4*, and *psor5* being the most damaging variants.

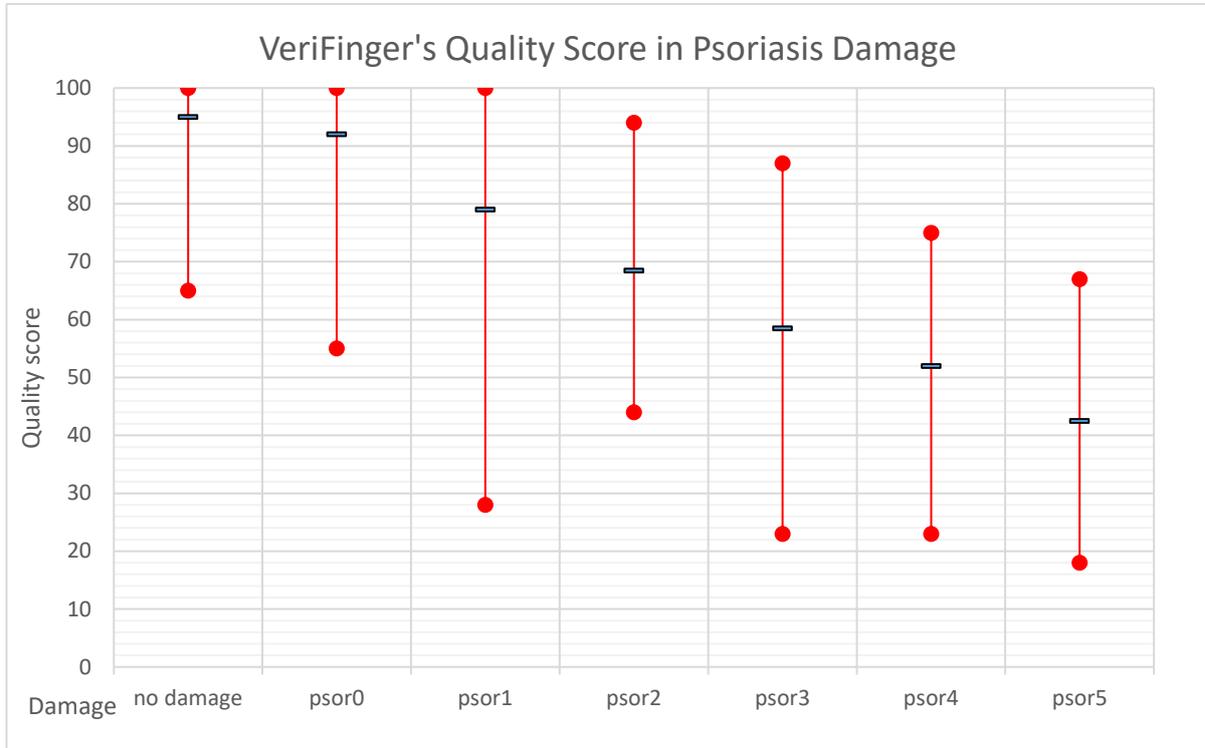


Figure 6.49: Graph of VeriFinger's quality score in psoriasis damage.

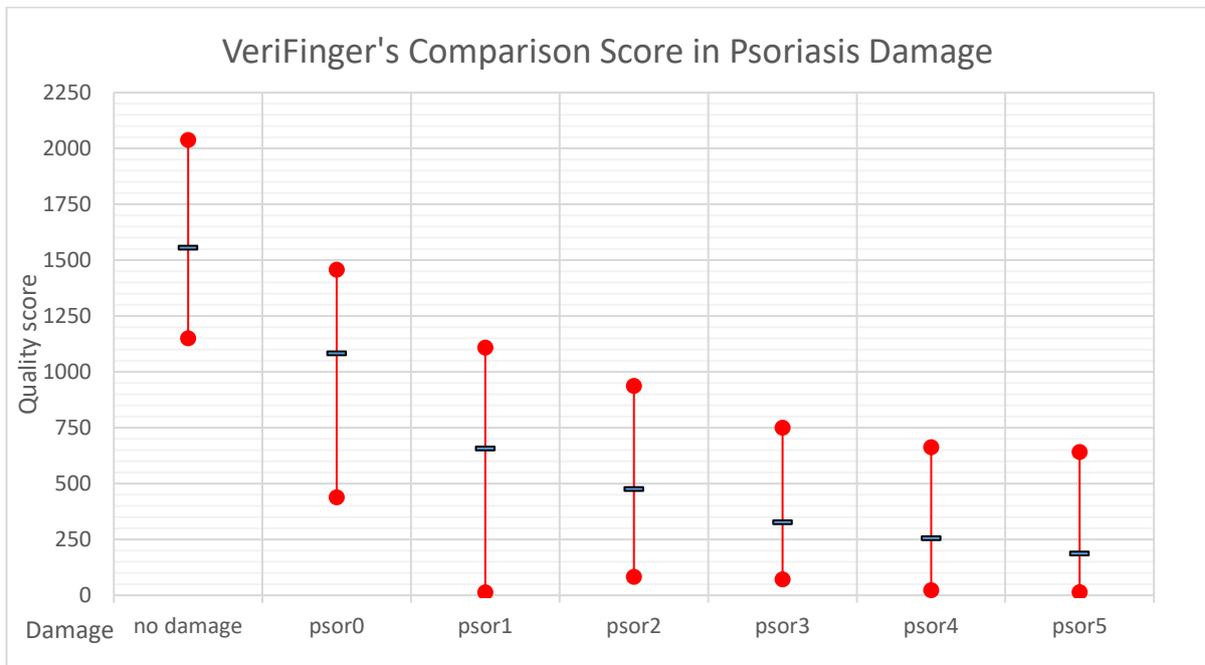


Figure 6.50: Graph of VeriFinger's comparison score in psoriasis damage.

7 Other Inspected Damages

Other researched areas are described in this chapter. Theoretically, any damage from any source can be simulated to the synthetic fingerprints. The simulation of a fingerprint spoof and the simulation of factors caused by the use of everyday detergents or lotions are described in this chapter. The exact influence on the fingerprint is determined and following that, a consideration of them as a possible extension of the application is described. In the scope of (other) specific damages, interesting work on altered fingerprints is shown in [90].

7.1 Fingerprint Spoofs

The simulations of factors that are specific for fingerprint spoofs can be useful in recognizing these spoofs directly from the fingerprint image they produce. First, however, a database of images to analyse the damage done to the fingerprint is required.

7.1.1 Spoof Production

Nowadays, it is well known that there are many various materials that can be used for production of fingerprint spoofs. The whole process of this production can be divided into several parts, according to input data that is available. These parts are shown in the overview in Figure 7.1. If the fingerprint mould is created with the consent of a genuine user, then it is cooperative method. This way the mould is created directly from user's finger. However, oftentimes there is not this luxury of cooperation and the mould has to be created indirectly by using other information. For example, a cut-off finger could be used – this method is popularized by the movies, but it still requires a bit of “cooperation” from the user. Other groups of non-cooperative methods presume access to the sensor, which could be spoofed – by either the reactivation of a previously used fingerprint or by spoofing the fingerprint synthesis sensor. Fingerprint reactivation is the activation of the residue remaining on the sensor by gently breathing on the sensor or using a wet sponge. Fingerprint synthesis gets a fingerprint image by reconstructing it from an enrolled template in the biometric solution database [91]. The last and most commonly used non-cooperative method uses the latent fingerprint left by a genuine user. There are several methods that reveal latent fingerprints. Fingerprint images could be acquired by using powder and brush to reveal it and Scotch tape to take it. Another very fast method is based on the use of an electrospun nanofiber mat. If the fingerprint image can be seen by the naked eye, then it could be photographed and the digital image can be obtained that way. [16] [52] [92] [93] [A2]

Fingerprint synthesis and latent fingerprints both end with an electronic format of the fingerprint image. These can be improved using graphical software and after that, the final step is to create mould. It could be used to either create a stamp with the fingerprint and use it like a direct mould (using wax,

Play-Doh, etc.) or a negative image could be printed onto the printed circuit board (PCB). In the following images and experiments, a PCB mould is used. [52] [92] [93] [94] [95] [96] [A2]

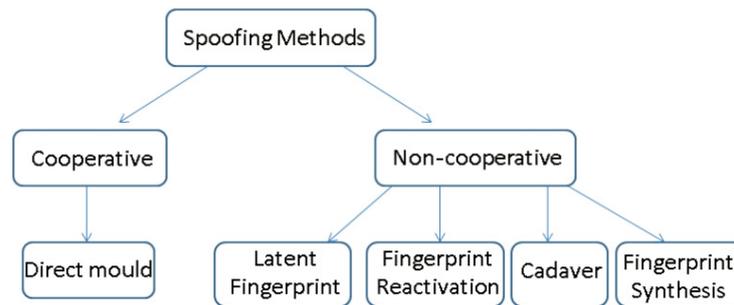


Figure 7.1: Overview of various spoofing methods. [92]

With the prepared mould (using source fingerprint image), the only missing thing is the material from which the spoof will be created. Materials can be divided into three groups:

- **Technical industry:** aquarium silicon (black and transparent), epoxy resin kit for Epoxy CHS 1200, epoxy resin “Havel Composite L285”, and epoxy resin “HobbyKing”.
- **Food industry:** gelatine, aspic, and gummy bears.
- **Creative materials:** Fimo standard, Fimo air, Kera, WePAM, Oyumare, Play-Doh, Premo, glass colours, Cernit, gel wax, Kato, Siligum, latex, and wax sheets.

When the material is in the required shape, i.e. it is dry, it supposedly perfectly fills the mould, then it is taken out and a fingerprint spoof is done. [52] [92] [93] [A2]

7.1.2 Spoof Images

From the various materials mentioned in Subchapter 7.1.1, images were acquired. In this subchapter, a short description of the materials and images from it are shown. All subchapters are based on [92] [93] [A2].

7.1.2.1 Technical Industry Materials

Aquarium silicon is used as the name suggests, to stick glass plates together. This material is soft after drying, flexible, and durable with clearly visible ridges, allowing the minutiae points to be precisely located. Images are shown in Figure 7.2ab.

Fingerprint spoofs made of **epoxy resin CHS1200** are very tough. Ridges are clearly visible with no air bubbles inside. The toughness of this material could cause some troubles due to the inflexibility and fragility of a spoof when using touch-based sensors. Images are shown in Figure 7.2cd.

The **epoxy resin Havel Composite L285** is a material that was originally designated for the aircraft industry and scale model building. This epoxy together with a hardener creates a very viscous mixture which does not contain air bubbles and fills the mould perfectly. Images are shown in Figure 7.2ef.

Spoofs from epoxy resin **HobbyKing** is tough and fragile. Images are shown in Figure 7.2gh.

7.1.2.2 Food Industry Materials

Gelatin is too soft and a fragile material. Its direct removal from the mould usually damages the spoofed fingerprint. The fingerprint can be removed with tape, which does not damage the spoof. Images are shown in Figure 7.2ij.

When using **aspic**, some similar problems occurred. The problem is the removal of the spoof from the mould, which is due to the sparseness of aspic and the decomposition of the spoof itself after drying. Some attempts of removing this material through complex methods were made, e.g., by cooling down the spoof or using the glue tape, however, none of them resulted in usable fingerprint falsification. This material failed to be suitable for fingerprint spoof manufacturing purposes.

Gummy bears were also tested, but for an unknown reason they were too sticky. Different ways of melting this material was tested, but the results were not stable enough. The only method which helped a little bit was freezing the mould with the material. The spoof was sticky, but useable.

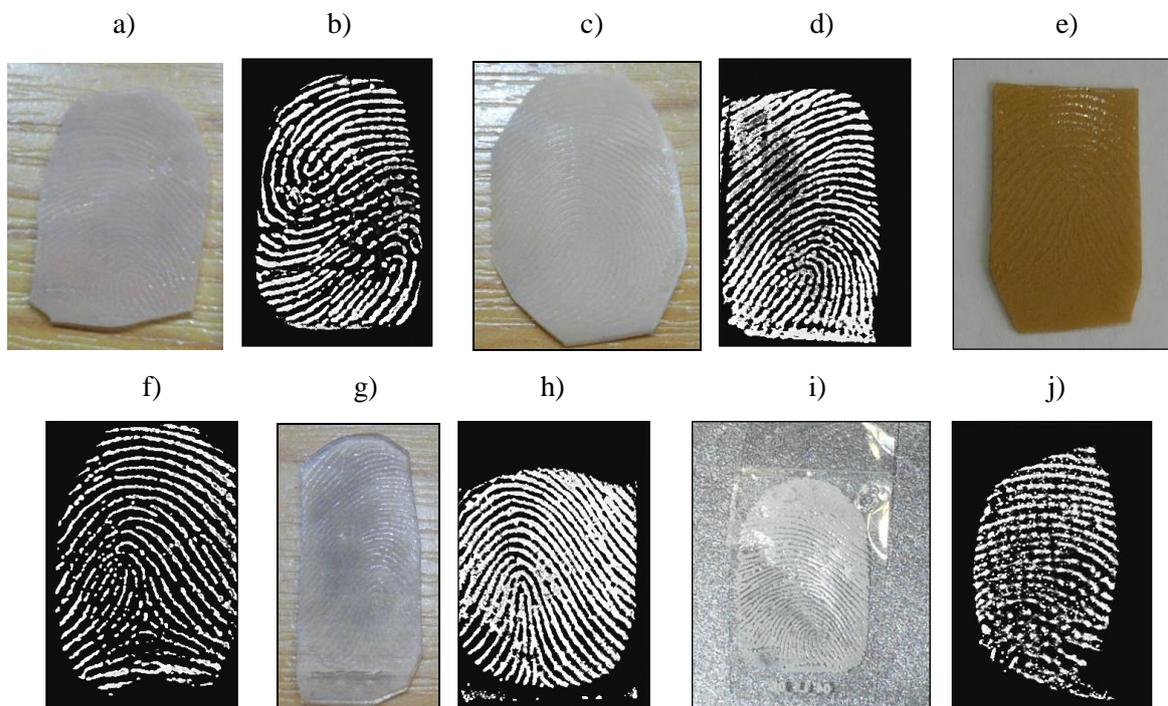


Figure 7.2: Examples of spoofs from different materials – a) b) aquarium silicon, c) d) epoxy resin CHS1200, e) f) epoxy resin Havel Composite L285, g) h) epoxy resin HobbyKing, i) j) gelatin spoof.

7.1.2.3 Creative Materials

Fimo Standard is a polymer clay which is used to make jewellery and accessories. It is easily malleable and can be hardened by baking it in the oven. After that, the clay is hard and keeps its shape so it can be drilled, sliced, etc. The production of fingerprint spoofs was done without the baking part. The Fimo Standard spoofs were very malleable. Images are shown in Figure 7.3ab.

Oyumare is a silicone-based reusable modelling compound. When heated in boiling water, it becomes soft. After cooling, it hardens back into a rubbery, flexible plastic. Images are shown in Figure 7.3cd.

Glass Colours are special colours which can be used on glass, windows, mirrors, bathroom tiles, etc. The colours are applied to a plastic foil, and they can be peeled off and stuck to another similar surface. Images are shown in Figure 7.3ef.

Cernit is a polymeric clay and is considered one of the strongest ones. It can be hardened by baking it in an oven. Fingerprint spoofs were made without the hardening phase. The results are more persistent than the other polymeric clays, but it still fades after several uses. Images are shown in Figure 7.3gh.

Gel wax itself is a combination of polymer resin and mineral oil. When heated in boiling water, the gel turns to a liquid state. After cooling down to room temperature it is solid (jelly) again. There is a risk of damage to the spoof caused by air bubbles. Images are shown in Figure 7.3ij.

Kato polyclay is considered to be the strongest polymeric clay. It can be hardened by baking it in the oven. When creating spoofs, Kato was not hardened. Images are shown in Figure 7.3kl.

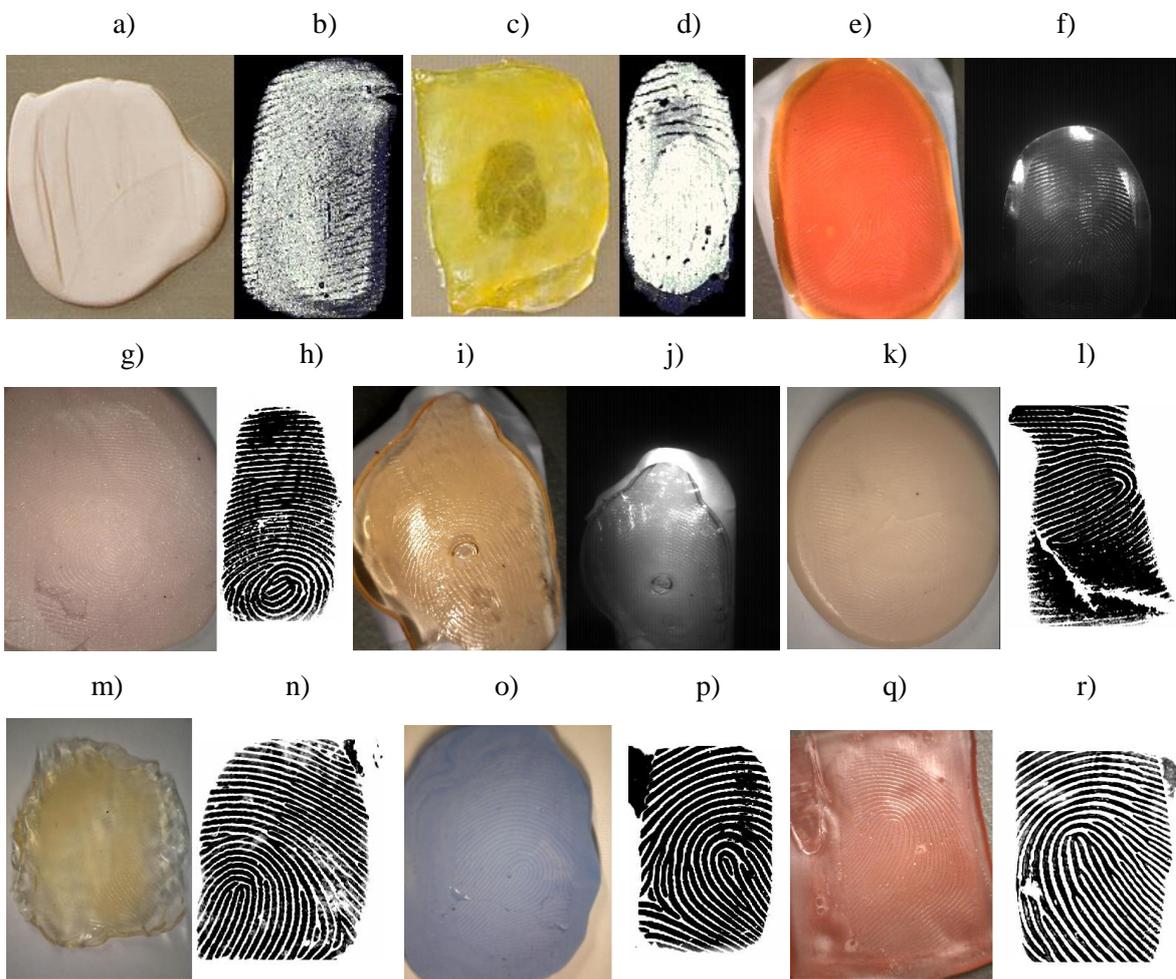


Figure 7.3: Examples of spoofs from different creative materials – a) b) Fimo standard, c) d) Oyumare, e) f) glass colours, g) h) Cernit, i) j) gel wax, k) l) Kato, m) n) latex, o) p) Siligum, q) r) wax sheets.

Latex is a white fluid that can be found inside of some plants. It is a base for natural rubber, tires, etc. When making spoofs, the biggest challenge is to make an equally thick layer of latex. Images are shown in Figure 7.3mn.

Siligum is two components (blue and white) silicon modelling paste. When both components are mixed together in a 1-to-1 ratio, a light blue mixture is made. In this state it should be pushed into the mould, and after around 5 to 15 minutes it makes a solid, but supple spoof. Images are shown in Figure 7.3op.

Different types of **waxes** are synthesized by some plants and animals. A wax sheet was used when creating the fingerprint spoofs, sheets which were originally designed for dentists. They can be heated until the wax gets soft and can be pushed to the mould. Images are shown in Figure 7.3qr.

7.1.3 Spoof Damage Analysis

As can be seen, fingerprint spoofs generally have a lower quality than real fingerprints, and these flaws in quality could be simulated as fingerprint damage done by using a spoof fingerprint. A couple examples of these flaws are: **air bubbles** in the material, which are created by an imperfect filling of the mould (can be seen in Figure 7.3ijr), **broken ridges** in the edge of the fingerprint caused by imperfect mould creation (can be seen in Figure 7.2j and Figure 7.3bdl), and **imperfect edges** of the fingerprint because of the too regular (or specifically irregular) edges of the mould (which can be seen in Figure 7.2pr and Figure 7.3npr). Some of these imperfections can be avoided by the thorough creation of a mould and fingerprints spoofs, but some materials make creation of a fingerprint spoof without flaws almost impossible. There are several flaws described in the literature [97]: background noise, overall shape, clear external contours, missing sections, unreal distortion, unexpected appearance (residue from spoof material), air bubbles, absence of sweat pores, narrow valleys, and reproducible artefacts.

Nevertheless, the difference between the damages of real, damaged fingerprint and an almost perfect spoof is a very difficult task [S1]. There are some software-based liveness detection systems (more recently called “presentation attack detection systems” [98]) that look for perspiration, skin elasticity, or the specific characteristics of the images [99]. In general, a lot of methods emerged from LivDet competitions [100]. Many of the algorithms have problems with “new” materials (different than that used for training) [101] [102].

In summary, it was determined not to simulate this type of damage. There is some promising research going on regarding presentation attack detection using user-specific effects [103]. Perhaps the results of this research could be used for the meaningful design of this damage simulation. On the other hand, it turns out that using synthetic fingerprint as a source image for mould creation can be very beneficial. When testing the materials, an imperfect mould can influence the results. Using synthetic fingerprints removes these imperfections. That is also the reason why every damage that can be seen in Figure 7.3 is made by a spoof creation and material.

7.2 Detergents or Lotions

The last subchapter describes fingerprints influenced by detergents and lotions used in everyday life. There are situations where users are forced to clean their hands. On the contrary, it is really common that the user has unclean hands. As stated in Subchapter 3.2, liquids and conductive materials are the worst. The situation when a user is going through a fingerprint access device after washing their hands or using hand lotion can be very common as well as the situation when a user is unlocking his or her mobile phone or laptop, utilizing fingerprint technology with a polluted hand. [53]

Once again, a preliminary database has to be acquired. In this case, only a small database was acquired. When the database was created, the emphasis was on getting the damaged fingerprints. The process of damaging the real fingerprint before the acquirement has to be described in detail to quantify the amount of damage done by various detergents or lotions. The acquired database shows that almost all detergents or lotions leave marks that damage the fingerprint. The fingerprint database based on these specific types of pollution of a user's hand clearly shows that damage done to the fingerprint can be quite severe. With specific combinations of detergents and sensors, it can occur where parts of the finger that are polluted are invisible to the sensor. Simulating these everyday damages can be used in creating a specific database for testing recognition biometric systems in the previously stated situations.

As can be seen in Figure 7.4, damage can be very specific, which can lead to either a specific simulation or to adapting the damages from the database. Overall, this is a promising area of future research. [S1]



Figure 7.4: Examples of fingerprints damaged by a) shower gel, b) lotion and c) dish washer liquid.

8 Conclusion

This thesis covers the state-of-the-art techniques of fingerprint acquirement and recognition. What has also been described are the methods of generating a synthetic fingerprint and fingerprint reconstruction. This description was focused on the SFinGe generator and its methods. All of the phenomena that can supposedly damage the image of a fingerprint created by a biometric device are listed. Based on this information, a Fingerprint Generation Petri net was created. The SyFDaS application that was implemented follows the Fingerprint Generation Petri net. It contains a fingerprint generator and a damage simulator in a clear GUI, which is suitable for expansion. There are described algorithms for database generation, for basic damage simulations – a damaged sensor, pressure and moisture, and fingerprint distortion and for advanced damage simulations – a swipe mode, a narrow sensor, warts, an atopic dermatitis, and psoriasis.

The core of this doctoral thesis is the discussion of the phenomena that are specific to swipe sensors, skin diseases, and other interesting damages. A thorough analysis of the phenomena specific to swipe sensors are shown in two categories. The first category consists of the phenomena that are directly specific to these sensors, i.e. a narrow acquired fingerprint and a long fingerprint. The second category focuses on phenomena that are common to touch sensors but have to be simulated differently with a swipe sensor. The main idea is that the swipe sensor can be divided into small segments of a touch sensor that are merged together. Damage simulations have to be done for each segment with the acknowledgement that the input data of these simulations have to be similar and then merged together.

Skin diseases can be a big problem when using widespread biometric systems. The systems that are currently used are not capable of detecting skin diseases or enhancing the quality of fingerprints with diseases, which basically makes the biometric system unsuitable for some users. The main reason is that it is very difficult to obtain access to a database of fingerprints with skin diseases. The available database of skin diseases that influences fingerprints has been analysed. Skin diseases and an algorithm for their detection has also been proposed. The several possibilities regarding how to simulate the damage done by skin diseases to fingerprints is also described. The first possibility is to directly simulate the effects of a disease. Warts and atopic dermatitis are examples of this approach. A more thorough description of these diseases is given and algorithms to simulate them are proposed. The second possibility is to use machine learning or neural networks. An algorithm to simulate psoriasis is an example of using this method.

The last area of interest is fingerprint spoofing and detergents. The methods for the production of fingerprint spoofs have been described. Various materials which can and were used to produce spoofs have been analysed. The possible damages made when producing fingerprint spoof were discussed, and the next category deals with the effects of detergents or lotions on fingerprints. The preliminary database needed for research in this area is described.

As an input, there were 300 undamaged synthetic fingerprints (from three generators and in two width variants), 20 basic damages from diseases, and 1,151 basic and combined damages in swipe sensors used. Overall, 348,300 different damaged synthetic fingerprint images were generated. All of the 43 basic damages were visually checked and compared with real images with respective damage.

Reference fingerprints with no damage and all 1,171 damages were verified using four different quality measurement methods (VeriFinger's quality score, VeriFinger's comparison score, NFIQ, and Oravec's quality metric). All damages have their median scores lower than the respective scores of the reference images. The best of the damage was extreme psoriasis damage (*psor5*), which has its average median scores only 38 % of the reference scores.

These results show that fingerprint images were successfully damaged. The generated database and all of the results can immediately be used for the analysis of quality measurements methods as well as for the creation of new methods for quality estimation. The size of the database (and the possibility to generate new data if needed) allows for the use of machine learning or neural networks to assist with this task. Fingerprint image enhancement methods could be analysed and their limitation could be found through a generated database. For both of these tasks there is the possibility to define new instances of damage (by changing the input values for the desired damage type, or in the case of swipe sensor simulation by giving different inputs to merging algorithms). There are a great number of possibilities in the area of public education regarding synthetic images and their damaged impressions. One of the reasons is that there is not a risk of revealing a person's identity with the exemplary images shown. Images mainly focused on diseases could be used for educating forensic experts and dermatologists. The achieved results are a stepping stone for other areas of research.

Future work can be focused on several areas. The first of them being the extension of simulated damages. Different diseases can be simulated (the probabilistic method could be used), some damages in swipe sensor simulation could also be done (the second phalange, faults of reconstruction algorithm, another approach on the merge algorithm), and generally some new damages can be simulated like the effects of sensor technology (or the background of the images in general), motion blur made by non-cooperative behaviour and effects of detergents, lotions, and other everyday products.

The second area of the future research could deal with synthetic images for the development of new algorithms or the enhancement of existing ones. Examples could be the development of the detection and extraction of the damaged regions in fingerprint images, the reconstruction and enhancement of quality in these damaged regions, or "only" the recognition and evaluation of severity of these damages. Created database of images could be used for this purpose right now, but it would be even better to have a precise annotation of the damage done to images (which is possible with the damage simulation done to perfect images). All these algorithms together could expand the usability of fingerprint recognition in general.

The final area of research could focus on using machine learning and neural networks. These approaches require a huge database as an input, but if these input images are provided, they would show incredible results. A synthetic database is great for this. On the other hand, there is a risk of overfitting these synthetic damages (a method will then detect only synthetic damages and not generalized ones). Nevertheless, machine learning and neural networks could be used to address the problems mentioned in the second area. In addition, the method for presentation attack detection as an optional part of fingerprint image enhancement could be imagined with synthetic images damaged to look like spoofs, thus avoiding the question whether it is more important to detect a spoof or to enhance the quality of an image.

In summary, the theoretical description of fingerprint acquirement, recognition, synthetic generation, skin diseases, and other damages was given and a practical solution to synthetic fingerprint

damage simulation (and generation) with a focus on swipe mode, narrow sensor, damaged sensor, pressure/moisture, fingerprint distortion, warts, atopic dermatitis and psoriasis was also described. A lot of damages were proposed, all of which were verified to inflict the assumed damage to the fingerprint image. New and interesting areas for future research were proposed, some of which are very promising and can be researched with some additional work. Other areas, however, can now be explored with the presented results.

Bibliography

- [1] Kanich O.: *Fingerprint Damage Simulation – A Simulation of Fingerprint Distortion, Damaged Sensor, Pressure and Moisture*. Lambert Academic Publishing GmbH & Co. KG, 2014, p. 57. ISBN 978-3-659-63942-5.
- [2] Lai K.K., Kanich O., Dvořák M. et al.: *Biometric-Enabled Watchlists Technology*. IET Biometrics, IET, 2017, p. 10. ISSN 2047-4938.
- [3] Li S.Z., Jain A.K.: *Encyclopedia of Biometrics*. Springer, 2015, p. 1651. DOI 10.1007/978-1-4899-7488-4.
- [4] International Organization for Standardization: *International standard ISO/IEC 2382-37:2017 Information technology – Vocabulary – Part 37: Biometrics*. ISO/IEC JTC 1/SC 37 Biometrics, 2017, p. 27.
- [5] Drahanický M., Orság F., Doležel M. et al.: *Biometrie (Biometrics)*. Computer Press a.s., 2011, p. 294. ISBN 978-80-254-8979-6.
- [6] Maltoni D., Maio D., Jain A.K. et al.: *Handbook of Fingerprint Recognition*. Springer, 2009, p. 512. ISBN 978-1-8488-2254-2.
- [7] Jain A.K., Bolle R., Pankanti S. et al.: *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publications, 1999, p. 411. ISBN 978-0-7923-8345-1.
- [8] Matyáš V., Říha Z.: *Security of Biometric Authentication Systems*. International Journal of Computer Information Systems and Industrial Management Applications, 2011, pp. 174-184. ISSN 2150-7988.
- [9] Kanich O., Drahanický M.: *State of the Art in Fingerprint Recognition*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 28. ISBN 978-1-78561-224-4.
- [10] Drahanický M.: *Fingerprint Recognition Technology – Related Topics*. Lambert Academic Publishing GmbH & Co. KG, 2011, p. 172. ISBN 978-3-8443-3007-6.
- [11] Chaloupka R.: *Generátor Otisků Prstů (Fingerprint Generator)*. Master's thesis FIT BUT, 2007, p. 47.
- [12] Březinová E.: *Outer Hand Physiology and Diseases*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 26. ISBN 978-1-78561-224-4.
- [13] Štork J. et al.: *Dermatovenerologie (Dermatovenerology)*. Galén, 2013, p. 502. ISBN 9788072628988.
- [14] U.S. Department of Justice: *The Fingerprint Sourcebook*. CreateSpace Independent Publishing Platform, 2014, p. 428. ISBN 978-1502828422.

- [15] Drahanický M., Kanich O., Březinová E. et al.: *Experiments with Optical Properties of Skin on Fingers*. International Journal of Optics and Applications, 2017, pp. 37-46. DOI 10.5923/j.optics.20160602.03.
- [16] Drahanický M., Kanich O., Březinová E.: *Is Fingerprint Recognition Really so Reliable and Secure?*. Challenges for fingerprint recognition – spoofing, skin diseases and environmental effects, Handbook of Biometrics for Forensic Science, Springer, 2017, p. 21. ISBN 978-3-319-50671-5.
- [17] Drahanický M., Kanich O., Pernický R. et al.: *Verarbeitung von beschädigten Fingerabdrücken in der polizeilichen Praxis*. Datenschutz und Datensicherheit, Springer, 2017, pp. 407-414. ISSN 1614-0702, DOI 10.1007/s11623-017-0803-2.
- [18] Federal Bureau of Investigation: *Integrated Automated Fingerprint Identification System – Web page* [online]. 2014. [cit. 2014-5-21]. Available online: http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis.
- [19] Drahanický M.: *Biometric Systems* [online]. Course at the FIT BUT, 2017. [cit. 2017-7-28] Available online: <http://www.fit.vutbr.cz/study/courses/BIO/>.
- [20] Cappelli R., Maltoni D.: *On the Spatial Distribution of Fingerprint Singularities*. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2009, pp. 742-748. DOI 10.1109/TPAMI.2008.243.
- [21] Zhao Q., Jain A.K., Paulter N.G. et al.: *Fingerprint Image Synthesis Based on Statistical Feature Models*. 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), IEEE, 2012, pp. 23-30. ISBN 978-14-673-1384-1.
- [22] Rak R., Matyáš V., Říha Z. et al.: *Biometrie a identita člověka (Biometrics and Person's Identity)*. Grada, 2008, p. 664. ISBN 978-80-247-2365-5.
- [23] Du R.W., Yang C., Kou C.I.: *System for Fingerprint Image Reconstruction Based on Motion Estimate Across a Narrow Fingerprint Sensor*. U.S. Patent US7212658B2, 2004, p. 25.
- [24] Ratha N.K., Govindaraju V.: *Advances in Biometrics: Sensors, Algorithms and Systems*. Springer, 2008, p. 503. ISBN 978-1-84628-920-0.
- [25] O’Gorman L., Xia X.: *Method and System for Capturing Fingerprints from Multiple Swipe*. U.S. Patent US0123714A1, 2003, p. 20.
- [26] Chou B.C.S.: *Sweep-type fingerprint sensors module*. U.S. Patent US7200250B2, 2007, p. 11.
- [27] Xia X., O’Gorman L.: *Innovations in Fingerprint Capture Devices*. Pattern Recognition, Elsevier, 2003, pp. 361-369. DOI 10.1016/S0031-3203(02)00036-5.

- [28] Mardiansyah A.Z., Bejo A., Hidayat R.: *Fingerprint Image Reconstruction for Swipe Sensor Using Predictive Overlap Method*. MATEC – the 2nd International Conference on Engineering and Technology for Sustainable Development, 2018, p. 4. DOI 10.1051/mateconf/201815401042.
- [29] Kanich O., Drahanický M.: *Currently Used Swipe Fingerprint Sensors*. International Journal of Bio-Science and Bio-Technology, SERSC, 2016, pp. 381-386. ISSN: 2233-7849.
- [30] Aukorius E., Boccara A.C.: *Fingerprint Imaging from the Inside of a Finger with Full-field Optical Coherence Tomography*. Biomedical Optics Express, Optical Society of America, 2015, pp. 4465-4471. DOI 10.1364/BOE.6.004465.
- [31] Mainguet J.F.: *Personal Website – Fingerprint* [online]. 2015 [cit. 2015-26-02]. Available online: <http://fingerchip.pagesperso-orange.fr/biometrics/types/fingerprint.htm>.
- [32] Apple Insider: *Apple preparing software update to enhance functionality of iPhone 5s Touch ID* [online]. 2016 [cit. 2016-13-12]. Available online: <http://appleinsider.com/articles/14/02/28/apple-preparing-software-update-to-enhance-functionality-of-iphone-5s-touch-id>.
- [33] Koundinya P., Theril S., Feng T. et al.: *Multi Resolution Touch Panel with Built-in Fingerprint Sensing Support*. Design, Automation and Test in Europe Conference and Exhibition (DATE), IEEE, 2014, p. 6. DOI 10.7873/DATE.2014.258.
- [34] NEXT Biometrics Group: *The NEXT Advantage* [online]. 2015 [cit. 2015-26-02]. Available online: http://nextbiometrics.com/the_products/the_next_advantage/.
- [35] Charlot B., Parrain F., Galy N. et al.: *A Sweeping Mode Integrated Fingerprint Sensor With 256 Tactile Microbeams*. Journal of Microelectromechanical Systems, IEEE, 2004, pp. 636-644. ISSN 1057-7157, DOI 10.1109/JMEMS.2004.832195.
- [36] Cappelli R., Ferrara M., Maltoni D.: *Minutiae-Based Fingerprint Matching*. Cross Disciplinary Biometric Systems, Intelligent Systems Reference Library, Springer, 2012, pp. 117-150. DOI 10.1007/978-3-642-28457-1_7.
- [37] Wang P.S.P., Yanushkevich S.N.: *Biometric Technologies and Applications*. Artificial Intelligence and Applications, 2007, pp. 249-254. ISBN 978-0-88986-631-7.
- [38] Yanushkevich S.N., Stoica A., Shmerko V.P. et al.: *Biometric Inverse Problems*. CRC Press, 2005, p. 416. ISBN 978-0849328992.
- [39] Yanushkevich S.N.: *Synthetic Biometrics: A Survey*. The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 676-683. DOI 10.1109/IJCNN.2006.246749.
- [40] Cappelli R.: *SFinGe: An Approach to Synthetic Fingerprint Generation*. In BT 2004 – International Workshop on Biometric Technologies, 2004, pp. 147-154.

- [41] Ansari A.H.: *Generation and Storage of Large Synthetic Fingerprint Database*. Master's thesis Indian Institute of Science, 2011, p. 47.
- [42] Feng J., Jain A.K.: *Fingerprint Reconstruction: From Minutiae to Phase*. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2010, pp. 209-223. ISSN 0162-8828, DOI 10.1109/TPAMI.2010.77.
- [43] Kücken M., Newell A.C.: *A Model for Fingerprint Formation*. European Physical Society, EPL (Europhysics Letters), 2004, pp. 141-146. DOI 10.1209/epl/i2004-10161-2.
- [44] Kanich O., Drahanický M.: *Simulation of Synthetic Fingerprint Generation Using Petri Nets*. IET Biometrics, IET, 2017, p. 17. ISSN 2047-4938, DOI 10.1049/iet-bmt.2016.0041.
- [45] Cappelli R., Ferrara M., Maltoni D.: *Generating Synthetic Fingerprints*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 24. ISBN 978-1-78561-224-4.
- [46] Galbally J., Cappelli R., Lumini A. et al.: *Fake Fingertip Generation from a Minutiae Template*. 19th International Conference on Pattern Recognition, IEEE, 2008, p. 4. DOI 10.1109/ICPR.2008.4761456.
- [47] Cappelli R., Maio D., Lumini A. et al.: *Fingerprint Image Reconstruction from Standard Templates*. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2007, pp. 1489-1503. DOI 10.1109/TPAMI.2007.1087.
- [48] Vinoth A., Saravanakumar S.: *An Analysis of Altered Fingerprint Detection, Recognition and Verification*. International Journal of Computer Science and Mobile Computing, 2006, pp. 178-182. ISSN 2320-088X.
- [49] Feng J., Jain A.K., Ross A.: *Fingerprint Alteration*. MSU Technical Report, MSU-CSE-09-30, 2009, p. 13.
- [50] Mihaiu A-I.: *Changes in the Papillary Structure (I)*. Romanian Journal of Forensic Science, Romanian Society of Legal Medicine, 2012, pp. 1148-1158. ISSN 2069-2617.
- [51] Tuč D.: *Testing of the Environmental Influences on Fingerprints Sensors*. Bachelor's project FIT BUT, 2005, p. 63.
- [52] Drahanický M., Kanich O.: *Vulnerabilities of Biometric Systems*. Security and Protection of Information 2015, 2015, pp. 53-60. ISBN 978-80-7231-997-8.
- [53] Heidari M., Kanich O., Drahanický M.: *Processing of Fingerprints Influenced by Skin Diseases*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 34. ISBN 978-1-78561-224-4.
- [54] Alessandroni A., Cappelli R., Ferrara M. et al.: *Definition of Fingerprint Scanner Image Quality Specifications by Operational Quality*. Biometrics and Identity Management, Springer, 2008, pp. 29-36. DOI 10.1007/978-3-540-89991-4_4.

- [55] University of Bologna: *Biometric System Laboratory – Web page* [online]. 2014 [cit. 2014-1-13]. Available online: <http://biolab.csr.unibo.it/research.asp>.
- [56] Reisig W.: *Understanding Petri Nets*. Springer-Verlag, 2013, p. 230. ISBN 978-3-642-3327-7.
- [57] Češka M.: *Petriho Síťě (Petri Nets)*. Akademické nakladatelství CERM Brno, 1994, p. 94. ISBN 8-085-86735-4.
- [58] Badouel E., Bernardinello L., Dorondeau P.: *Petri Net Synthesis*. Springer-Verlag, 2015, p. 339. ISBN 978-3-662-47966-7.
- [59] Popova-Zeugmann L.: *Time and Petri Nets*. Springer-Verlag, 2013, p. 209. ISBN 978-3-642-41114-4.
- [60] Gonzalez R.C., Woods R.E.: *Digital Image Processing (3rd Edition)*. Prentice Hall, 2008, p. 954. ISBN 978-0-1316-8728-8.
- [61] Singh S., Murshed N., Kropatsch W.: *Modelling Plastic Distortion in Fingerprint Images*. Second International Conference on Advances in Pattern Recognition (ICAPR2001), Springer, 2001, pp. 369-376. ISBN 978-3-540-41767-2, DOI 10.1007/3-540-44732-6_38.
- [62] Ralston A.: *Základy Numerické Matematiky (Basics of Numerical Mathematics)*. Academia, 1973, p. 636.
- [63] Haluzíková A.: *Numerické Metody (Numerical Methods)*. BUT, 1989, p. 124. ISBN 80-214-0039-0.
- [64] Press W.H., Teukolsky S.A., Vetterling W.T. et al.: *Numerical Recipes in C: The Art of Scientific Computing (2nd Edition)*. Cambridge University Press, 2002, p. 925. ISBN 0-521-43108-5.
- [65] Mainguet J.F.: *Fingerprint-reading System*. U.S. Patent US6459804B2, 1996, p. 11.
- [66] Ratha N., Bolle R.: *Automatic Fingerprint Recognition Systems*. Springer, 2004, p. 458. ISBN 978-0-387-95593-3, DOI 10.1007/b97425.
- [67] Bárta M.: *Generation of Skin Disease into the Synthetic Fingerprints*. Master's thesis FIT BUT, 2016, p. 60.
- [68] Oravec T.: *Methodology of Fingerprint Image Quality Measurement*. Master's thesis FIT BUT, 2018, p. 55.
- [69] NEUROtechnology: *MegaMatcher 10.0, VeriFinger 10.0, VeriLook 10.0, VeriEye 10.0 and VeriSpeak 10.0 SDK – Developer's Guide*. Version: 10.0.0.0., 2018, p. 2521.
- [70] University of Bologna, *FVC ongoing* [online]. 2018 [cit. 2018-06-03]. Available online: <https://biolab.csr.unibo.it/FVCOnGoing/UI/Form/PublishedAlgs.aspx#&&P31GGJ/RJNGcPuomUGuJNNWnvH92kBxijVZU9vA5IHsj3ZwH8fAAL50o9QhBvV4OiG/8ztCeIpFHNY76gOYN5eOevHNNH1cGJsN+BmA8UD3nzldkLRwy3HRR+By11M03NVon+/yAQtPIH63rtuq+/0uE9vRY=>.

- [71] Garris M.D., Tabassi E., Wilson C.L.: *NIST Fingerprint Evaluations and Developments*. Proceedings of IEEE, IEEE, 2006, pp. 1915-1926. ISSN 1558-2256, DOI 10.1109/JPROC.2006.885130.
- [72] Tabassi E., Wilson C.L., Watson C.I.: *Fingerprint Image Quality*. NISTIR 7151, National Institute of Standards and Technology, 2004, p. 72.
- [73] Habif T.P.: *Clinical Dermatology (4th Edition)*. Mosby, 2004, p. 1004. ISBN 978-0-323-01319-2.
- [74] Wolff K., Johnson R.A., Suurmond D.: *Fitzpatrick's Color Atlas and Synopsis of Clinical Dermatology (5th Edition)*. McGraw-Hill, 2005, p. 1085. ISBN 0-07-144019-4.
- [75] Jain A.K., Flynn P., Ross A.A.: *Handbook of Biometrics*. Springer-Verlag, 2008, p. 556. ISBN 978-0-387-71040-2.
- [76] Barotová Š.: *Detector of Skin Diseases by Fingerprint Technology*. Bachelor's thesis FIT BUT, 2017, p. 50.
- [77] Doležel M., Drahanský M., Urbánek J. et al.: *Influence of Skin Diseases on Fingerprint Quality and Recognition*. New Trends and Developments in Biometrics, IntechOpen, 2012, pp. 275-303. DOI 10.5772/51992.
- [78] Barotová Š., Drahanský M., Pernický R.: *Detection of Ridge Damages in Fingerprint Recognition Caused By Skin Diseases*. International Journal of Signal Processing, SERSC, 2016, pp. 125–146. DOI 10.14257/ijcip.2016.9.11.13.
- [79] James W.D., Berger T.G., Elston D.M.: *Andrews' Diseases of the Skin Clinical Dermatology (10th Edition)*. Elsevier, 2006, p. 691. ISBN 0-7216-2921-0.
- [80] Khanna N., Singh S.: *Bhutani's Color Atlas of Dermatology*. Jaypee Brothers Medical Publishers (P) Ltd., 2015, p. 498. ISBN 978-93-5152-302-4.
- [81] Kerkhof van de P.C.M.: *Textbook of Psoriasis*. Blackwell Publishing Ltd., 2003, p. 348. ISBN 1-4051-0717-0.
- [82] Russ J.C., Neal F.B.: *The Image Processing Handbook (7th Edition)*. CRC Press, 2016, p. 1035. ISBN 978-1-4987-4028-9.
- [83] Powers D.: *Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation*. Journal of Machine Learning Technologies, Bioinfo Publications, 2011, pp. 37-63. ISSN 2229-3981.
- [84] Bárta M., Drahanský M.: *Generation of Skin Diseases into Synthetic Fingerprints*. International Journal of Image Processing, CSC Journals, 2016, pp. 229-248. ISSN 1985-2304.
- [85] Schellhaas U., Gerber W., Hammes S. et al.: *Pulsed Dye Laser Treatment Is Effective in the Treatment of Recalcitrant Viral Warts*. Dermatologic surgery, 2008, pp. 67-72. ISSN 1076-0512.

- [86] Ciconte A., Campbell J., Tabrizi S. et al.: *Warts Are not Merely Blemishes on the Skin: A Study on the Morbidity Associated with Having Viral Cutaneous Warts*. Australasian journal of dermatology, 2003, pp. 169-173. ISSN 0004-8380.
- [87] Ring J., Alomar A., Bieber T. et al.: *Guidelines for Treatment of Atopic Eczema (Atopic Dermatitis) Part I*. Journal of the European Academy of Dermatology and Venereology, 2012, pp. 1045-1060. ISSN 0926-9959.
- [88] Lee C., Chang C., Johar A. et al.: *Fingerprint Changes and Verification Failure Among Patients with Hand Dermatitis*. JAMA Dermatology, 2013, pp. 294-299. ISSN 2168-6068.
- [89] Košťák D.: *Generation of Skin Disease Effects into Synthetic Fingerprints from Anguli Generator*. Bachelor's thesis FIT BUT, 2018, p. 44.
- [90] Papi S., Ferrara M., Maltoni D. et al.: *On the Generation of Synthetic Fingerprint Alterations*. 2016 International Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, 2016, p. 6. DOI 10.1109/BIOSIG.2016.7736930.
- [91] Galbally J., Cappelli R., Lumini A. et al.: *An Evaluation of Direct Attacks Using Fake Fingers Generated from ISO Templates*. Pattern Recognition Letters, Elsevier, 2010, pp. 725-732. DOI 10.1016/j.patrec.2009.09.032.
- [92] Spurný J., Doležel M., Kanich O. et al.: *New materials for spoofing touch-based fingerprint scanners*. Proceedings of International Conference on Computer Application Technologies 2015, 2015, p. 15. ISBN 978-1-4673-8211-3.
- [93] Dražanský M., Kanich O., Dvořák M.: *Spoofing methods in hand-based biometrics*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 32. ISBN 978-1-78561-224-4.
- [94] Rattani A., Ross A.: *Automatic Adaptation of Fingerprint Liveness Detector to New Spoof Materials*. 2014 IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2014, p. 8. DOI 10.1109/BTAS.2014.6996254.
- [95] Al-Ajlani A.: *Survey on Fingerprint Liveness Detection*. 2013 International Workshop on Biometrics and Forensics (IWBF), IEEE, 2013, p. 5. ISBN 978-1-4673-4987-1.
- [96] Ghiani L., Yambay D., Mura V. et al.: *LivDet 2013 Fingerprint Liveness Detection Competition 2013*. 2013 International Conference on Biometrics (ICB), IEEE, 2013, p. 6. ISBN 978-1-4799-0310-8, DOI 10.1109/ICB.2013.6613027.
- [97] Champod C., Espinoza M.: *Forgeries of Fingerprints in Forensic Science*. Handbook of Biometrics Anti-Spoofing, Springer, 2014, pp. 13-34. ISBN 978-1-4471-6523-1.
- [98] Schuckers S.: *Presentations and Attacks, and Spoofs, Oh My*. Image and Vision Computing, Elsevier, 2016, pp. 26-30. DOI 10.1016/j.imavis.2016.03.016.

- [99] Schuckers S., Rowe R.: *Antispoofing Methods in Hand-based Biometrics*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 16. ISBN 978-1-78561-224-4.
- [100] Ghiania L., Yambay D.A., Mura V. et al.: *Review of the Fingerprint Liveness Detection (LivDet) competition series: 2009 to 2015*. Image and Vision Computing, Elsevier, 2016, pp. 110-128. DOI 10.1016/j.imavis.2016.07.002.
- [101] Rattani A., Scheirer W.J., Ross A.: *Open Set Fingerprint Spoof Detection Across Novel Fabrication Materials*. IEEE Transactions on Information Forensics and Security, IEEE, 2015, pp. 2447-2460. DOI 10.1109/TIFS.2015.2464772.
- [102] Rattani A., Ross A.: *Automatic Adaptation of Fingerprint Liveness Detector to New Spoof Materials*. IEEE International Joint Conference on Biometrics, IEEE, 2014, p. 8. DOI 10.1109/BTAS.2014.6996254.
- [103] Ghiani L., Marcialis G.L., Roli F.: *Fingerprint Presentation Attacks Detection Based on the User-specific Effect*. 2017 IEEE International Joint Conference on Biometrics, IEEE, 2017, p. 7. DOI 10.1109/BTAS.2017.8272717.

Research and Development Activities

Publications:

- [53] Heidari M., Kanich O., Drahanský M.: *Processing of Fingerprints Influenced by Skin Diseases*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 34. ISBN 978-1-78561-224-4. (my share 15 %)
- [93] Drahanský M., Kanich O., Dvořák M.: *Spoofing methods in hand-based biometrics*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 32. ISBN 978-1-78561-224-4. (my share 40 %)
- [9] Kanich O., Drahanský M.: *State of the Art in Fingerprint Recognition*. Hand-based Biometrics: Methods and Technologies, IET, 2018, p. 28. ISBN 978-1-78561-224-4. (my share 50 %)
- [17] Drahanský M., Kanich O., Pernický R. et al.: *Verarbeitung von beschädigten Fingerabdrücken in der polizeilichen Praxis*. Datenschutz und Datensicherheit, Springer, 2017, pp. 407-414. ISSN 1614-0702, DOI 10.1007/s11623-017-0803-2. (Indexed by DBLP, my share 25 %)
- [2] Lai K.K., Kanich O., Dvořák M. et al.: *Biometric-Enabled Watchlists Technology*. IET Biometrics, IET, 2017, p. 10. ISSN 2047-4938. (IF 1.836, Indexed by Web of Science – 2017 Q2, SCOPUS – 2017 Q2, DBLP, my share 25 %)
 - I. Cited by: Lai K., Yanushkevich S.N.: *Watchlist Risk Assessment Using Multiparametric Cost and Relative Entropy*, 2017 IEEE Symposium Series on Computational Intelligence, IEEE, 2017, p. 7. DOI 10.1109/SSCI.2017.8285219.
- [44] Kanich O., Drahanský M.: *Simulation of Synthetic Fingerprint Generation Using Petri Nets*. IET Biometrics, IET, 2017, p. 17. ISSN 2047-4938, DOI 10.1049/iet-bmt.2016.0041. (IF 1.836, Indexed by Web of Science – 2017 Q2, SCOPUS – 2017 Q2, DBLP, my share 90 %)
 - II. Cited by: Nedjah N., Wyant R.S., Mourelle L.M. et al.: *Efficient Fingerprint Matching on Smart Cards for High Security and Privacy in Smart Systems*, Information Sciences, Elsevier, 2017, p. 18. DOI 10.1016/j.ins.2017.12.038.
- [16] Drahanský M., Kanich O., Březinová E.: *Is Fingerprint Recognition Really so Reliable and Secure?*. Challenges for fingerprint recognition – spoofing, skin diseases and environmental effects, Handbook of Biometrics for Forensic Science, Springer, 2017, p. 21. ISBN 978-3-319-50671-5. (Indexed by SCOPUS, my share 45 %)

- III. Cited by: Iloanusi O.N., Ezema C.A.: *A Quantitative Impact of Fingerprint Distortion on Recognition Performance*. Information Security Journal, 2017, pp. 267-275. ISSN 1939-3555, DOI 10.1080/19393555.2017.1383535.
- IV. Cited by: Pagnin E., Mitrokotsa A.: *Privacy-Preserving Biometric Authentication: Challenges and Directions*. Security and Communication Networks, Hindawi, 2017, p. 9. ISSN 1939-0114, DOI 10.1155/2017/7129505.
- [15] Drahanický M., Kanich O., Březinová E. et al.: *Experiments with Optical Properties of Skin on Fingers*. International Journal of Optics and Applications, 2017, pp. 37-46. DOI 10.5923/j.optics.20160602.03. (Indexed by Web of Science, my share 40 %)
- [29] Kanich O., Drahanický M.: *Currently Used Swipe Fingerprint Sensors*. International Journal of Bio-Science and Bio-Technology, SERSC, 2016, pp. 381-386. ISSN: 2233-7849. (Indexed by SCOPUS – 2016 Q4, my share 85 %)
 - V. Cited by: Mardiansyah A.Z., Bejo A., Hidayat R.: *Fingerprint Image Reconstruction for Swipe Sensor Using Predictive Overlap Method*. MATEC – the 2nd International Conference on Engineering and Technology for Sustainable Development, 2018, p. 4. DOI 10.1051/matecconf/201815401042.
- [52] Drahanický M., Kanich O.: *Vulnerabilities of Biometric Systems*. Security and Protection of Information 2015, 2015, pp. 53-60. ISBN 978-80-7231-997-8. (my share 50 %)
- [92] Spurný J., Doležel M., Kanich O. et al.: *New materials for spoofing touch-based fingerprint scanners*. Proceedings of International Conference on Computer Application Technologies 2015, 2015, p. 15. ISBN 978-1-4673-8211-3. (Indexed by SCOPUS, my share 30 %)
 - VI. Cited by: Bécue A., Champod C.: *Fingermarks and Other Body Impressions – A Review (July 2013 – July 2016)*. Proceedings of the Eighteenth International Forensic Science Managers Symposium, INTERPOL, 2016, p. 72.
- [1] Kanich O.: *Fingerprint Damage Simulation – A Simulation of Fingerprint Distortion, Damaged Sensor, Pressure and Moisture*. Lambert Academic Publishing GmbH & Co. KG, 2014, p. 57. ISBN 978-3-659-63942-5. (Monography, my share 100 %)

Accepted Publications:

- [A1] Drahanický M., Kanich O.: *Influence of skin diseases on fingerprints*. Biometrics under Biomedical Considerations, Springer, 2018, p. 40. (my share 40 %)
- [A2] Kanich O., Mézl M., Drahanický M.: *Creative Materials Used for Fingerprint Spoofs*. Proceedings of International Workshop on Biometrics and Forensics 2018, IEEE, 2018, p. 8. (my share 60 %)

Submitted Publications:

- [S1] Drahanický M., Kanich O.: *Enhancement of Fingerprint Image Quality versus Liveness Detection on Fingers*. Computer Vision and Image Understanding, Elsevier, 2018, p. 24. ISSN 1077-3142. (IF 2.498, my share 25 %)

Publications in Preparation:

- [P1] Kanich O., Drahanický M.: *Swipe Sensor Damage Simulation into Synthetic Fingerprint*. IET Biometrics, IET, 2018, p. 19. ISSN 2047-4938. (IF 1.382, Indexed by SCOPUS, my share 95 %)
- [P2] Kanich O., Košťák D., Drahanický M.: *Psoriasis Damage Simulation into Synthetic Fingerprint*. 12th IAPR International Conference on Biometrics 2019, 2019, p. 8. (my share 65 %)

Projects:

- *Modern and open engineering studies*. MOST (EU – 2017-2022).
- *Secure and Reliable Computer Systems*. FIT-S-17-4014 (CZ – 2017-2019).
- *Tools and methods for video and image processing to improve effectivity of rescue and security services operations*. VI20172020068 (CZ – 2017-2020).
- *IT4Innovations excellence in science*. LQ1602 (CZ – 2016-2020).
- *Reliability and Security in IT*. FIT-S-14-2486 (CZ – 2014-2016).
- *New solutions for multimodal biometrics – enhancement of security and reliability of biometric technologies*. COST LD14013 (CZ – 2014-2016).

Teaching:

- Biometric Systems: 2014/2015, 2015/2016, 2016/2017, 2017/2018 – laboratories, projects, lectures, examinations.
- Assembly languages: 2014/2015 – laboratories.
- Machine Level Programming: 2015/2016, 2016/2017 – laboratories.
- Fundamentals of Artificial Intelligence: 2014/2015, 2015/2016, 2016/2017 – laboratories.
- Intelligent Sensors: 2014/2015 – laboratories.
- Parallel and Distributed Algorithms: 2015/2016 – projects.
- Leading of bachelor's thesis (Košťák D., Glesner L.), master's thesis (Oravec T.).
- Review of bachelor's thesis (Vančo P., Míšová M.), master's thesis (Bárta M.).

Others:

- Dražanský M., Dvořák R., Váňa J., et al.: *A multispectral lifecycle detector especially suited for the fingerprint recognition technology*. Utility model, ÚPV-31405. 2018.
- Commenting of standards ISO/IEC: JTC 1/SC 37 N 6754 and JTC 1/SC 37 N 6756, 2018.
- Excursion to School of Criminal Justice (ESC) of the University of Lausanne (prof. Champod), 12/2017.
- Doležal M., Kanich O., Dražanský M.: *Application for Detection of Fingerprint Area*. Software. 2016.
- Participation in 12th and 15th Summer School for Advanced Studies on Biometrics for Secure Authentication 2015 and 2018.
- Participation in 2 months research internship at University of Calgary (Alberta, Canada), 04-06/2016.